CSS-PMC: a Combined Enhancement/Compensation Scheme for Continuous Speech Recognition in Noise

J. A. Nolazco Flores & S. J. Young

CUED/F-INFENG/TR.128 June 1993

Cambridge University Engineering Department
Trumpington Street
Cambridge, CB2 1PZ
England

Email: jn106/sjy @eng.cam.ac.uk

Abstract

Training HMMs on the same conditions as in recognition makes models learn not only the features of the speech, but also those of the environment. However, attempting to produce models for all possible environments is impractical. One way to solve this problem is to compensate models trained on clean speech to give "artificially" adapted models. The goal of these noise adaptation techniques is to reach the same recognition performance as would be obtained by training in the noisy conditions.

However, even training in noise can only achieve limited recognition performance because the high variance at low SNR makes the features begin to overlap thereby reducing discrimination. The problem is even worse when the vocabulary grows. In order to improve recognition performance in very noisy environments, speech enhancement techniques must be useful. Enhancement schemes can improve the SNR, minimise the variance, and emphasise the important features of the signal, but at the expense of signal distortion. Minimising both signal distortion and noise, a signal with better features and lower variability is obtained.

In our earlier work [11], speech models were adapted to a signal enhanced by spectral subtraction using Parallel Model Compensation (PMC) in a scheme called SS-PMC. Although very good performance was demonstrated for the SS-PMC scheme, it does require a explicit word boundary detector and this limits its use in practice. In order to avoid this drawback, a Continuous Spectral Subtraction(CSS) scheme has been developed.

In this new system, speech models are adapted for a signal enhanced by this CSS scheme. It will be shown that the enhanced signal after being processed by the CSS can be represented by the addition of the noisy speech plus a correction term Δ^C in the linear domain. SS-PMC transforms the noise and speech model parameters from the cepstral domain to the linear domain, adds these parameters and the SS correction term, Δ, and then creates an adapted model by returning to the cepstral domain. Therefore, SS-PMC can be modified to compensate for the correction term Δ^C in the linear domain. This modified version of SS-PMC will be called the CSS-PMC method.

The results obtained by the CSS-PMC technique are very encouraging, showing that it is very effective to use adaptation techniques to compensate for the signal distortion which is a side effect of a CSS-based enhancement scheme.

Contents

1 Introduction

The practical application of speech recognition in the real world must deal with the serious environmental problem. The environment seriously degrades the performance of a speech recogniser designed on low noise input. This is why a lot of research effort has been given to this subject.

The environment is one of the causes of speech variations. It can cause speech-correlated noise, such as reverberation and reflection, and uncorrelated noise, either stationary or nonstationary. This variation affects the spectrum of the speech, that is the spectral peaks disappear, spurious spectral peaks appear, the spectral bandwidth changes, or other nonlinear transformations occur. Environment can also affect the way a person speaks, for example when a person speaks at low SNR not only the energy of the speech is increased, but the pitch and frequency are changed: this is known as Lombard effect.

Acoustic ambient noises are usually considered additive. Sources of this kind of noise are common, for example in the office environment, the office machinery such as typewriter or printer, disk and fans of personal computers or workstations, telephone ringing and background conversation; in car, the acoustic noise level due to engine, cooling fan, wind, tires and road are added and for this noise the SNR of the speech signal could drop -5 dB when the car cruises above 90 km/h, [9] [13]. Most noises are additive, this can be correlated or uncorrelated to speech. Noise also can be classified as stationary or nonstationary. The difficulty of tackling noise depends on the characteristics of the noise. This work deals with stationary uncorrelated additive noise.

Studies on the effect of noise on the performance of HMM based speech recognisers have shown that by training and testing, see Fig. 1, in the same conditions, the HMM recogniser achieves the best performance [7]. This is because the parameters of the HMMs "learn" the features of both the speech and the noise [3], [4] [13]. However, trying to build a database with models for all conditions is impractical. Therefore, one way to deal with this practical problem is to automatically adapt the clean speech models to the noise. The aim of any noise adaptation technique is to reach the recognition performance obtained when the HMMs are trained in the same environment.

One particular effective method of performing this adaptation is Parallel Model Combination [3]. In this case, the speech and noise parameters are transformed from the cepstral domain to the linear domain, the parameters are added, and finally they are returned to the cepstral domain. This adaptation method has been very successful in adapting clean models to noisy environments and very good results have been achieved, for example 93% word correct for Lynx Helicopter noise on a digit database at 0 dB [4] outperforming the results obtained by training and testing in the same noisy conditions.

As pointed out in [15], when adapting the speech models by adding noise, the level of noise that has to be added to effectively mask all background noise is rather high, specially at low SNR, and causes a significant reduction in accuracy. Therefore, recognition performance in very noisy environments is not completely solved by training in noise, and eventually these noise adaptation techniques must reach a limit. The variance becomes so large that feature overlapping begins to affect discrimination, and the problem becomes worse when the vocabulary grows. For example, it has been shown, [7] that the recognition performance can drop to 80% at 0 dB even when training and testing was on the same

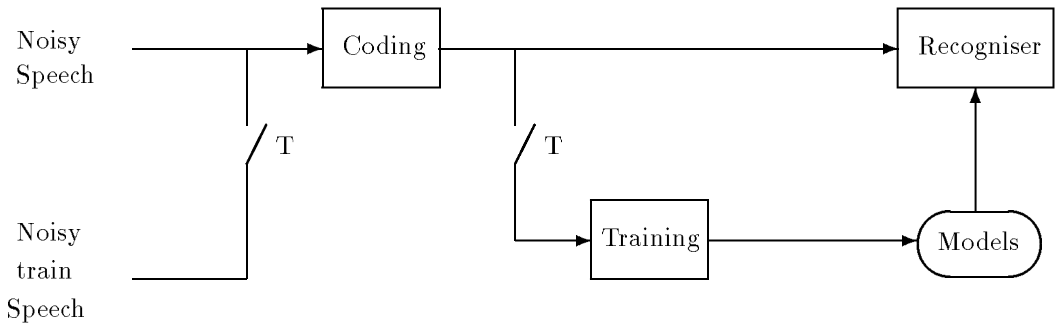

Figure 1: Training in Noise.

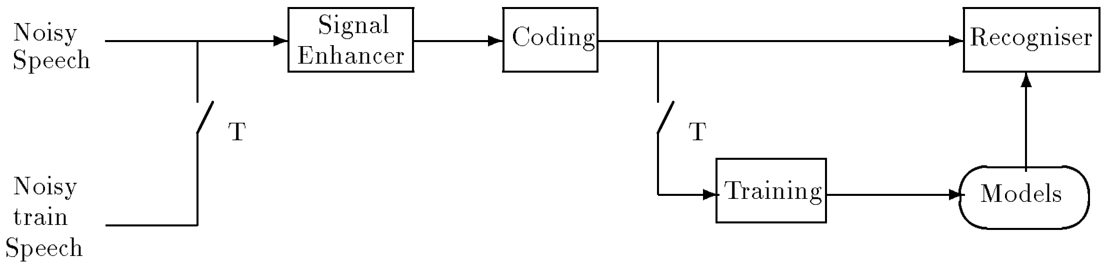

Figure 2: Training models on the enhanced speech.

noisy conditions. Therefore, in very noisy environments, or when the vocabulary grows, even training in noise is not enough to obtain good recognition performance. In order to improve recognition performance in very noisy environments, enhancement techniques are needed. These may attempt to improve the SNR, minimise the variance, or emphasis the main features of the interesting signal. However, all of these improvements are usually at the expense of signal distortion. If both signal distortion and noise are minimised, then a signal with better features and lower variability is obtained. However, if the good features of these enhancement techniques are to be exploited, then the speech models need to be compensated to the distorted signal.

Adapting the models to the enhanced signal should raise the achievable limit of recognition performance. In order to determine this limit, a test similar to training and testing on the same condition can be made. Training and testing on the enhanced speech signal is shown in Fig. 2. In this case, the enhancement algorithm affects both the training and testing database. Although, trying to create a model database for all environments is impractical this configuration gives an indication of the maximum performance which can be achieved with an enhancement scheme. In [11], it is shown that adapting(compensating) the HMMs to the signal distortion, for a SS scheme, this maximum is considerably higher than a scheme without HMMs adaptation. Moreover, the SS scheme and PMC were successfully combined to obtain an automatic method to adapt clean speech models to the enhanced speech signal. The experimental results of the SS-PMC scheme were very satisfactory showing that it is very effective to use adaptation techniques to compensate for the signal distortion which is a side effect of an SS-based enhancement scheme.

One drawback of Spectral Subtraction is that in order to obtain the noise estimate a word boundary detector is needed, and this word boundary estimator is a complex task specially for low SNR. In order to avoid the word boundary detector Continuous Spectral

Subtraction(CSS) is proposed, see Fig. 4. This CSS scheme obtains a low-frequency estimator of the time-frequency noisy speech and subtracts this estimator from the noisy speech. This low-frequency estimator is a function of the stationary noise and the low-frequency speech (smoothed speech).

In this work a HMM based recogniser with Gaussian state distributions is used. The Gaussian probabilistic density function, *pdf*, is completely modelled by the mean and the variance. Hence, in order to adapt the HMM, the mean and variance of each state have to be adapted as follows

$$\hat{\mu} = T_{\hat{\mu}}\{\mu\} \tag{1}$$

$$\hat{\Sigma} = T_{\hat{\Sigma}}\{\Sigma\} \tag{2}$$

where

$T_{\hat{\mu}}\{.\}$ is the transformation from the mean of the clean speech model to the enhanced speech model, and

$T_{\hat{\Sigma}}\{.\}$ is the transformation from the variance of the clean speech model to the enhanced speech model.

In this paper, the integration of Continuous Spectral Subtraction as enhancement technique, and PMC as adaptation technique is presented. Because PMC only adapts the HMMs to the noise, it has to be extended to include the effects of the distortion. It is shown in Sec. 2, that the enhanced speech spectrum can be represented in the linear domain by the spectrum of the noisy speech plus a correction term, Δ^C. Therefore, this extension turns out to be relatively straightforward.

The remainder of this paper is organised as follows. Chapter 2 presents the underlying theory and shows that the distorted speech (output) can be modelled as the addition in the linear domain of the noisy speech (input) plus a correction value. Chapter 3 reviews the details of the CSS-PMC algorithm. Finally, Chapter 4 presents experiments and results on the Noisex-92 database. Section 5 gives some comments and conclusions.

2 Theoretical Background

2.1 Spectral Subtraction

SS is attractive because in practice it has shown to be successful for both signal enhancement [2] [1] and speech recognition in noise [10]. This pre-processing scheme assumes zero-variance noise, but, because real noise is highly variable, especially for low SNR, SS sets a minimum positive value and subtracts an over-estimation of the noise. By using SS we obtain a signal with better features and lower variability. However, over-estimation and flooring makes Spectral Subtraction a non-linear compensator and hence the noise level is reduced at the expense of introducing distortion into the speech signal. Over-estimation can be seen as a way to make a trade-off between reducing the noise and distorting the speech, the optimum over-estimation value is when both the noise and the distortion is minimum. Experimental results show that the optimum over-estimation factor is about two for the magnitude spectrum.

There are different spectral subtraction schemes, but the work presented here is based on the scheme describe by Van Campernolle [14], see Fig. 3,

$$D(Y) = Y - \alpha\hat{N}$$

$$Y_D(Y) = \begin{cases} D(Y) & D(Y) > \beta Y \\ \beta Y & otherwise \end{cases} \tag{3}$$

where

$Y_D(Y)$ is the enhanced signal which is a distorted estimation of the speech S

Y is the either the power or the magnitude spectrum of the noisy speech

$\hat{N}$ is the noise estimator

α is an over-estimation factor, and

β is the spectral flooring.

SS needs a word boundary detector to estimate the noise, $\hat{N}$. The need for this word boundary detector presents serious implementation problems for low SNR. The next section describes a modification of the standard SS method such that no word boundary detector is needed.

2.2 Continuous Spectral Subtraction

A practical SS scheme as defined in 3 needs a word boundary detector but word boundary detector algorithms are not reliable, specially at low SNR. To avoid the need for this word boundary detector, an average from the last n frames of the time-frequency noisy signal can be computed and subtracted from the current frame of the time-frequency signal. This scheme will be refered as Continuous Spectral Subtraction (CSS).

Therefore, the CSS scheme, see Fig. 4, is as follows

$$D(Y) = Y - \alpha\hat{Y}^{(n)}$$

$$Y_D(Y) = \begin{cases} D(Y) & D(Y) > \beta Y \\ \beta Y & otherwise \end{cases} \tag{4}$$

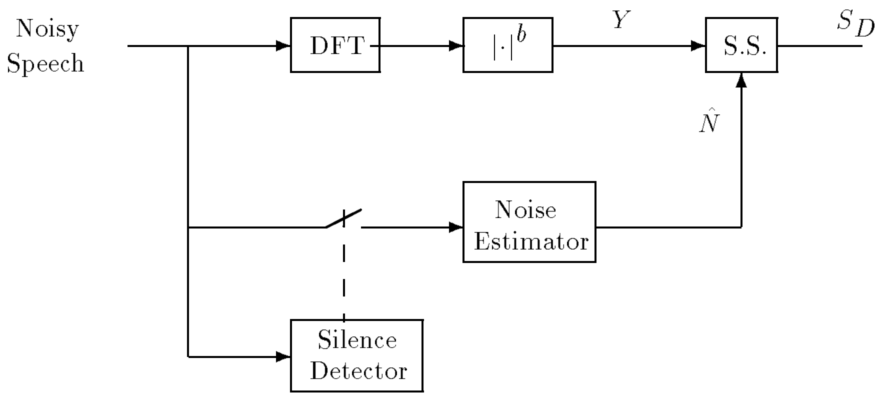

Figure 3: Spectral Subtraction, $b = 1$ gives magnitude subtraction, $b = 2$ gives power subtraction.

where

$Y_D(Y)$ is the enhanced signal which is a distorted estimation of the speech S

Y is the either the power or the magnitude spectrum of the actual noisy speech or background noise input frame

$\hat{Y}^{(n)}$ is the average of the last n frames of Y

α is an over-estimation factor, and

β is the spectral flooring.

In this scheme, the choice of n is very important. If we assume that the noisy speech is the addition of the speech plus the noise, that is $\hat{Y}^{(n)} = \hat{N}^{(n)} + \hat{S}^{(n)}$, then for very large n we lose the transitional information in S and the scheme is only useful for highly stationary noise, on the other hand for small n, we will distort the speech too much and a lot of information will be lost. The optimum value will depend on the vocabulary size of the speech recogniser, on the number of sounds in the language, on the kind of noise, etc. In order to simplify notation, in most of the cases $\hat{Y}$ will be used instead of $\hat{Y}^{(n)}$.

The aim is to adapt the speech models to the enhanced speech by CSS as in [11] is done for SS. The next section is concerned with how the mean and variance of the signal are affected when we process them by this Spectral Subtractor.

This scheme has some similarity to the Spectral Normalisation(SN) scheme proposed in [5], but there are important differences. First, in this work the spectral subtraction is performed in the linear domain to tackle additive noise, and SN performs the spectral subtraction in the log domain to tackle convolutive noise. Secondly, in this work n frames of the time-frequency noisy signal are averged in time. This average operation is an estimation of the mean when the noise is modelled as a HMM and, we can think of this as a low pass filtering operation in time on the time-frequency signal. Thirdly, SN does not peform over-estimation, perhaps because of the high signal distorion caused by this operation: this is an important difference because over-estimation is a key feature of SS. Fourthly, the SN scheme uses a fix flooring constant, in that case, flooring is not important because SN does not rely on over-estimation. Finally, the last difference is the compensation technique, in SN the filtering operation is affecting the speech signal so they compensate the models for this distortion by training with the same low-pass filter.

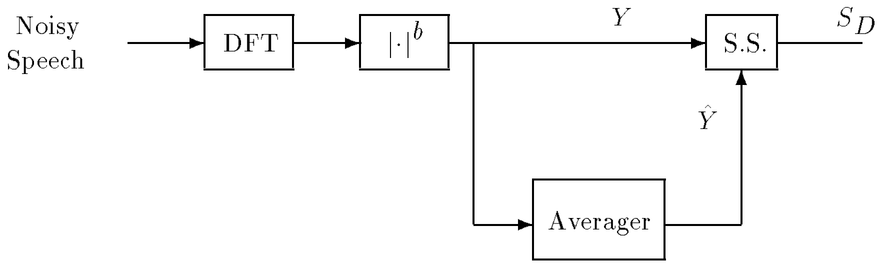

Figure 4: Continuous Spectral Subtraction, $b = 1$ gives magnitude subtraction, $b = 2$ gives power subtraction.

For CSS the over-estimation is adding another level of distortion caused not only for the speech but for the noise, therefore we need to compensate the models to this distortion the same way as SS-PMC [11] does.

Another scheme which can be related to the this work is the one by Hirsch *et al* [6]. They avoid using a boundary detector by time high-pass filtering of the time-frequency signal[1], such that the low frequencies are removed

$$Y_D = g(Y)$$

where $g(Y)$ is a high-pass linear filter. This scheme does not use a word boundary detector, but the filtering operation returns a speech signal with both less information and time distortion. But, a signal with better features is obtained if the noise reduction is higher than the distortion. Moreover, the distortion can easily be compensated by training the models with the same time high-pass filter applied to the clean speech. Because this scheme filters the noise, another drawback of this technique is that they do not use information about the noise to compensate the signal distortion.

2.3 Continuous Spectral Subtraction Analysis

At the CSS output an enhanced signal is obtained but it is distorted. To determine the impact of this distortion, its effect on the mean and variance of the signal need to be calculated. For simplicity, it is assumed that each frequency channel is statistically independent, hence it is only necessary to develop the theory for one-dimensional case.

The expected value of the enhanced speech, Y_D, is

$$\mathbf{E}[Y_D] = \int_{-\infty}^{\infty} Y_D(Y)dY$$

by using the SS scheme defined in eq. 4, we obtain

$$\mathbf{E}[Y_D] = \int_{a^C(\alpha,\beta,\hat{Y})}^{\infty} (Y - \alpha\hat{Y})P(Y)dY + \int_{-\infty}^{a^C(\alpha,\beta,\hat{Y})} \beta Y P(Y)dY$$

[1]The time domain of the time-frequency signal is what they call modulation frequency.

where $a^C(\alpha,\beta,\hat{Y}) = \frac{\alpha\hat{Y}}{1-\beta}$. Following as in [11], we obtain

$$\mathbf{E}[Y_D] = \mathbf{E}[Y] - \alpha\hat{Y} + (\beta-1)B(\alpha,\beta,\hat{Y},P(Y)) + \alpha\hat{Y}A(\alpha,\beta,\hat{Y},P(Y))$$

where

$$A(\alpha,\beta,P(Y)) = \int_{-\infty}^{a^C(\alpha,\beta,\hat{Y})} P(Y)dY \tag{5}$$

$$B(\alpha,\beta,\hat{Y},P(Y)) = \int_{-\infty}^{a^C(\alpha,\beta,\hat{Y})} YP(Y)dY \tag{6}$$

Defining

$$\Delta_\mu^C(\alpha,\beta,\hat{Y},P(Y)) = -\alpha\hat{Y} + (\beta-1)B(\alpha,\beta,\hat{Y},P(Y)) + \alpha\hat{Y}A(\alpha,\beta,\hat{Y},P(Y)) \tag{7}$$

we obtain

$$\mathbf{E}[Y_D] = \mathbf{E}[Y] + \Delta_\mu^C(\alpha,\beta,\hat{Y},P(Y)) \tag{8}$$

From this equation, it can be observed that the effect on the distorted signal can be expressed as the addition of the noisy speech and the correction values Δ_μ^C in the linear domain, and this function depends on the spectral subtraction parameters, α and β, the smoothed noisy speech, $\hat{Y}$, and the *pdf* $P(Y)$.

By comparing eq. 8 with eq. 1, we see that in this case $T_{\hat{\mu}}\{\mu\}$ is the addition in the linear domain of the expected value of the noisy signal, $\mathbf{E}[Y]$, plus a correction constant, Δ_μ^C. This is an interesting result, because it shows that, in the SS domain, we have the freedom to use any noisy adaptation algorithm and the correction Δ_μ^C can be compensated independently. For example, the PMC [3] adaptation technique assumes that the addition of the expected value of S, $\mathbf{E}[S]$, plus the expected value of N, $\mathbf{E}[N]$, is equal to the expected value of $\mathbf{E}[Y]$. This is an approximation which can be used in eq. 8, but we are not restricted to using this approximation. If more accurate methods of estimating $\mathbf{E}[Y]$ are developed, then they can be used instead.

Most of the algorithms for speech recognition in noise prefer not to compensate the variance, and it is common to use fixed variances. Fixed variance techniques replace the state variances with a single global variance which is obtained from all the words in the training database. Although, this is not a very elegant way to tackle the problem some good results have been obtained using this approach [12]. Parallel Model Combination(PMC), see Sec. 3, combines in the linear domain the speech and noise parameters, and transforms them back to the cepstral domain to obtain both mean and variance compensated models. In [4], Gales & Young show that this technique is more successful than the fixed variance technique. Therefore, we can either replace the state variance of the clean speech by a fixed variance or use the combination of the clean speech and noise variances generated by PMC.

As in the latter case, we can also extend the equations to compensate for the effect of the distortion on the variances. The effect of CSS on the variance can be calculated as follows

$$\mathbf{V}[Y_D] = \mathbf{E}[Y_D^2] - \{\mathbf{E}[Y_D]\}^2$$

because we have calculated $\mathbf{E}[Y_D]$, we only need to calculate $\mathbf{E}[Y_D^2]$, then

$$\mathbf{E}[Y_D^2] = \int_{a^C(\alpha,\beta,\hat{Y})}^{\infty} (Y - \alpha\hat{Y})^2 P(Y)dY + \int_{-\infty}^{a^C(\alpha,\beta,\hat{Y})} \beta^2 Y^2 P(Y)dY$$

where $a^C(\alpha,\beta,\hat{Y}) = \frac{\alpha\hat{Y}}{1-\beta}$. Proceeding as in [11] we obtain

$$\begin{aligned}\mathbf{V}[Y_D] \;\; &= \;\; \mathbf{V}[Y] + \{\mathbf{E}[Y]\}^2 - 2\alpha\hat{Y}\mathbf{E}[Y] + \alpha^2\hat{Y}^2 + 2\alpha\hat{Y}B(\alpha,\beta,\hat{Y},P(Y)) \\ &\quad -\alpha^2N^2A(\alpha,\beta,\hat{Y},P(Y)) + (\beta^2-1)C(\alpha,\beta,\hat{Y},P(Y)) - \{\mathbf{E}[Y_D]\}^2.\end{aligned}$$

where

$$C(\alpha,\beta,\hat{Y},P(Y)) = \int_{-\infty}^{a^C(\alpha,\beta,\hat{Y})} Y^2 P(Y)dY$$

and $A(\alpha,\beta,\hat{Y},P(Y))$ and $B(\alpha,\beta,\hat{Y},P(Y))$ are defined as 5 and 6, respectively.

Defining

$$\begin{aligned}\Delta_\Sigma^C \;\; &= \;\; \{\mathbf{E}[Y]\}^2 - 2\alpha\hat{Y}\mathbf{E}[Y] + \alpha^2\hat{Y}^2 + 2\alpha\hat{Y}B(\alpha,\beta,\hat{Y},P(Y)) - \alpha^2N^2A(\alpha,\beta,\hat{Y},P(Y)) \\ &\quad +(\beta^2-1)C(\alpha,\beta,\hat{Y},P(Y)) - \{\mathbf{E}[Y_D]\}^2 \qquad (9)\end{aligned}$$

we obtain

$$\mathbf{V}[Y_D] = \mathbf{V}[Y] + \Delta_\Sigma^C(\alpha,\beta,\hat{Y},P(Y)) \qquad (10)$$

By comparing eq. 10 with eq. 2, we can see that in this case $T_\Sigma\{\mathbf{V}[Y_D]\}$ is the addition of the variance of the noisy signal, $\mathbf{V}[Y]$, plus a correction on the CSS domain, $\Delta_\Sigma^C(\alpha,\beta,\hat{Y},P(Y))$.

Assuming the additivity of the speech and the noise in the linear domain and substituting it in eq. 8 we obtain

$$\mathbf{E}[Y_D] = \mathbf{E}[S] + \mathbf{E}[N] + \Delta_\mu^C(\alpha,\beta,\hat{Y},P(Y)) \qquad (11)$$

From this equation it can be seen how the clean speech is distorted by the CSS for the addition of $\mathbf{E}[N]$ and Δ_μ^C.

As in SS-PMC, compensating Δ^C is not directly implementable at signal processing stage since Δ_μ^C or Λ_μ^C depend on the spectrum of the underlying clean speech signal which is not known. In this case, the model means themselves provide the required estimates of the clean speech spectrum. Therefore, Δ^C is going to be implemented using a model adaptation algorithm.

In the adaptation algorithm, see eqs. 8 and 10, we also need $\hat{S}$ ($\hat{Y} \approx \hat{S} + \hat{N}$) to compensate for Δ_μ^C and Δ_Σ^C. Now, let us define the average operation over n frames of the speech as follows

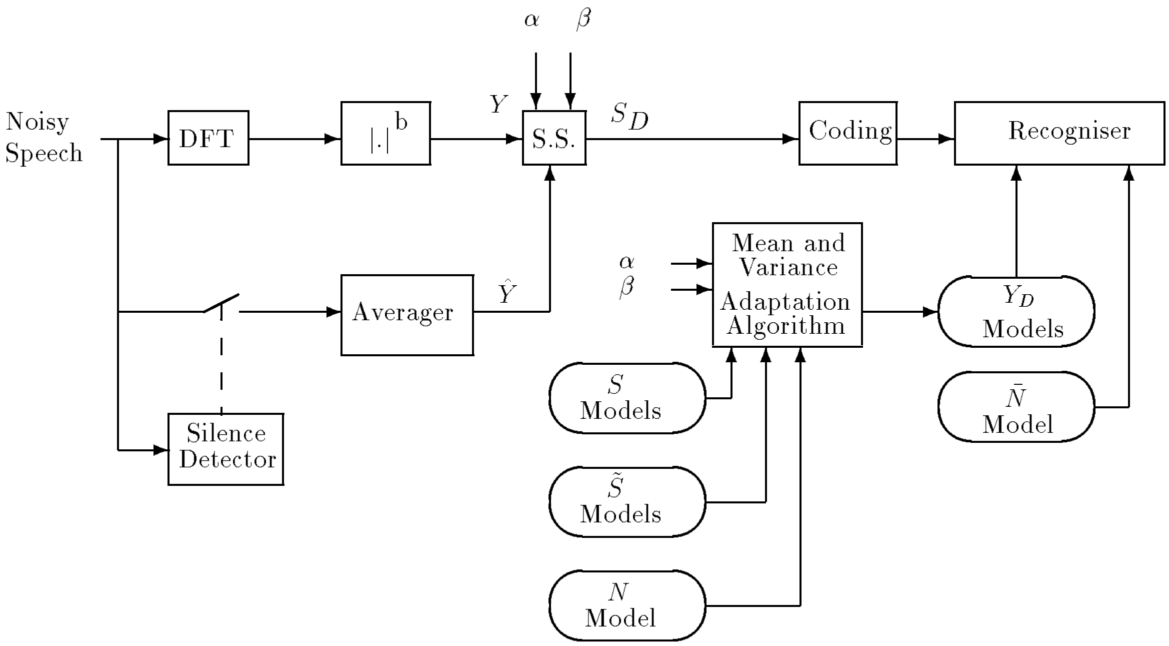

Figure 5: HMM compensation using the clean models, S, and the smoothed speech models, $\tilde{S}$, the noise model, N and the residual noise model, $\bar{N}$.

$$\hat{S}(jT,\omega) = \sum_{k=1}^{k=n-j} w(k-j)\frac{S((k-j)T,\omega)}{n}$$

where

$$w(k) = \begin{cases} 1 & 0 < k \leq n \\ 0 & otherwise \end{cases}$$

and

n is length of the window.

In a practical speech recogniser it is impossible to store all of this smoothed speech, therefore, we will create a model, $\tilde{S}$, for each smoothed speech recognition unit (i.e. word, phoneme, etc.). Hence, in the adaptive algorithm, the HMMs, $\tilde{S}$, are approximations to the smoothed speech, $\hat{S}$.

Using this approximation, when models are trained with the clean speech, $E[Y_D]$ and $V[Y_D]$, eqs. 8 and 10, can be solved with the configuration shown in Fig. 5. We can observe that this configuration needs the clean speech, S, the smoothed speech models, $\tilde{S}$, and the noise model, N, for the CSS-PMC algorithm, and for the recogniser, it needs the compensated models, Y_D, and the residual noise model, $\bar{N} = g(N)$, where $g(.)$ is the continuous spectral subraction function.

2.4 Training with the Pre-distorted Speech

Basically, eq. 8 and 10 compensate a noisy signal enhanced by CSS by modelling the speech signal distortion caused for both the over-estimation of the CSS and the noise. This problem can be split into two subproblems, first, the distortion caused by the CSS, and second the distortion caused by the noise. Hence, it is possible to solve the former subproblem by training the models on clean speech passed through the CSS process, and leaving the latter for mathematical modelling leading to a more accurate model adaptation.

Let us define $\mathbf{E}[S_D]$ as the clean speech after being preprocessed by the CSS. By a similar development, as in the last section we obtain

$$\mathbf{E}[S_D] = \mathbf{E}[S] + \Delta_\mu^C(\alpha, \beta, \hat{S}, P(S))$$

by substituting this equation in eq. 11 we obtain

$$\mathbf{E}[Y_D] = \mathbf{E}[S_D] + \mathbf{E}[N] + \Delta_\mu^C(\alpha, \beta, \hat{Y}, P(Y)) - \Delta_\mu^C(\alpha, \beta, \hat{S}, P(S))$$

Defining

$$\Lambda_\mu^C(\alpha, \beta, \hat{Y}, \hat{S}, P(S), P(Y)) = \Delta_\mu^C(\alpha, \beta, \hat{Y}, P(Y)) - \Delta_\mu^C(\alpha, \beta, \hat{S}, P(S)) \tag{12}$$

we obtain

$$\mathbf{E}[Y_D] = \mathbf{E}[S_D] + \mathbf{E}[N] + \Lambda_\mu^C(\alpha, \beta, \hat{Y}, \hat{S}, P(S), P(Y)) \tag{13}$$

Proceeding as before for the variance distortion, when the HMMs are trained with clean speech distorted by CSS we obtain

$$\mathbf{V}[Y_D] = \mathbf{V}[S_D] + \mathbf{V}[N] + \Lambda_\Sigma^C(\alpha, \beta, \hat{Y}, \hat{S}, P(Y), P(S)) \tag{14}$$

where

$$\Lambda_\Sigma^C(\alpha, \beta, \hat{Y}, \hat{S}, P(Y), P(S)) = \Delta_\Sigma^C(\alpha, \beta, \hat{Y}, P(Y)) - \Delta_\Sigma^C(\alpha, \beta, \hat{S}, P(S)) \tag{15}$$

Therefore, eq. 13 and 14 show how to compensate the models when these models are trained with the clean speech distorted by the continuous spectral subtraction scheme. The advantage of training with the distorted speech is that we reduce the number of approximations in the system.

When models are trained with the clean speech distorted by the spectral subtractor $E[Y_D]$ and $V[Y_D]$, eqs. 13 and 14, can be solved with the configuration shown in Fig. 6, observe that this configuration also needs the clean models, but these clean models are only used to calculate Λ_μ^C and Λ_Σ^C.

In theory, compensating at the training stage and compensating using an adaptive algorithm is the same. In practice there are differences. Firstly, when compensating at the training stage, SS works directly at the frame level which is more accurate than using the adaptive algorithm which compensates using the model parameters which spread the information over more than one frame, Secondly, our equations are inherently approximate anyway. Therefore, if we can compensate during training, we would expect better results.

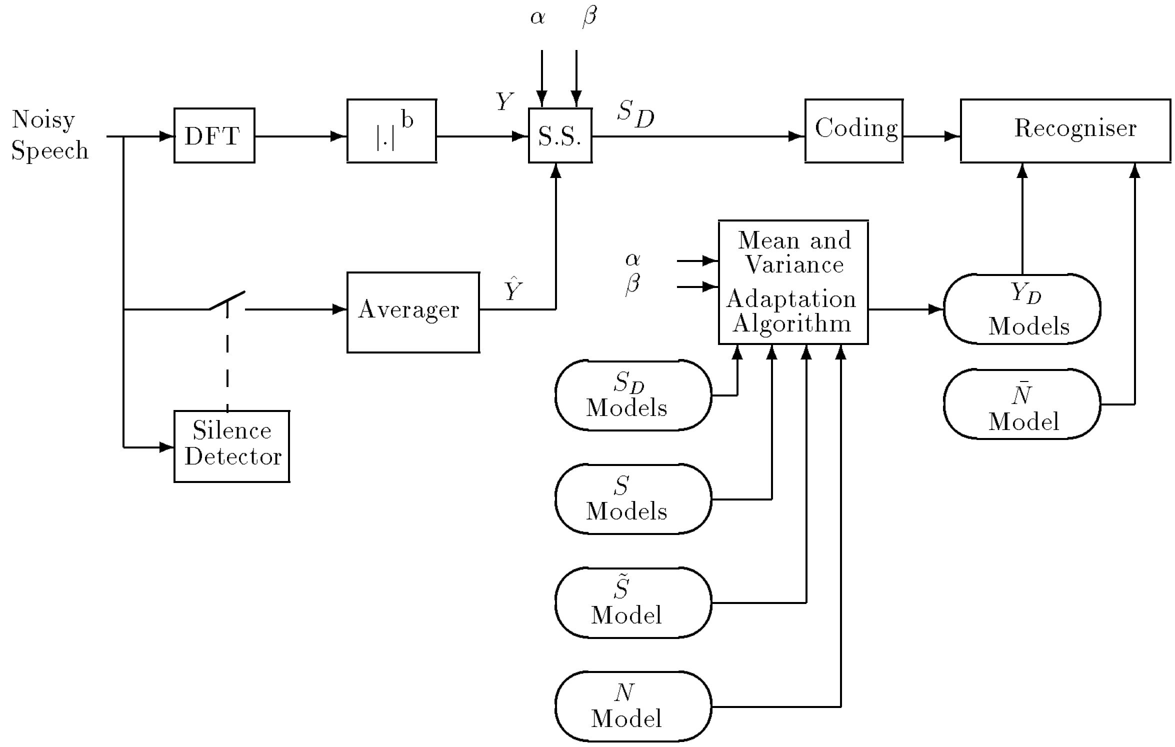

Figure 6: HMM compensation using the SS distorted clean speech Models, S_D, the clean speech models, S, the smoothed speech models, $\tilde{S}$, the noise model, N and the residual noise model, $\bar{N}$.

2.5 Solutions of the Integrals A, B and C

A potential problem of the compensation equation are the solution of the integrals $A(\alpha, \beta, P(Y))$, $B(\alpha, \beta, P(Y))$ and $C(\alpha, \beta, P(Y))$ since the transformation can yield on untractable *pdf* P(Y).

By assuming that Y has a log normal pdf, P(Y) is completely defined by ξ and ψ, therefore, we can rewrite $A(\alpha, \beta, P(Y))$, $B(\alpha, \beta, \hat{Y}, P(Y))$ and $C(a^C(\alpha, \beta), P(Y))$ as $A(\alpha, \beta, \hat{Y}, \xi, \psi)$, $B(\alpha, \beta, \hat{Y}, \xi, \psi)$ and $C(\alpha, \beta, \hat{Y}, \xi, \psi)$. Solving this integral as in [11], we obtain

$$A(\alpha, \beta, \hat{Y}, \xi, \psi) = G\left(\frac{ln(a^C(\alpha, \beta, \hat{Y})) - \xi}{\psi}\right) \tag{16}$$

$$B(\alpha, \beta, \hat{Y}, \xi, \psi) = \mathbf{E}[Y]G\left(\frac{ln(a^C(\alpha, \beta, \hat{Y})) - (\xi + \psi^2)}{\psi}\right) \tag{17}$$

$$C(\alpha, \beta, \hat{Y}, \xi, \psi) = \mathbf{E}[Y^2]G\left(\frac{ln(a^C(\alpha, \beta, \hat{(Y)})) - (\xi + 2\psi^2)}{\psi}\right) \tag{18}$$

where
$\mathbf{E}[Y] = e^{\xi + \psi^2/2}$
$\mathbf{E}[Y^2] = e^{2(\xi + \psi^2)}$

The cumulative function $G(x)$ does not have an exact solution but for small values, it can be approximated by a look-up table and for large values, it can can be approximated by $G(x) = \frac{1}{x}e^{-x^2}$ or by $G(x) = (\frac{1}{x} + \frac{1}{x^3})e^{-x^2}$. Moreover, $\mathcal{N}(0,1)$ is a symmetric function, hence the size of the table look-up can be reduced by half using the following equation [8]

$$G(x) = 1 - G(-x).$$

The above expressions for $A(\alpha, \beta, \hat{Y}, \xi, \psi)$, $B(\alpha, \beta, \hat{Y}, \xi, \psi)$ and $C(\alpha, \beta, \hat{Y}, \xi, \psi)$ allow the required correction factors Δ_μ^C and Δ_Σ^C to be calculated in terms of the expected values of Y and Y^2 and the parameters ξ and ϵ of the log normal distirbution $P(Y)$. As shown in the appendix,

$$\mathbf{E}[Y] = e^{\xi + \psi^2/2}$$

and

$$\mathbf{E}[Y^2] = e^{2(\xi + \psi^2)}$$

from which it is straightforward to show that

$$\psi^2 = ln(\mathbf{V}[Y] + \{\mathbf{E}[Y]\}^2) - 2ln(\{\mathbf{E}[Y]\}) \tag{19}$$

and

$$\xi = ln(\mathbf{E}[Y]) - \psi^2/2 \tag{20}$$

replacing eq. 19 in eq. 20 we obtain

$$\xi = -0.5ln(\mathbf{V}[Y] + \{\mathbf{E}[Y]\}^2) + 2ln(\{\mathbf{E}[Y]\}^2)$$

3 The CSS-PMC Algorithm

In the previous section, it has been shown that assuming the spectral distributions in the linear domain are log normal then it is possible to calculate either the correction values Δ_{μ}^{C} and Δ_{Σ}^{C} or Λ_{μ}^{C} and Λ_{Σ}^{C}. Δ_{μ} and Δ_{Σ} are computed when the clean models, S are going to be adapted to the distorted speech, see eq. 8 and 10, or the correction values Λ_{μ}^{C} and Λ_{Σ}^{C} , when the noisy speech models are adapted, see eq. 12 and eq. 15, to the distorted speech. The assumptions of log normality in the linear domain cover the main representations used for speech recognition. Recognition may use either the log filter bank parameters or a cepstral representation. However, these are linked by a linear transformation and both may be assumed to be log normal.

The calculation of Δ_{μ}^{C} and Δ_{Σ}^{C} or Λ_{μ}^{C} and Λ_{Σ}^{C} requires the spectral subtraction parameters, α and β, the *pdf* P(Y) which in this case is completely defined by ξ_Y and ψ_Y,the *pdf* P(S) which in this case is completely defined by ξ_S and ψ_S, an estimate of the noise, N, and an estimate of the clean speech, S, and an estimate of the smoothed speech, $\tilde{S}$. All these model estimates are in the linear domain.

As noted earlier, the PMC technique developed by Gales & Young [3] allows the latter expectations to be calculated by assuming that the HMM Gaussian output distributions characterise N and S in the log domain. These parameters are then mapped into the linear domain where additivity is assumed to hold, thereby allowing $E[Y]$ and $V[Y]$ to be calculated. Given the theory developed in section 2, it is straightforward to extend this PMC approach to the CSS case. As shown in Fig. 7, the basic PMC framework is unaltered save for the inclusion of the Δ^C and Λ^Cfactors in the linear domain. Thus, the overall steps in the CSS-PMC scheme are

1. Transform the cepstral (or log) means and variances back into the linear domain as in standard PMC.

2. Calculate $E[Y]$ and $E[Y^2]$ assuming additivity of the speech and noise.

3. Calculate ξ_Y, ψ_Y, ξ_S and ψ_S given the noisy speech mean and variance, $E[Y]$ and $V[Y]$, and the clean speech mean and variance, $E[S]$ and $V[S]$, see eq. 19 and 20.

4. Calculate Δ_{μ}^{C} and Δ_{Σ}^{C} and adjust the means and variances, or calculate, when the speech is pre-distorted S_D, Λ_{μ}^{C} and Λ_{Σ}^{C} and adjust the means and variances.

5. Transform the compensated means and variances back to the cepstral (or log) domain as in standard PMC.

Fig. 9 shows the block-diagram representation for the Δ_{μ}^{C} and Δ_{Σ}^{C} calculation can be calculated by a substraction of Δs, therefore this block-diagram is also used to calculate Λ_{μ} and Λ_{Σ}. In order to make the explanation of the algorithm clear, the problem is divided into three general steps. Each step is completed with intermediate steps. Fig. 9 shows the general steps separated by thicker lines. The calculations are based on the equations developed in Sec. 2, specifically, on eqs. 16, 17, 18, 7, 9, 20 and 19.

The realization of this block diagram in a sequential computer can be done has follows, given the mean and variance, μ and Σ, and the parameters of the SS, α and β, as input parameters, the general steps to solve are:

- Obtain a, ξ and ψ.
- Calculate $A(a,\xi,\psi)$, $B(a,\xi,\psi)$ and $C(a,\xi,\psi)$
- Calculate Δ_μ^C and Δ_Σ^C

The detailed intermediate steps should be clear from Fig. 9.

Once the correction factors have been applied we return to the standard PMC algorithm and transform from the linear domain back to the cepstral (or log) domain.

Δ_μ^C and Δ_Σ^C need estimation of the average noisy speech $\hat{Y}$, and Λ_μ^C and Λ_Σ^C also needs the average clean speech, $\hat{S}$. Fig. 8 shows how to obtain these values when the smoothed models, $\hat{S}$, and the noise model, $\hat{N}$, are trained in the cepstral domain.

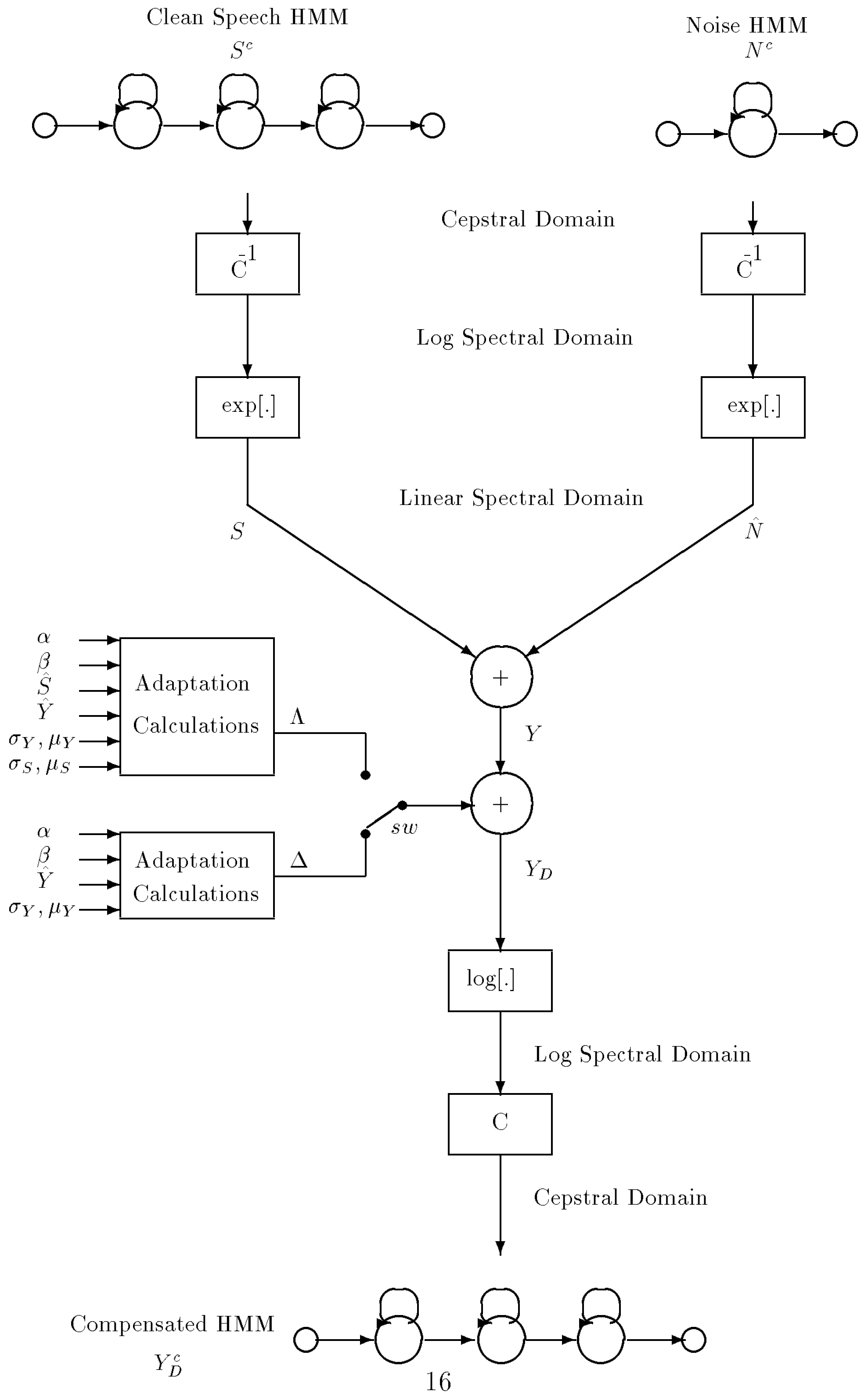

Figure 7: Block-diagram representation of CSS-PMC. sw connects Δ when the cleans speech models are adapted, and sw connects Λ when CSS distorted clean speech models are adapted.

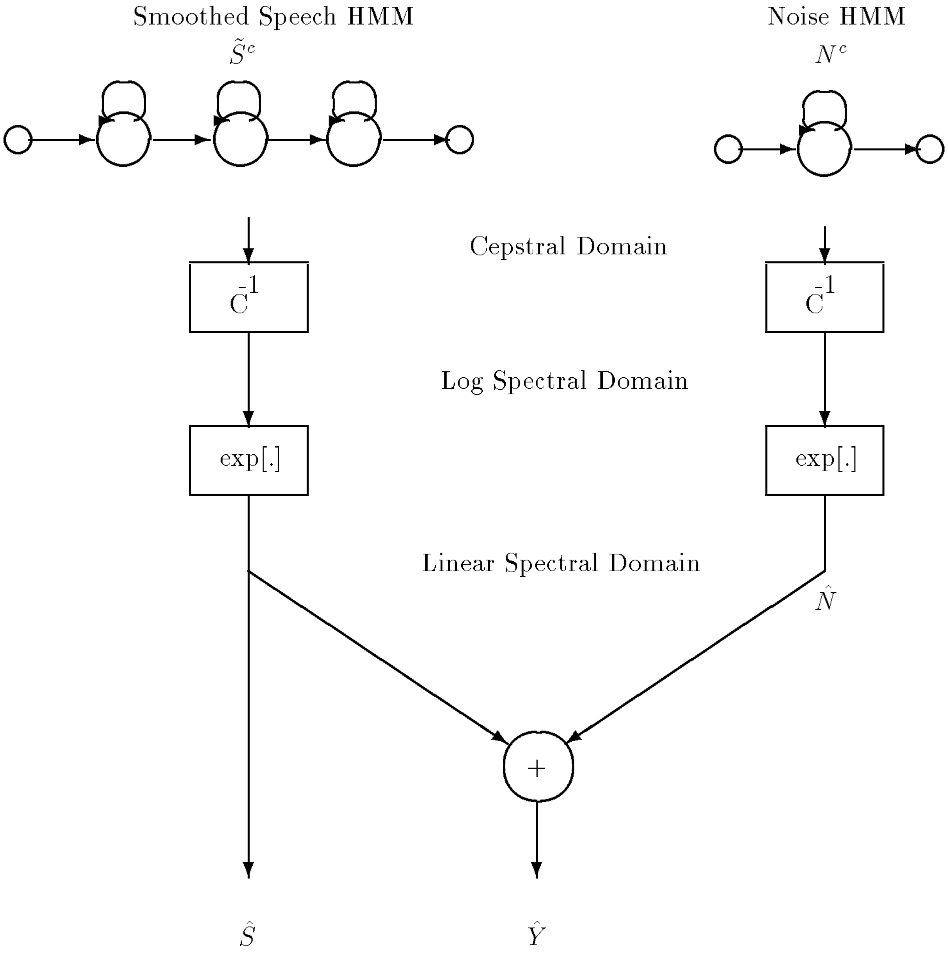

Figure 8: $\hat{Y}$ and $\hat{S}$ calculated from HMMs trained with the MFCCs.

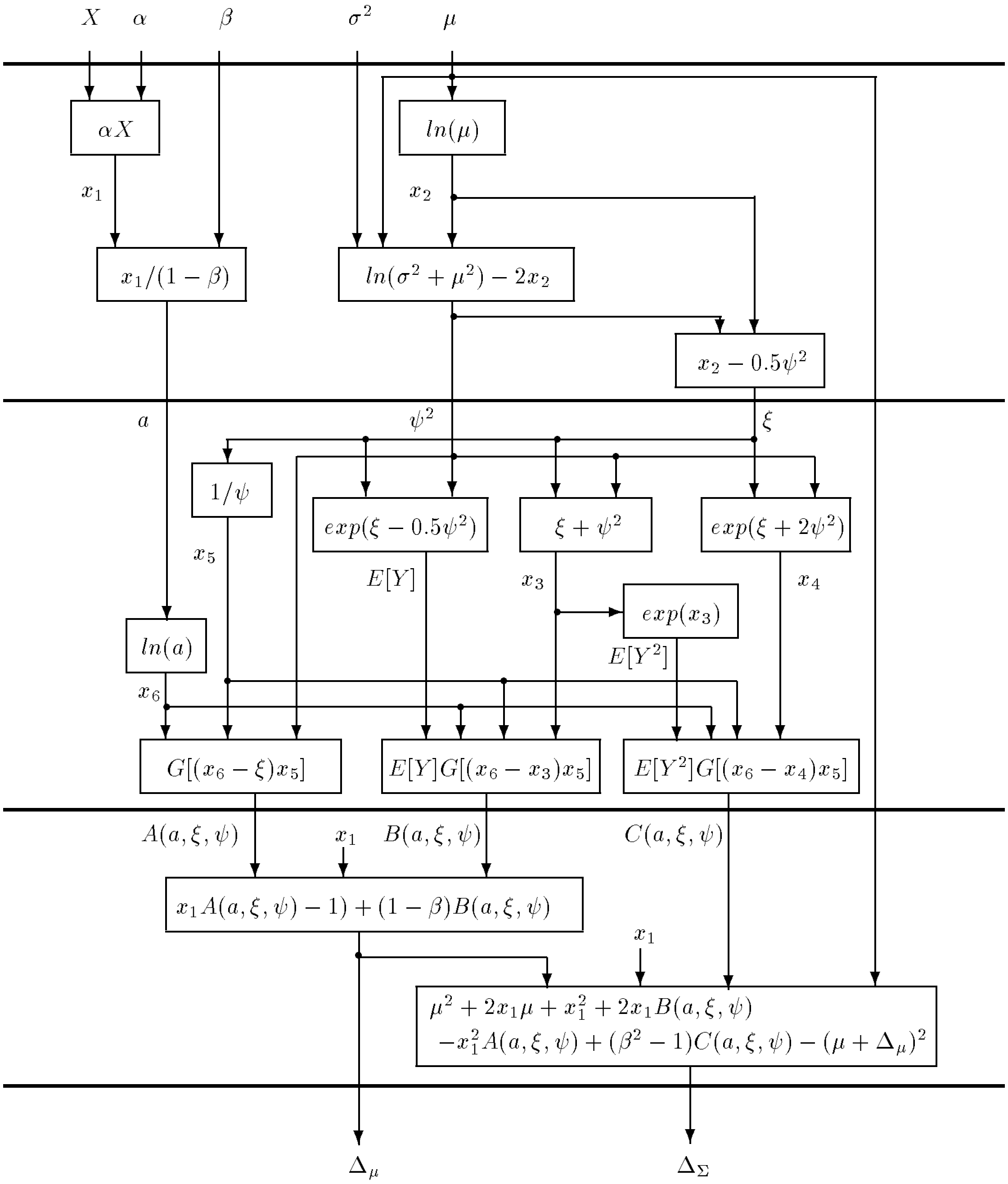

Figure 9: Block-diagram representation for Δ_μ and Δ_Σ, set $\sigma=\sigma_Y$, $\mu=\mu_Y$ and $X=\hat{Y}$ to calculate $\Delta^C_{\hat{Y}}$, and $\sigma=\sigma_S$, $\mu=\mu_S$ and $X=\hat{S}$ to calculate $\Delta^C_{\hat{S}}$

4 Experiments and Results

The CSS-PMC technique described in the previous sections has been evaluated using the Noisex-92 database. This database was created by artificially adding noise to clean speech, therefore, it does not exhibit the Lombard effect. This database contains digits, 100 training utterances recorded in silence and 100 testing utterances with different noises, e.g. car and helicopter, at levels in the range +18 dB to -6 dB. Lynx helicopter, car and F16 were the noises used for our experiments. These noises were chosen because the scheme assumes stationary noise, and these are more or less stationary. 0 dB and -6 dB were the levels used to test since these are the noisiest conditions. This database has a male speaker and a female speaker, using their native language, English. The male speaker was used for our experiments.

The baseline recogniser used 10 state word-based HMMs, with 8 emitting states and single Gaussian diagonal covariance matrix output probability distributions. The speech was pre-processed using a 25 ms Hamming window, and then parameterised into the first 14 cepstral coefficients obtained from the power spectrum (PMFCC). The speech signal was sampled every 10 ms. The clean speech model variances for the baseline recogniser were replaced by a fixed-variance. This fixed-variance was obtained from all the training data.

For the CSS-PMC scheme, a HMM for each digit was trained using either the clean speech, S, or the pre-distorted clean speech, S_D. The noise, N, and residual noise models, $\bar{N}$, used a single emitting state model and it was trained on all the available noise data. This noise model uses 1 emitting state and single gaussian diagonal covariance output probability distribution. The topology for all models was left-right with no skips and diagonal covariances were assumed throughout. For each frame, a set of 15 MFCC coefficients were computed. The zeroth cepstral coefficient is computed and stored since it is needed in the CSS-PMC mapping procedure. However, it is subsequently dropped in the actual recognition process. CSS-PMC compensates for the means as discussed in Sec. 3. In order to obtain wider variances, the term Δ_Σ was set to zero. Another way to keep the variance wider is using fixed-variance, but no experiments were tried using fixed-variance.

Recognition used a standard connected word Viterbi decoder constrained by a syntax consisting of silence followed by a digit in a loop. Thus, no explicit end-point detector was used and insertion/deletion errors occurred as well as classifications errors. The results are in terms of percentage(%) accuracy where for N tokens, S substitution errors, D deletion errors and I insertion errors, accuracy is calculated as $[(N-S-D-I)/N]100\%$. The error counts themselves were calculated by using a DP string matching algorithm between the recognised digit sequence and the reference transcription. Since NOISEX data is synthetic, the gain matching term g can be set exactly. Hence for all experiments here $g = 1$ was used. All the training and testing used version 1.4 of the portable HTK HMM toolkit [16], with suitable extensions to perform CSS-PMC.

The continuous spectral subtraction scheme discribed in section 2 is used for the signal enhancement. The parameter α was varied from 1.0 to 2.4, and β was fixed at 0.1. Some experiments varying β were performed, and it was found that $\beta = 0.1$ has a good behaviour and values around this have similar recognition performance. No attempt was made to apply time smoothing.

As was discussed in Sec. 1, an upper limit on recognition performance can be estimated by applying the enhancement technique to the training noise data, see Fig. 2, such that the models "learn" the features of the enhanced signal. Fig. 10, shows the results on the Lynx noise when PCSS-PMFCC [2] preprocessing are applied for different values of α. Fig. 10 also shows the result when the spectral subtractor is applied before the filterbank, we will refer this latter preprocessing as the bPCSS-PMFCC. The theory developed in Sec. 2 assumes that the spectral subtraction is applied after filterbank processing, but assuming that the filterbank is some kind of frequency smoother and that the smoother does not have a significant effect on the subtracted signal, hence the theory of Sec. 2 can also be applied to the bPCSS-PMFCC case.

Fig. 10 indicates the best performance of each of the schemes for Lynx, Car and F16 noises at 0 and -6 dB SNR. For all cases, the introduction of the speech enhancer improves the recognition performance. The best recognition performance, when training and testing in the same noise environment without any enhancement corresponds to $\alpha = 0$. For example, Fig. 10 shows that when training and testing in the noise conditions for the Lynx noise at -6 dB, 40% recognition performance is obtained for PCSS-PMFCC. This represents the best performance that we can expect to obtain with an ideal model noise adaptation algorithm. However, when the enhacement scheme is included, the best recognition performance goes up to 60% ($1.4 \leq \alpha \leq 2.4$). Hence, by adding the enhancement scheme, the upper limit of recognition performance is increased. Therefore, compensating for the distortion of the CSS scheme, in the best case it should be possible to reach this improved recognition performance. It can be observed that a similar improvement is obtained by bPCSS-PMFCC.

In general, for these noises at 0 and -6 dB SNR, for any value of α, when with CSS is used after the filterbank better results are obtained. Although this suggest the use of CSS after the filterbank preprocessing, it is useful to see how well both PCSS-PMFCC and bPCSS-PMFCC enhancement schemes work with our compensation algorithm.

It can be observed that these schemes reach maximum recognition performance for some values of α (e.g. maximum recognition performance for PCSS-PFCC is at $1.4 \leq \alpha \leq 2.0$ for Lynx noise at -6 dB). These α represent the values of α at which maximise noise reduction and minimise speech information loss, and hence give maximum discrimination. It is expected that these values will be dependent on the SNR, the noise variance and the vocabulary size.

To test the CSS-PMC scheme, the models were trained with either the clean speech, S, or the clean speech processed by the CSS, S_D, decribed in Sec. 2. If the models were trained with the clean speech, S, then the adaptation algorithm compensate for Δ^C, otherwise for Λ^C. For pre-processing PCSS was used, before and after the filter bank, and as discussed above, in order to keep the variance wider, we only compensated for the means. In order to compensate for all the assumptions and approximations, Δ_μ is weighted by γ. The experiments show that values of γ around 0.7 are good for 0 dB SNR, and around 0.6 for -6 dB SNR.

Fig. 11 and 12 show the results for 0 and -6 dB SNR for Lynx, Car and F16 noises

[2] PCSS means that CSS is applied in the power domain and PMFCC means that MFCCs are obtained from the power spectrum.

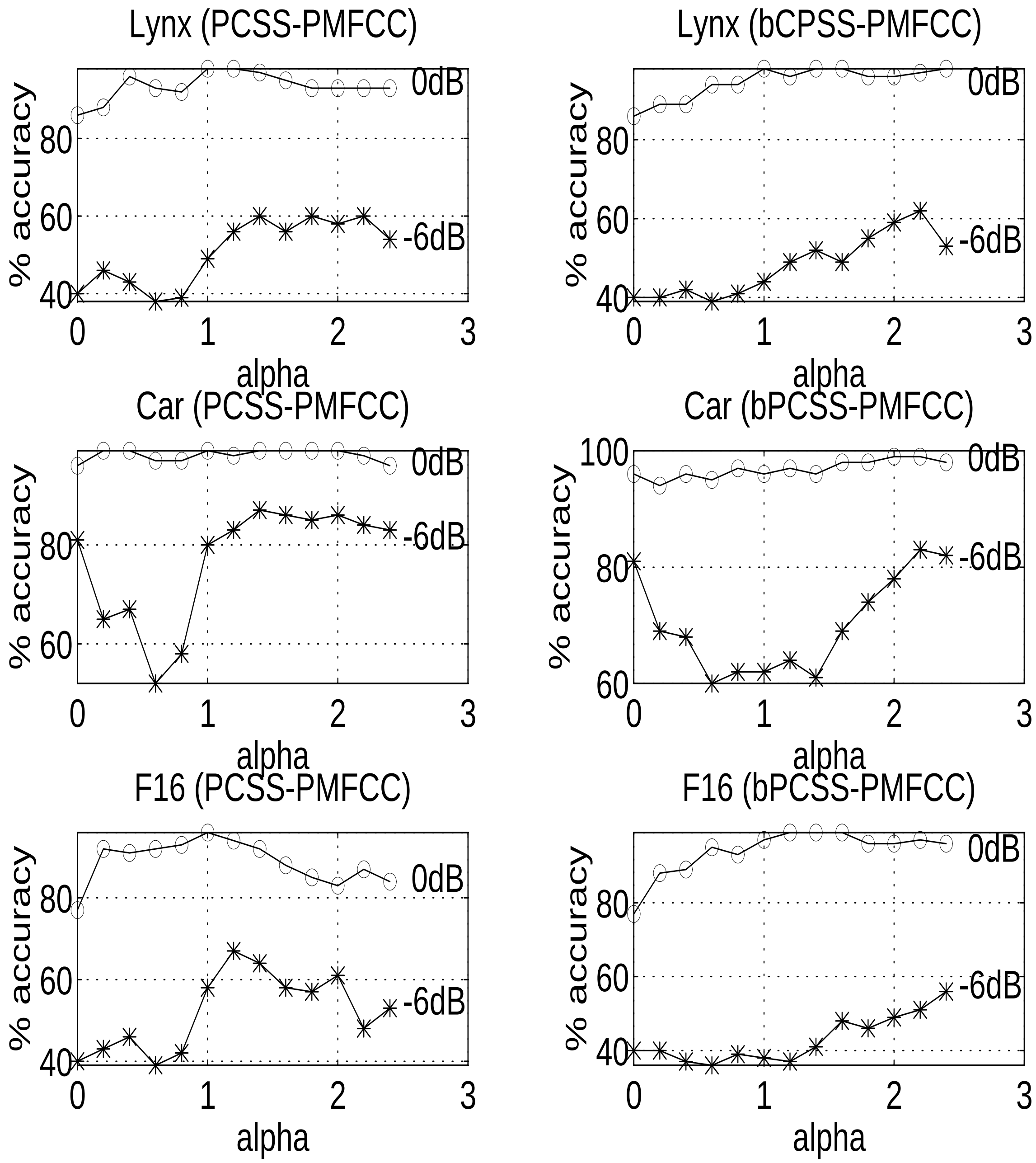

Figure 10: Upper limit recognition performance for Lynx, Car and F16 at 0 and -6 dB SNR for PCSS-PMFCC and bPCSS-PMFCC. $\hat{Y}$ was obtained using a window length of 20, this is $n = 20$.

for bPCSS-PMFCC and PCSS-PMFCC, respectively. This figure shows the performance for $\gamma = 0.7$, $\gamma = 0.6$, $\gamma = 0.5$ and the upper limit indicated by "best". From these figures, we can observe that in most of the cases the recognition performance of the CSS-PMC reachs the upper limit. In general the maximum recognition performance was obtained for $1.0 \leq \alpha \leq 1.6$.

Table 1 sumarises the baseline results for a standard fixed variance HMM based recognition system, PSS and bPSS when ideal word detector is used. The noise estimator is obtained from the average of the twenty spectral samples before the words starts. These results were obtained for $0.0 \leq \alpha \leq 3.0$ and $\beta = 0.1$. As noted earlier, the value of β was fixed at 0.1 from the outset. This may not therefore represent an optimum setting. Table 2 summarises the accuracy obtained for the standard PMC algorithm, see [4], for Lynx, Car and F16.

Table 3 shows the results when CSS-PMC compensate the clean models using Δ^C. The noisy estimator, $\hat{Y}$, is obtained using a window length of the twenty spectral samples, this is $n = 20$. Very good results were obtained when the PSS was applied before the filterbank, and when SS was applied at filterbank level the results were only significantly better than PMC results for the F16 noise.

Table 4 sumarises the performance obtained using the CSS-PMC scheme for PCSS-PMFCC and bPCSS-PMFCC, when training with the distroted speech. Again, very good results were obtained when CSS was performed before the filterbank, and slightly better results than PMC were obtained when SS was applied at filterbank level.

Finally, table 5 and 6 shows how the recognition accuracy change for Lynx, Car and F16 noises when the average window length is varied from $n = 10$ frames, 100 ms, to $n = 50$ frames, 500 ms., and training using the distorted speech for PCSS-PMFCC and PCSS-PMFCC. We can observe that the recognition performance is quite high even for $n = 20$ frames (200 ms), which means the effectivity of compensating the speech distortion. Again, this have two important advanages, first the CSS can be used for quasi-stationary noise, and this technique can be used for smaller recognition units such as phonemes, therefore, continuous speech recognition task are possible.

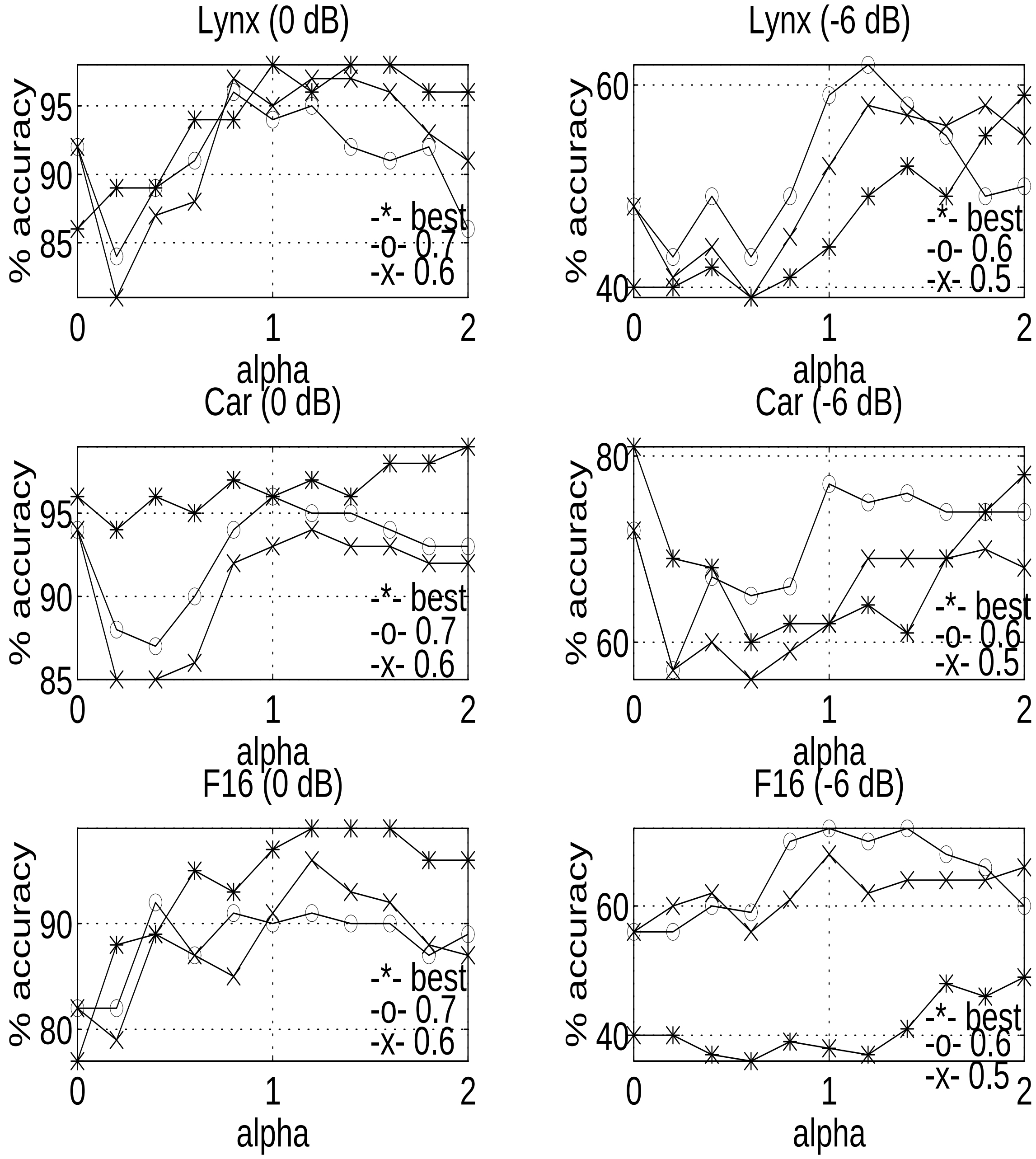

Figure 11: Recognition performance for Lynx, Car or F16 at 0 or -6 dB SNR for bPCSS-PMFCC when training with speech distorted by CSS and compensating Λ^C, (n=20), weighted for $\gamma = 0.7$, $\gamma = 0.6$ and $\gamma = 0.5$. This figure also show the upper limit indicated by “best”.

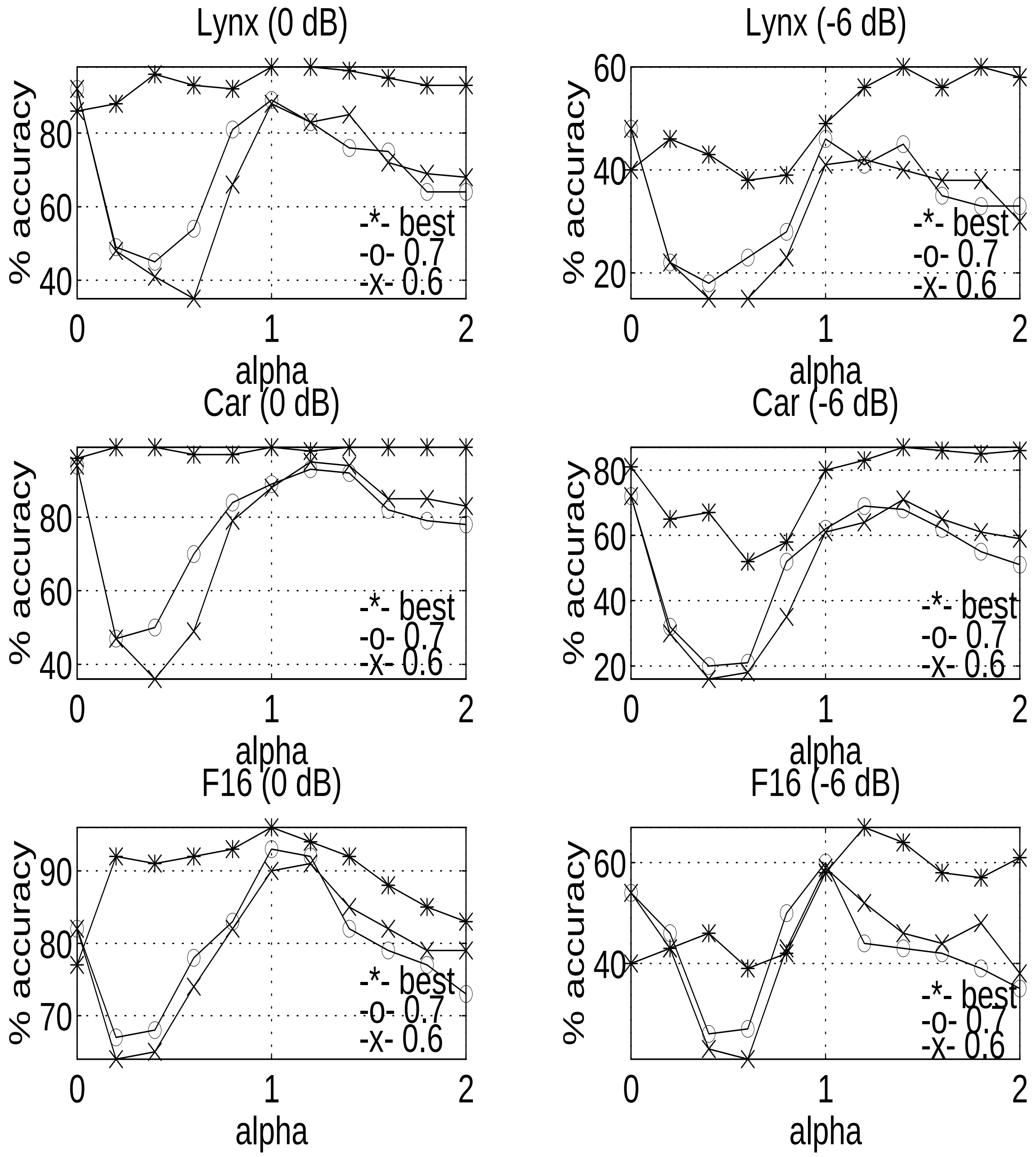

Figure 12: Recognition performance for Lynx, Car or F16 at 0 or -6 dB SNR for PCSS-PMFCC when training with speech distorted by CSS and compensating Λ^C, (n=20), weighted for $\gamma = 0.7$ and $\gamma = 0.6$. This figure also show the upper limit indicated by "best".

	SNR(dB)	Lynx (%)	Car (%)	F16 (%)
Std. HMM	0	32	42	42
	-6	20	16	12
PSS-PMFCC	0	70	81	60
(SS)	-6	46	56	43
bPSS-PMFCC	0	78	83	64
(SS)	-6	50	53	46
PSS-PMFCC	0	97	99	99
(SS-PMC)	-6	82	92	85
bPSS-PMFCC	0	100	100	99
(SS-PMC)	-6	81	92	81

Table 1: Correct words baseline results ($0 \leq \alpha \leq 3.0$ and $\beta = 0.1$).

	SNR(dB)	Lynx (%)	Car (%)	F16 (%)
PMC	0	92	94	82
	-6	48	72	54

Table 2: Accuracy baseline results ($0 \leq \alpha \leq 3.0$ and $\beta = 0.1$).

Method	SNR (dB)	Lynx (%)	Car (%)	F16 (%)
bPCSS-PMFCC	0	96	99	98
	-6	65	75	75
PCSS-PMFCC	0	81	94	97
	-6	44	73	67

Table 3: Accuracy for CSS-PMC using the clean speech models, S, and window length of 20 frames.

Method	SNR (dB	Lynx (%)	Car (%)	F16 (%)
bPCSS-PMFCC	0	97	96	96
	-6	62	78	72
PCSS-PMFCC	0	89	96	93
	-6	46	71	60

Table 4: Accuracy for CSS-PMC using the distorted speech models, S_D, and window length of 20 frames.

	SNR	n=10	n=20	n=30	n=40	n=50
Lynx	0	95	97	97	97	96
	-6	61	62	65	64	64
Car	0	92	96	97	95	94
	-6	73	78	78	80	77
F16	0	91	96	94	95	96
	-6	69	72	70	73	72

Table 5: Accuracy for Car, Lynx and F16 using bPCSS-PMFCC when varying the average window lengths, when using the distorted models, S_D.

	SNR	n=10	n=20	n=30	n=40	n=50
Lynx	0	86	89	87	87	87
	-6	46	46	47	48	47
Car	0	91	96	95	94	93
	-6	67	71	74	74	79
F16	0	87	93	92	93	90
	-6	57	60	60	61	62

Table 6: Accuracy for Car, Lynx and F16 using PCSS-PMFCC when varying the average window lengths, when using the distorted models, S_D.

5 Conclusions

By enhancing noisy speech we obtain a signal with better features and less variability, at the expense of signal distortion. In this work, we have proposed the use of adaptation techniques to compensate the clean speech models to match them to the distorted speech produced by a signal enhancer.

The experiments show that using an ideal model adapting algorithm applied to enhanced signal, we obtain higher potential performance than is possible using an adaptation scheme alone. These results encourage us to study the best way to adapt the distortion of the signal caused by the enhancement techniques.

We developed the necessary compensation for a Continuous Spectral Subtractor enhancer and we tested on the NOISEX-92 database. First, we showed that the effect of the CSS on noisy speech can be separated in the linear domain into the effect of the noise on the speech and the distortion caused by the CSS, Δ^C or Λ^C. Because the standard PMC algorithm returns the model to the linear domain, compensating the distortion is easily achieved on this linear domain by modifying the PMC adaptation technique to compensate for the enhanced signal. We refered to this modified approach as the CSS-PMC scheme.

Second, the theory behind of our adaptation algorithm is based on several assumptions and approximations. In order to compensate for these approximations, we weight the distortion compensation by a γ factor, and the best performance was obtained when this factor was around 0.7 for 0 dB SNR, and when this factor was around 0.5 for -6 dB SNR. Finally, we tested CSS-PMC, and the results we obtained were comparable or better than the results of an ideal adaptation algorithm.

The compensating equations Δ^C and Λ^C were developed assuming a log normal distribution of the signal before being preprocessed by the CSS. Δ^C is used when the models are trained using the clean speech, and Λ^C is used when the models are trained with the distorted speech.

The experimental results show that for 0 dB the recognition performance is slightly better when the models are trained with the clean speech, however the experimental results for -6 dB show slightly better recognition performance when the models are trained using the pre-distorted speech.

It was also shown that the CSS scheme obtains good recognition performance even for small window lengths, such as 200 ms, allowing the cancellation of quasi-stationary noises.

The Noisex-2 database artificially adds real noise to the clean speech, hence there is no Lombard effect. If we wish to test on real noisy speech, we will also need to compensate for this effect.

6 Acknowledgment

Thanks to CONACyT for the financial support for this research and to Mark Gales for many fruitful discussions on this work.

pwd

A Appendix.

By definition,

$$\mathbf{B}(a,\xi,\psi)=\int_{-\infty}^{a}Y\frac{1}{Y\sqrt{2\pi}\psi}e^{-\frac{1}{2\psi^2}(ln(Y)-\xi)^2}dY$$

this integral can be solved by the variable substitution $z=ln(Y)$ to give

$$\mathbf{B}(a,\xi,\psi)=\int_{-\infty}^{ln(a)}\frac{e^z}{\sqrt{2\pi}\psi}e^{-\frac{1}{2\psi^2}(z-\xi)^2}dz$$

which can be re-expressed as follows

$$\mathbf{B}(a,\xi,\psi)=\int_{-\infty}^{ln(a)}\frac{1}{\sqrt{2\pi}\psi}e^{-\frac{1}{2\psi^2}(z^2-2\xi z+\xi^2)+z}dz$$

completing the square gives

$$\mathbf{B}(a,\xi,\psi)=e^{\xi+\psi^2/2}\int_{-\infty}^{ln(a)}\frac{1}{\sqrt{2\pi}\psi}e^{-\frac{1}{2\psi^2}(z-(\xi+\psi^2))^2}dz$$

hence

$$\mathbf{B}(a,\xi,\psi)=e^{\xi+\psi^2/2}G\left(\frac{ln(a)-(\xi+\psi^2)}{\psi}\right)$$

and if $ln(a)=\infty$ we obtain

$$E[Y]=e^{\xi+\psi^2/2} \tag{21}$$

Similarly,

$$\mathbf{C}(a,\xi,\psi)=\int_{-\infty}^{a}Y^2\frac{1}{Y\sqrt{2\pi}\psi}e^{-\frac{1}{2\psi^2}(ln(Y)-\xi)^2}dY$$

again using the variable substitution $z=ln(Y)$ gives

$$\mathbf{C}(a,\xi,\psi)=\int_{-\infty}^{ln(a)}\frac{e^{2z}}{\sqrt{2\pi}\psi}e^{-\frac{1}{2\psi^2}(z-\xi)^2}dz$$

and by completing the square we obtain

$$\mathbf{C}(a,\xi,\psi)=e^{2(\xi+\psi^2)}G\left(\frac{ln(a)-(\xi+2\psi^2)}{\psi}\right)$$

If $ln(a)=\infty$ we obtain

$$E[Y^2]=e^{2(\xi+\psi^2)}.$$

By using eq. 21, $E[Y^2]$ can also be expressed as follows

$$E[Y^2]=E[Y]e^{\psi^2}$$

References

[1] Berouti, M. Schwartz, R., Makhoul, J. *Enhancement of Speech Corrupted by Acoustic Noise*, Proc. ICASSP, pp208-211, 1979.

[2] Boll, S.F., *Suppression of Acoustic Noise in Speech Using Spectral Subtraction*, IEEE Trans. on ASSP, Vol. ASSP-27, No. 2, 1979.

[3] Gales, M.J. & Young, S.J., *An improved approach to the Hidden Markov Model Decomposition of Speech and Noise*, Proc. ICASSP, 1992.

[4] Gales, M.J. & Young, S.J., *Cepstral Parameter Compensation for HMM Recognition in Noise*, to appear in Speech Communications.

[5] Geller,D. Heab-Umbacti, R., Ney, H., *Improvements in Speech Recognition for Voice Dialing in the Car Environment*, Proceedings to ESCA Workshop "Speech Processing in Adverse conditions", pp. 203-206, France, Nov. 1992.

[6] Hirsch, H.G., Meyer, P., Ruehl, H.W., *Imporved Speech Recognition using high-pass filtering of subbands envelops*, Proc. European Conf. on Speech Communications and Technology, Genova, pp. 413-16, Sept. 1991.

[7] Juang., B.H., *Speech Recognition in Adverse Environments*, Computer Speech and Language, Vol. 5, pp. 275-294, 1991.

[8] Lass, H. & Gottlieb, P., *Probability and Statistics*, Addison Wesley, 1971.

[9] Lecomte, I., et. al., Proc. ICASSP, pp. 512-519, Glasgow, 1989.

[10] Lockwood, P. & Boudy, J. *Experiments with a Non-linear Spectral Subtractor (NSS), Hidden Markov Models and the Projection, for Robust Speech Recognition in Cars*, ESCA Proc. EUROSPEECH, pp. 79-82, 1991.

[11] Nolazco Flores, J.A. & Young S.J. *Adapting a HMM-based Recogniser for Noisy Speech Enhanced by Continuous Spectral Subtraction*, CUED/INF-ENG/TR.123, April, 1993.

[12] Paul, D.B., *A Speaker-stress Resistant HMM Isolated Word Recogniser*, Proceedings ICASSP, pp. 713-716, Dallas, 1987.

[13] Roe, D.B., *Speech Recognition with a Noise-adapting Codebook, pp. 1139-1142*, Proc. ICASSP, Dallas, 1987.

[14] Van Campernolle, D., *Noise Adaptation in a Hidden Markov Model Speech Recognition System*, Computer, Speech and Language, Vol. 3, pp.151-167, 1989.

[15] Van Campernolle, D., *Increased Noise Inmmunity in Large Vocabulary Speech Recognition with the aid of Spectral Subtraction*, Proc. ICASSP, 27.6, pp.1143-1146, Dallas, 1987.

[16] Young, S. J.,*HTK Version 1.4: Reference and User Manual. Cambridge University Engineering Dept., Speech Group, August, 1991.*