Hidden Markov Model State-Based Noise Cancellation

V. L. Beattie and S. J. Young
CUED/F-INFENG/TR 92 (1992)

February 6, 1992

1 Introduction

This report summarizes the work done at Cambridge University Engineering Department on developing noise-robust algorithms within a Hidden Markov Model recognition system. The aim of this research has been to develop noise-robust algorithms for use within the HMM recognition framework, as opposed to noise cancellation or compensation at the preprocessing level. Such algorithms can take advantage of the clean speech model presented by the HMMs themselves. This research formed part of ESPRIT Project p.2101, "Adverse-Environment Recognition of Speech" (ARS).

Recent work has concentrated on improvement and evolution of approaches developed earlier at CUED and presented in [7, 9]. These earlier methods attempted to eliminate noise by filtering the parameter vector sequences during recognition. One of the main drawbacks of the algorithms (state-based filtering and state-based averaging) was the requirement that the parameter vectors be directly based on spectral energy levels. Development work over the last nine months has thus focussed on generalising state-based Wiener filtering methods for application to other types of data, particularly cepstral data.

The first approach described is a direct adaptation of the state-based Wiener filtering developed previously at CUED [2, 9] for cepstral data. This involved integrating the original method into a combined preprocessing and recognition system, where the energy-level parameters (either filterbank or FFT output) are first put through noise-cancelling filters, and then through the cepstral transform. The resulting noise-cancelled vectors are then used to recognise the utterance.

The second adaptation to the original algorithm was an implementation of state-based cepstral means correction based on Wiener filtering. This method implements the cepstral equivalent of a waveform domain filter, thus effectively compensating for HMM means distortion due to noise. The technique applies the fact that in the spectral domain, filtering corresponds to a multiplication of the signal spectrum and the filter frequency response. Due to the log scaling involved in homomorphic filtering, in the cepstral domain this corresponds to an additive correction.

The other major activity covered by this report has been the testing and comparison of algorithms. This has involved comparative testing of the state-based filtering and mean correction algorithms with other approaches. Testing of combinations of algorithms has also been undertaken where appropriate.

2 State-Based Filtering with Cepstral Data

The first extension of previous work was to implement the original state-based filtering method in a combined preprocessing/recognition system. The aim of this adaptation was to enable the technique of state-based filtering to be applied in systems where the final parameter vectors are not directly based on spectral energy levels, but where some intermediate stage (e.g. filterbank or FFT values) represents an estimate of the signal power spectrum. The idea is to implement state-based filtering at this intermediate stage, and then pass the filtered vectors on to further (non-linear) processing. This method can then be used in systems employing, for instance, cepstral vectors, which tend to give higher and more robust baseline performance than straightforward energy-level based front ends such as filterbanks.

2.1 Adaptation for Cepstral Data

The off-line design of the noise-cancelling filters for this adapted method requires both the intermediate spectral vectors and the final cepstral vectors. The intermediate stage used in the filtering process was simulated filterbank values, derived from an FFT using mel-spaced triangular filters. Using models trained on clean cepstral data, the cepstral training set can be segmented, and these segmentation points used to assign the intermediate level vectors to the HMM states. Binned FFT vectors can also be obtained from periods of background noise, and using this speech and noise information the filters are designed as described previously in [2, 7, 9]. The adapted algorithm is summarized below.

2.2 Algorithm Description

In Hidden Markov Model state-based filtering, recognition scores are updated based on a weighted sum of noisy input vectors. The vectors used for this version of the algorithm are simulated filterbank output obtained by binning short-term FFTs into mel-spaced frequency bins. The weights used in the summation process are determined by designing an FIR Wiener filter.

The first stage of the algorithm involves the design of the noise cancelling Wiener filters. The filter design process for a single Hidden Markov Model is outlined below. We use the following definitions:

- $s_{i,k}(t)$ speech energy level for state i and simulated filterbank bin k.
- $n_k(t)$ noise energy level for bin k.
- $u_{i,k}(t)$ noisy speech energy level for state i and bin k.
- dur_i average duration (number of vectors) for state i.
- p_i filter length (number of vectors in weighted sum) for state i.
- $r_{s_{i,k}}(\tau)$ autocorrelation function for speech signal $s_{i,k}(t)$, for state i, bin k, and delay τ. Calculated from clean training data.
- $r_{n_k}(\tau)$ autocorrelation function for noise signal $n_k(t)$, for bin k and delay τ.
- M_s number of clean speech training sequences.
- T length of unknown test sequence.
- K_f dimension of intermediate phase (binned FFT) vector.
- K_c dimension of cepstral vector, not including delta-cepstral coefficients.

We assume that HMMs divide speech into stationary segments, and that the background noise characteristics change slowly relative to the speech. Therefore, $s_{i,k}(t)$ and $n_k(t)$ are assumed stationary for the purpose of filter design. The values of $r_{s_{i,k}}(\tau)$ and $r_{n_k}(\tau)$ are estimated from clean speech training data and from recorded background noise sequences. In addition, the simulated filterbank energy levels are considered independent. Although not strictly true, it allows us to consider each channel or bin as a separate one-dimensional signal.

In the following discussion we consider the problem for a particular state and channel. For simplicity we omit the i and k variables; thus $r_{s_{i,k}}(\tau)$ is abbreviated to $r_s(\tau)$, $r_{n_k}(\tau)$ to $r_n(\tau)$, etc.

The optimal FIR Wiener filter given a contaminated signal $u(t)$ and a desired signal $s(t)$ is determined from the following equation [8]:

$$\mathbf{R}_u \mathbf{h} = \mathbf{r}_{s,u} \tag{1}$$

where

$$\mathbf{R}_u = \begin{bmatrix} r_u(0) & r_u(1) & r_u(2) & \cdots & r_u(dur-1) \\ r_u(1) & r_u(0) & r_u(1) & \cdots & r_u(dur-2) \\ r_u(2) & r_u(1) & r_u(0) & \cdots & r_u(dur-3) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_u(dur-1) & \cdots & \cdots & \cdots & r_u(0) \end{bmatrix}$$

$$\mathbf{h} = \begin{bmatrix} h(0) \\ h(1) \\ h(2) \\ \vdots \\ h(dur-1) \end{bmatrix}$$

$$\mathbf{r}_{s,u} = \begin{bmatrix} r_{s,u}(0) \\ r_{s,u}(1) \\ r_{s,u}(2) \\ \vdots \\ r_{s,u}(dur-1) \end{bmatrix}$$

h is the vector (of dimension dur) of Wiener FIR filter coefficients.

$r_u(t)$ is the noisy signal autocorrelation function

$r_{s,u}(t)$ is the cross-correlation function between the desired signal $s(t)$ and the noisy signal $u(t)$.

The speech and noise signals are assumed to be additive. This a reasonable assumption to make for the noisy environments considered, although not exact as it does not take into account phenomena such as the Lombard effect. u(t) can now be estimated as follows:

$$u(t) = gs(t) + n(t) \tag{2}$$

The scaling factor g is an SNR-matching gain (discussed in further detail in Section 3) which compensates for the difference between speech energy levels in the clean and unknown utterances. With the added assumption that the speech and noise signals are uncorrelated, the functions r_u and $r_{s,u}$ can now be solved for in terms of r_s and r_n. Since estimates of r_s and r_n have been calculated, equation 1 can now be solved to determine the vector $\mathbf{h}$ of FIR coefficients.

During recognition, the input to the combined recognition/noise cancellation system is intermediate phase (simulated filterbank) vectors of unknown noisy speech. These are put through the noise-cancelling filters attached to each state to obtain a smoothed estimate of the speech power spectrum. The filter output is defined as follows:

$$\bar{O}_{i,t} = \sum_{\tau=t-p_i+1}^{\tau=t} O_\tau h_i(\tau) \tag{3}$$

This smoothed vector $\bar{O}_{i,t}$ is log scaled and cosine transformed to obtain the cepstral vector, which is then used to compute the state output for the Viterbi algorithm as in standard recognition. The delta-cepstral values which make up the other half of the parameter vector are obtained via normal preprocessing. They are not obtained at run-time from the filtered cepstral values due to the complexity of calculating delta cepstra over a window of filtered cepstral vectors during recognition. This complexity arises from the fact that in state-based filtering, each state will produce a different smoothed vector.

2.3 Computational Considerations

The computational cost of this method is the cost of the original state-based filtering algorithm, plus the additional cost of cepstral processing the filtered vectors during recognition. The cost of filter design, as calculated in [2], is $K_f[p_i \times dur_i \times M_s + {p_i}^3 + O(p_i^2)]$ multiply and accumulate operations per HMM state. The cost of filtering during recognition is $p_i \times K_f \times T$ multiply and accumulate operations per HMM state. For the implementation of state-based filtering using MFCC data, there is also the run-time cost of log compressing the filtered vector and applying the discrete cosine transform. Even ignoring the cost of the logarithm and cosine functions (at minimum a table look-up), this additional cost will be significant: $K_c \times K_f \times T$ multiply and adds per HMM state.

3 Cepstral Means Correction Based on Wiener Filtering

As indicated by the results given in the next section, only small increases in recognition rate were gained using the state-based filtering method described above. These improvements will not be sufficient in a noisy environment, nor are they large enough to justify the high computational cost. For this reason a method employing the same filtering techniques in a different way was developed. Although this second approach still lies within the general framework of state-based filtering, we refer to it as state-based cepstral means correction to differentiate it from our earlier work.

Berstein and Shallom [5] implement a cepstral means correction for template-based recognition which hypothesizes a Wiener filter based correction vector $\mathbf{c}_w$ for each dynamic programming match of cepstral template vector $\mathbf{c}_s$ and unknown noisy speech vector $\mathbf{c}_u$, using local estimates of the speech and background noise power spectra. Their system employs cepstra derived from linear prediction coefficients in a dynamic time warping recognition framework. Although significant improvement is obtained, the computational requirements are prohibitive. This is due to the fact that corrections for all possible template to noisy speech matches are calculated on-line, and to the fact that the Wiener filter estimation is done in an inefficient manner.

CUED's work on state-based cepstral shifts has involved implementing a method related to [5] within an HMM system. This involves broadening the stationarity assumption for the speech signal, so that the entire segment of speech assigned to a state is assumed stationary, rather than just individual frames. Calculating a cepstral shift for each HMM state, rather than each template vector, involves a large computational savings in itself. In addition more efficient ways of calculating the shift have been derived. However, the amount of computation required is still high; further reductions can be made by fixing some of the parameters required to calculate $\mathbf{c}_w$ based on overall estimates. Calculation of the required shift can then be made off-line, thus reducing the effective run-time cost to zero.

3.1 Algorithm Description

The development of a cepstral means correction approach was motivated by the need to improve the performance of state-based filtering and at the same time limit computational cost. The filtering algorithm is outlined below. We use the following definitions:

$S_i(\omega)$ some estimate of the clean speech power spectrum for state i.

$U_i(\omega)$ some estimate of the noisy speech power spectrum corresponding to state i.

$N(\omega)$ some estimate of the noise power spectrum corresponding to state i.

$W_i(\omega)$ some estimate of the Wiener filter frequency response for state i.

E_s average speech energy.

E_n average noise energy.

E_u average noisy speech energy.

M_s number of clean speech training sequences.

K_f dimension of intermediate phase (binned FFT) vector.

K_c dimension of cepstral vector, not including delta-cepstral coefficients.

For cepstral means correction, the filtering problem is formulated as follows. We consider a Wiener filter applied in the waveform domain to a particular frame of speech, as opposed to filtering at the parameter vector level over several frames as in state-based filtering. At the spectral energy level, this filter has a multiplicative effect:

$$S_i(\omega) = U_i(\omega)W_i(\omega) \tag{4}$$

This multiplicative relationship has been used for noise cancellation approaches such as [1, 6]. The log compression applied to the spectra in the cepstral transform results in the following relationship:

$$\log S_i(\omega) = \log U_i(\omega) + \log W_i(\omega) \tag{5}$$

Since the DCT subsequently applied to the data is a linear transformation, the additive relationship holds for the cepstral vectors themselves:

$$\mathbf{c}_{s_i} = \mathbf{c}_{u_i} + \mathbf{c}_{w_i} \tag{6}$$

Thus if the filter frequency response $W_i(\omega)$ can be estimated, an additive cepstral correction vector $\mathbf{c}_{w_i}$ can also be found via the cepstral transform. Assuming that speech and noise are additive, and that the speech and noise signals are locally stationary, the following standard Wiener filter will give optimum noise cancellation:

$$W_i(\omega) = \frac{g\, S_i(\omega)}{g\, S_i(\omega) + N(\omega)} \tag{7}$$

As in Equation 2, the scaling factor g is an SNR-matching gain which compensates for the difference between speech energy levels in the clean and unknown utterances. In our case, the "locality" in which the signal is assumed stationary is the Hidden Markov Model state itself; in other words we make the approximate assumption that HMMs segment speech into stationary segments. Thus a separate correction will be calculated for each model state, as indicated by the subscript i. Speech and noise additivity is also assumed as previously.

From Equation 6 it is apparent that for the steady-state case (i.e. remaining within one state), the delta cepstral correction will be zero. When the window from which the delta values are calculated spans a transition from one state to the next, however, a small but non-zero correction value could result. However in practice it is not possible to calculate this correction based on the shifted cepstral input. This is because the "future" half of the window needed to calculate the delta cepstra is not known. Therefore the uncorrected (noisy) delta cepstral values are used, as in the state-based filtering method.

In order to solve Equation 7, estimates of the speech, noise, and noisy speech signal characteristics must be made. Information about the noise is available from the data itself, assuming a reasonably accurate endpointing process to separate speech from non-speech background. Speech information is also available, in the form of the clean speech training data. The average speech spectrum $S_i(\omega)$ associated with each state can be found by using trained models to segment the clean speech training data set. As with state-based filtering, these segmentation points can be used to determine which frames of speech are assigned to each state. The average spectra of these frames is estimated using the simulated filterbank vectors, which are calculated as an intermediate product in cepstral processing.

The only quantity which we still need to estimate in Equation 7 is the gain g. This scaling factor reflects the ratio of the average test speech energy to the average training speech energy, in order to compensate for effects such as louder speech in a noisy environment, or even different volume levels of recording equipment. The gain is set as:

$$g = \frac{(E_{ns} - E_n)}{E_s} \tag{8}$$

Where E_n and E_s are calculated based on samples of the background noise and samples of the training data respectively. E_{ns}, the unknown speech plus noise signal energy, is calculated from a few samples of the noisy test data. In a real-time system, of course, some initial estimate would have to be made for the gain at the start of recognition, and then updated based on the unknown speech input to the system.

3.2 Computational Considerations

If global estimates for the gain parameter and the noise spectrum are used in estimating the Wiener filter frequency response, the state-based cepstral corrections can be calculated entirely off-line. The main computational cost of this process is the expense of estimating average values for the noise and speech power spectra. This cost is $K_f \times dur_i \times M_s$ additions per HMM state for the speech and $K_f \times F_n$ additions for the noise, where F_n is the number of noise frames used to estimate the noise power spectrum. Additional computation involved in this calculation is the cost of Equation 7, i.e. $2K_f$ multiplies and K_f additions per HMM state. A final cost is the cepstral transform to calculate $\mathbf{c}_w$, which requires $K_f \times K_c$ multiply and add operations per state, plus whatever cost is associated with the log and cosine functions involved. Since the HMM cepstral means can simply be shifted by the appropriate correction vector prior to recognition, there is no run-time cost for the algorithm. Our investigations indicate that performance is not particularly sensitive to the exact values of the gain and average noise power spectrum. However if the noise characteristics changed significantly over time, the corrections would have to be re-calculated, ideally during periods of silence when the recogniser is idle.

It is also possible to use a local estimate for the noise spectrum, based on the noise frames immediately preceding the unknown utterance. In this case the cepstral corrections must be recalculated for each unknown utterance. The noise spectral estimate will involve a cost of $K_f \times F_n$ addition operations. The recalculation of the correction vectors takes place immediately prior to recognition, and involves that same cost as above, i.e. $K_f(K_c+2)$ multiplies and $K_f(K_c+1)$ adds for each HMM state. There is no further cost since the clean speech power spectra required for estimating $W_i(\omega)$ can be calculated beforehand.

Local values for the gain parameter in Equation 7 can also be obtained, based on the actual frame-by-frame energy level of the unknown speech. In order to apply this information, however, the cepstral shifts have to be calculated separately for each frame. Preliminary tests indicated that this added cost is not justified, because using a global estimate for the gain performed better than a local one anyway. This is not a surprising result since the gain parameter is intended to reflect the overall energy ratio between clean speech and speech embedded in noise, rather than local fluctuations.

4 Results

4.1 The Baseline Recognition System

All of the algorithms were implemented using the HTK Hidden Markov Model Toolkit software package and library [10]. All tests used a single mixture, Gaussian pdf HMM trained using a fixed variance vector for all HMM states. This variance is calculated from all of the clean training data for each database, and is not re-estimated. Aside from this distinction, standard initialisation and Baum-Welch training procedures are used. In the tests denoted as "standard" in the results tables, eight-state HMMs were used, with training vectors consisting of twelve cepstra plus twelve delta-cepstra parameters derived from ten millisecond windows of signal. The remaining tests used 16-state HMMs, with training vectors consisting of ten cepstral coefficients. The original reason for the 16-state tests was for comparison to other algorithms developed within the ARS project, however the results are included here for completeness and as evidence of the robustness of the algorithm.

Work has been undertaken on linear prediction-based, filterbank-based, and FFT-based cepstra. Similar recognition improvement has been observed for all cepstral front ends tested; however the results presented use FFT-based cepstral vectors only. This is in the interest of standardisation and comparability, since this type of MFCC front-end is widely used in the ARS project.

4.2 Test Databases

Isolated word data provided by partners in the ARS project was used for testing. The results tables are labeled by database: “CSELT” refers to a 34-word, four speaker Italian language database, and “ENST” refers to two 43-word, single speaker French language databases. The noisy data used for testing was collected in a car travelling at high speeds (110km or 130km) on the motorway.

4.3 Comparison and Combination of Algorithms

The performance of the algorithms developed was measured against those of other techniques for noise-robust speech recognition. The comparisons made were mainly to other algorithms researched in the ARS project, although a comparison with a projection distance measure technique [4] is also made.

The combination of different algorithms for improved overall performance was also considered. Combining two techniques is not always appropriate, since algorithms which aim to improve performance in noise are usually attempting some form of noise cancellation. Using two such methods in combination would result in over-correction for the noise. However combination is sometimes possible; here we attempt to integrate state-based filtering with state-based cepstral means correction, and also combine cepstral mean correction with the projection distance measure.

Obviously not all possible comparisons and combinations have been considered. We have attempted to give a reasonable comparative evaluation of our algorithms; however time constraints and the difficulty of comparing results across different databases and preprocessors limit the number of comparisons that can be made.

In the interest of space and conciseness, not all results for all possible methods (and combinations of methods) are shown. For algorithms whose performance was not particularly promising, or not good enough to justify the computational cost, only a representative selection of results are presented here.

4.4 State-based Filtering with Cepstral Output

Although positive results have been obtained for state-based filtering in a filterbank-based system [9], the results shown in Table 1 for cepstral data were less promising. Part of the problem may be that cepstral processing is a priori more robust to noise. In addition the combined system segments speech using the cepstral sequences; the division thus obtained may not correspond to quasi-stationary segments in the underlying intermediate stage vectors. The combination of state-based filtering and cepstral correction was likewise not particularly successful. For this combination the filters were designed to eliminate the noise variance; i.e. r_n was calculated as noise variance rather than autocorrelation. Since cepstral correction compensates for the effects of the noise mean, the two algorithms might be expected to be completely complementary. However the results indicate that performance is slightly worse than for cepstral correction alone.

4.5 Cepstral Means Correction

The cepstral means correction technique, on the other hand, gave consistent and significant performance improvement for all data tested. Results are shown in Tables 2, 3 and 4. In each case, two versions of the algorithm were used. The first uses a global estimate of the noise power spectrum (based on a random selection of noise sequences.) The second uses a local estimate for the noise, based on the background noise immediately preceding the unknown utterance. In some cases, this latter approach gave the best performance by a small margin; however due to its inconsistent performance, sensitivity to end-pointing accuracy, and higher computational cost, it was dropped in later testing (and in combined method testing) in favor of the global noise estimate. All results shown here use a global estimate for the gain parameter g. A global estimate for the gain offers both higher performance and lower cost relative to a local gain estimate.

4.6 Cepstral Projection Distance Measure

Another noise-robust technique for cepstral data which has been developed recently is the use of distance measures based on projection rather than Euclidean distance [3, 4]. It can be shown [3] that projection distance measures will result in smaller distortions than Euclidean distance measures for additive white noise. As the results in Tables 5, 6 and 7 demonstrate, however, the weighted cepstral projection measure (defined as in [4]) does not provide as good performance as cepstral means correction. In some cases, in fact, performance is quite poor. One explanation for this may be that cepstral projection only partially compensates for noise distortion (unless the noise vector is in the same direction as the underlying speech.) Moreover the projection measure may also reduce the distance between a noisy input vector and an "incorrect" clean model vector, thus leading to misrecognition. Finally of course the noise involved is not white noise which may reduce the effectiveness of the projection measure.

The combination of cepstral projection and cepstral correction seems to provide slightly worse performance than either technique alone. Assuming that the cepstral correction results in a relatively "clean" unknown vector, using the projection measure may introduce errors due to the reduced distance between model and unknown vectors which do not truly match. This possibility is supported by the results reported by Mansour and Huang [3], who found a degradation in recogniser performance on clean speech when the projection measure was used.

4.7 Interpretation and Evaluation of Results

The extremely marginal gains obtained by the state-based filtering algorithm suggest that it is not a viable approach towards building a noise-robust HMM system. There are several possible reasons for the poor performance of the algorithm. In the first case, the method is best suited for modelling and compensating for the noise variance in the parameter space. However the error introduced by vector-level noise variance may not be crucial, particularly with fixed variance HMMs. Furthermore, any improvement obtained may be counteracted by the loss of speech information due to the process of smoothing over a window of input vectors. In the case of state-based filtering with cepstral output, the effects of eliminating noise through the filtering algorithm may be overshadowed by the larger improvement obtained simply by using the noise-robust cepstral transform.

State-based cepstral correction, on the other hand, significantly increases performance in noise at a very low computational cost. It compares favourably to other cepstral correction noise cancellation methods thus far developed, and has the advantage of requiring no manually set thresholds or scaling factors. It also does not require noisy data, except for the estimation of gain factor g and the background noise power spectrum. The latter can easily be obtained from periods of "silence" in the input speech. Furthermore, since performance is not particularly sensitive to these values, run-time operation could presumably obtain equally good performance even if other system factors, such as end-pointing methods, are not exact. Finally state-based cepstral correction also provides more consistent performance on car noise than a method such as cepstral projection, which is ideally aimed at eliminating contaminating white noise from input speech.

The overall poor results obtained from attempting to combine algorithms suggest that the interaction between the various techniques is strong. This is true even when the two methods combined (e.g. state-based filtering and state-based averaging) would seem to be complementary.

5 Conclusions

The advantage of developing noise compensation and cancellation algorithms within a Hidden Markov Model framework is the availability of a specific model of the speech, provided by the models themselves. Using this information explicit state-based filters can be implemented. Two types of filtering have been described. The first approach, state-based filtering, applies noise cancellation in the parameter space by smoothing the parameter vector signal. The state-specific nature of the filtering process, and the requirement of noise and speech additivity, result in a computationally intensive process. This is because filtering and non-linear processing of the filter output must be done separately for each model state. The improvements obtained, moreover, do not seem to justify this computational cost. State-based filtering is designed to reduce the effect of the noise variance (the noise mean is more straightforwardly dealt with using methods such as spectral subtraction.) However the results obtained here would seem to indicate that simple variance-broadening techniques such as grand variance are more noise-robust at a much lower computational cost.

The technique of cepstral means correction, on the other hand, shows consistently high performance. Cepstral means correction is the equivalent of within-frame Wiener filtering of the original waveform. In the cepstral parameter space this can be shown to correspond to an additive shift. State-specific cepstral shifts are implemented based on the speech associated with each state, with a stationary (or at least slowly changing) background noise model assumed.

This method results in consistently high performance and has the advantages of simplicity and very low computational cost. Furthermore, the general technique need make no assumptions about the nature of the noise; more complex noise models could be used to hypothesise a range of possible corrections in the case of varying noise.

We feel that this method could profitably be incorporated in any cepstral- based recognition system for noisy environments. The algorithm seems particularly successful on data with high levels of noise. The performance of cepstral means correction on larger and more complex systems (larger vocabulary for instance or with rapidly time-varying noise) remains a topic for further investigation.

References

[1] Y. Ephraim, "A Minimum Mean Square Error Approach for Speech Enhancement," IEEE Proc. Int. Conf. Acoust. Speech & Signal Process., ICASSP'90, pp. 829-832, 1990.

[2] V.L. Beattie & S.J. Young, "Hidden Markov Model Algorithms for Noisy Speech Recognition," Deliverable 4100/2 (HMM Enhancement in Noise), ESPRIT II Project N. 2101: ARS, pp. 44-57, July 1991.

[3] D. Mansour & B.-H. Huang, "A Family of Distortion Measures Based Upon Projection Operation for Robust Speech Recognition," IEEE Trans. on ASSP, ASSP-37(11), pp. 1659-1671, 1989.

[4] B. A. Carlson & M. A. Clements, "Application of a Weighted Projection Measure for Robust Hidden Markov Model Based Speech Recognition," IEEE Proc. Int. Conf. Acoust. Speech & Signal Process., ICASSP'91, pp. 921-924, 1991.

[5] A. D. Berstein & I. D. Shallom, "An Hypothesized Wiener Filtering Approach to Noisy Speech Recognition," IEEE Proc. Int. Conf. Acoust. Speech & Signal Process., ICASSP'91, pp. 913-916, 1991.

[6] A. Nadas, D. Nahamoo & M. A. Picheny, "Speech Recognition Using Noise-Adaptive Prototypes," IEEE Proc. Int. Conf. Acoust. Speech & Signal Process., ICASSP'88, pp. 517-520, 1988.

[7] V. L. Beattie & S. J. Young, "Speech Recognition in Noise: Experiments Using Hidden Markov Models," Proc. Inst. of Acoust. Autumn Conf., pp. 553-558, 1990.

[8] C.-T. Chen, *One-Dimensional Digital Signal Processing*, p. 412, Marcel Dekker, New York, 1979.

[9] V. L. Beattie & S. J. Young, "Noisy Speech Recognition Using Hidden Markov Model State-Based Filtering," IEEE Proc. Int. Conf. Acoust. Speech & Signal Process., ICASSP'91, pp. 917-920, 1991.

[10] S. J. Young, "HTK Reference Manual," CUED, December, 1990.

Speaker	Condition	Results		
		Baseline	Filtering	with Cep. Corr.
cr	130km	88.82	90.00	95.29
eb	130km	89.41	90.59	94.12
mb	130km	86.47	85.29	90.58
rc	130km	97.05	98.24	98.24

Table 1: CSELT Database (Standard)

Database No.	Speaker	Condition	Results		
			Baseline	Global	Local
1	yg	110km	63.41	95.66	95.97
2	em	110km	65.68	98.29	98.13

Table 2: ENST Databases with 16-state HMMs and 10 MFCCs

Database No.	Speaker	Condition	Results		
			Baseline	Global	Local
1	yg	110km	63.35	94.09	93.17
2	em	110km	93.63	95.81	98.14

Table 3: ENST Databases (Standard)

Speaker	Condition	Results		
		Baseline	Global	Local
cr	130km	88.82	96.47	94.70
eb	130km	89.41	95.29	95.88
mb	130km	86.47	95.29	96.47
rc	130km	97.05	98.24	97.05

Table 4: CSELT Database (Standard)

Database No.	Speaker	Condition	Results		
			Baseline	Proj.	with Cep. Corr.
1	yg	110km	63.41	57.67	94.57
2	em	110km	65.68	67.39	94.57

Table 5: ENST Databases 16-state HMMs and 10 MFCCs

Database No.	Speaker	Condition	Results		
			Baseline	Proj.	with Cep. Corr.
1	yg	110km	63.35	85.25	90.07
2	em	110km	91.93	94.11	91.93

Table 6: ENST Databases (Standard)

Speaker	Condition	Results		
		Baseline	Proj.	with Cep. Corr.
cr	130km	88.82	92.35	93.53
eb	130km	89.41	96.47	96.47
mb	130km	86.47	95.29	92.94
rc	130km	97.05	99.41	97.05

Table 7: CSELT Database (Standard)