MODEL-BASED TECHNIQUES FOR NOISE ROBUST SPEECH RECOGNITION

Mark John Francis Gales

Gonville and Caius College

September 1995

Dissertation submitted to the University of Cambridge
for the degree of Doctor of Philosophy

Declaration

This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration, except where stated. It has not been submitted in whole or part for a degree at any other university.

The length of this thesis including footnotes and appendices is approximately 36,000 words.

Summary

This thesis details the development of a model-based noise compensation technique, Parallel Model Combination (PMC). The aim of PMC is to alter the parameters of a set of Hidden Markov Model (HMM) based acoustic models, so that they reflect speech spoken in a new acoustic environment. Differences in the acoustic environment may result from additive noise, such as cars passing and fans; or convolutional noise, such as channel differences between training and testing. Both these classes of noise have been found to seriously degrade speech recognition performance and both may be handled within the PMC framework.

The effect of noise on the clean speech distributions and associated parameters was investigated. The shape of the corrupted-speech distribution could become distinctly non-Gaussian, even when Gaussian speech and noise sources were used. This indicates that, to achieve good performance in additive noise, some flexibility in the component alignments, or number, is required within a state. For the model parameters, additive noise was found to alter all the means and the variances of both the static and, where present, dynamic coefficients. This shifting of HMM parameters was observed in terms of both a distance measure, the average Kullback-Leibler number on a feature vector component level, and the effect on word accuracy. For best performance in noise-corrupted environments, it is necessary to compensate all these parameters.

Various methods for compensating the HMMs are described. These may be split into two classes. The first, non-iterative PMC, assumes that the frame/state component alignment associated with the speech models and the clean speech data is unaltered by the addition of noise. This implies that the corrupted-speech distributions are approximately Gaussian, which is known to be false. However, this assumption allows rapid adaptation of the model parameters. The second class of PMC is iterative PMC, where only the frame/state alignment is assumed unaltered. By allowing the component alignment within a state to vary, it is possible to better model the corrupted-speech distribution. One implementation is described, Data-driven Parallel Model Combination (DPMC). A simple and effective method of estimating the convolutional noise component in the presence of additive noise is also described.

These techniques were evaluated on three databases. NOISEX-92 is a small vocabulary speech recognition in noise task. The signal-to-noise ratios considered were extreme, going down to -6dB, and both additive and convolutional test sets were evaluated. The second test set was based on the 1000 word DARPA Resource Management test set. For this test set noise was added from the NOISEX-92 database. Again, both additive and convolutional noise conditions were considered. Finally, the 5000 word ARPA 1994 CSRNAB Spoke 10 evaluation data was examined. For all the databases and noise environments considered, PMC was found to achieve good noise robustness.

Acknowledgement

First, I would like to thank my supervisor Steve Young. By providing the right balance of suggestions, criticism and freedom, he has made this work possible. In particular, I am grateful for the support he has given me over the years in whatever I have decided to do. In conjunction with Phil Woodland and Julian Odell, he also provided the basic speech recognition software, HTK and the WSJ speech recognition system, used throughout this work.

Over the years I have benefited greatly from interaction with other members of the Speech, Vision and Robotics group at Cambridge University. There are too many people to mention individually, but I must thank Jonathon Foote, Valtcho Valtchev, Kate Knill (repeatedly), Steve Young and Carl Seymour, who had the dubious pleasure of proof-reading this thesis. This work was made possible by the excellent computing facilities in the group, thanks to Patrick Gosling, Andrew Gee, Tony Robinson, Carl Seymour and Richard Prager for maintaining the computer system.

I would also like to thank Martin Russell, my industrial supervisor, and my CASE award sponsors, the Speech Recognition Unit, DRA Malvern, for supplying both financial and technical support over the time of my studies. I must also thank the speech group at the IBM T.J. Watson Research Center, in particular Michael Picheny and Sree Balachrishnan-Aiyer, for making my visit to the United States so enjoyable and rewarding.

My research and conference trips have been funded by the EPSRC, DRA Malvern, the Royal Academy of Engineering, Gonville and Caius College Cambridge and Smiths Systems Engineering. I am very grateful to all of them.

Finally, I would like to thank my family for all their support over the years, despite my lack of phone calls.

Contents

1 Introduction **1**
1.1 Speech Recognition 2
1.2 Speech Recognition in Adverse Environments 4
1.3 Organisation of Thesis 5

2 Hidden Markov Model Speech Recognition Systems **6**
2.1 The Hidden Markov Model 6
2.2 Pattern Matching with HMMs 8
2.3 The Forward-Backward Algorithm 10
2.4 Estimation of HMM Parameters 11
2.5 Speech Recognition using HMMs 13
2.6 HMM Parameter Tying 14
2.7 Speech Parameterisation 15
2.7.1 The Linear-Spectral and Log-Spectral Domains 15
2.7.2 The Cepstral Domain 16
2.7.3 Linear Prediction Cepstral Domain 17
2.7.4 Dynamic Coefficients 17

3 Techniques for Noise Robustness **19**
3.1 Inherently Robust Speech Parameters 19
3.1.1 Cepstral Mean Normalisation 19
3.1.2 Short-term Modified Coherence 20
3.1.3 Linear Discriminant Analysis 20
3.1.4 Generalised Cepstral Analysis 20
3.1.5 RASTA-PLP 21
3.2 Clean Speech Estimation 22
3.2.1 Spectral Subtraction 22
3.2.2 Probabilistic Optimal Filtering 22
3.2.3 Code-book Dependent Cepstral Normalisation 23
3.2.4 State-Based Speech Enhancement 24
3.3 Model-Based Techniques 24
3.3.1 Linear Regression Adaptation 25
3.3.2 Model-Based Stochastic Matching 25
3.3.3 Noise Masking 26
3.3.4 Speech and Noise Decomposition 27
3.3.5 Hypothesised Wiener Filtering 28
3.4 Combination of Approaches 29

4 The Effects of Additive Noise **31**

5 Parallel Model Combination **35**

5.1 Basic Parallel Model Combination Description 35
5.2 Mapping from the Cepstral to the Log-Spectral Domain 37
5.3 The Mismatch Function 37
5.3.1 Mismatch Function for the Static Parameters in Additive Noise . . . 38
5.3.2 Mismatch Function for Dynamic Parameters in Additive Noise . . . 38
5.3.3 Mismatch Function for Additive and Convolutional Noise 39
5.4 Variations on the Mismatch Function 40
5.4.1 Domain of Noise Addition 40
5.4.2 Matrix-Based Dynamic Coefficients 41
5.4.3 Continuous-Time Dynamic Coefficients 42
5.5 Training HMMs on Statistical Data 43
5.6 Non-Iterative PMC 45
5.6.1 Numerical Integration 46
5.6.2 Log-Normal Approximation 47
5.6.3 Log-Add Approximation 48
5.7 Iterative PMC 49
5.8 Estimation of the Convolutional Noise Component 51
5.9 Complex-Noise Environments 53
5.10 Practical Implementations 54
5.10.1 Speech and Noise Model Covariance Approximations 55
5.10.2 Corrupted-Speech Model Covariance Approximation 56
5.10.3 Cepstral Smoothing 57
5.11 Training Models on Noise-Corrupted Data 57
5.11.1 Multi-Pass Training on the Noise-Corrupted Database 58
5.11.2 Single-Pass Training on the Noise-Corrupted Database 58
5.12 Performance Criteria 59
5.12.1 Distance Measures between Model Sets 59
5.12.2 Percentage Word Accuracy 60

6 Interfering Noise Sources Considered **61**

6.1 Noises from the NOISEX-92 Database 61
6.2 Noises from the ARPA 1994 Evaluation 64

7 Evaluation on NOISEX-92 **66**

7.1 The NOISEX-92 System 66
7.2 Results on NOISEX-92 with Additive Noise 67
7.3 Results on NOISEX-92 with Additive and Convolutional Noise 69
7.4 Discussion 72

8 Evaluation on a Noise-Corrupted Resource Management Task **73**

8.1 The Resource Management System 73
8.2 Results on the Clean RM Database 75
8.3 Results on RM with Additive Noise 76
8.3.1 Training on RM Additive Noise-Corrupted Data 76
8.3.2 PMC on Lynx Additive Noise-Corrupted RM 78
8.3.3 PMC on Operations Room Additive Noise-Corrupted RM 85

8.3.4 PMC on Machine Gun Additive Noise-Corrupted RM 87
8.4 Results on RM with Additive and Convolutional Noise 88
8.5 Discussion 91

9 Evaluation on the ARPA 1994 CSRNAB Spoke 10 94
9.1 The Wall Street Journal System 94
9.2 Results on the Development Data 95
9.3 Results on the Evaluation Data 96
9.4 Discussion 99

10 Conclusions and Future Work 102
10.1 Summary of Results 103
10.2 Future Work 104

A Derivation of the Delta and Delta-Delta Mismatch Function 106

B Numerical Integration 108

C Derivation of Multi-Variate Gaussian to Log-Normal Mapping 111

List of Figures

1.1 General speech recognition system 3
1.2 Simplified distortion framework 5

2.1 A 3-emitting state hidden Markov model 7
2.2 Composite hidden Markov model 9

3.1 3-dimensional Viterbi decoder 28

4.1 Plots of "corrupted-speech" distribution (solid), and maximum likelihood Gaussian distribution (dashed) 32
4.2 Plots of "corrupted-speech" distribution, maximum likelihood Gaussian distribution, and speech and noise decomposition distribution, noise mean=4 . 33
4.3 Plots of "corrupted-speech" distribution (dashed), maximum likelihood Gaussian distribution (dash-dot), and multiple mixture component distributions (solid), noise mean=4 34

5.1 The basic parallel model combination process 36
5.2 Non-iterative parallel model combination 45
5.3 Data-driven parallel model combination 50
5.4 Parallel model combination for additive and convolutional noise 52
5.5 Mapping a 3-D Viterbi decoder to a 2-D Viterbi decoder 54

6.1 Log power spectral density against frequency for Lynx Helicopter noise and Operations Room noise taken from the NOISEX-92 database 62
6.2 Log-Spectral values against mel-spaced frequency bin for noises from the NOISEX-92 database 62
6.3 Time domain plot of the Machine Gun noise taken from the NOISEX-92 database 63
6.4 Log-Spectral value against mel-spaced frequency bin for a single and two Gaussian mixture component models from the NOISEX-92 database 63
6.5 Log-Spectral value against mel-spaced frequency bin for the three ARPA development car noises at 18dB 64
6.6 Log-Spectral value against mel-spaced frequency bin for the three ARPA evaluation car noise levels 64

7.1 Average KL number for clean, Log-Normal compensated, mean-and-variance-compensated, and mean-compensated using the Numerical Integration approximation and Diagonal covariance approximation model sets on Lynx Helicopter noise at -6dB 67

7.2 Average KL number for clean, Diagonal and Full covariance approximations on Lynx Helicopter noise at -6dB 68
7.3 Average KL number for the PMC Numerical Integration approximation compensated models with and without convolutional noise compensation on Lynx Helicopter noise at 0dB 70

8.1 Average KL number for clean, DPMC-compensated and 24-dimensional truncated to 13-dimensions DPMC compared to 13-dimensional DPMC-compensated model sets on Lynx Helicopter additive noise-corrupted RM at 18dB 79
8.2 Average KL number for the clean, DPMC mean-and-variance-compensated, DPMC mean-compensated, the Log-Add approximation and Standard Log-Add approximation model sets on Lynx Helicopter additive noise-corrupted RM at 18dB 81
8.3 Average KL number for clean model set and DPMC-compensated model sets using Diagonal, Pseudo, Extended and Standard Diagonal covariance approximations on Lynx Helicopter additive noise-corrupted RM at 18dB . 82
8.4 Average KL number for clean model set and DPMC-compensated model set using the Diagonal, Pseudo and Extended covariance approximations on Operations Room additive noise-corrupted RM at 18dB 86
8.5 Average word error rate (%) on the Feb'89, Oct'89 and Feb'91 test sets against number of adaptation sentences using DPMC with the Diagonal covariance approximation on Lynx Helicopter additive and convolutional noise-corrupted RM at 18dB. 89

9.1 Average KL number for clean, DPMC mean-and-variance-compensated and mean-compensated model sets on Level 3 ARPA Spoke 10 evaluation noise 97

List of Tables

3.1 Different noise condition regions for noise masking function 26
3.2 Probability distributions assumed for various noise conditions and masking functions 27

5.1 Number of parameters per-mixture component for various covariance approximations 57

7.1 Word error rates (%) of the baseline and matched model sets: NOISEX-92 Lynx Helicopter additive noise only 67
7.2 Word error rates (%) of the PMC-compensated model sets: NOISEX-92 Lynx Helicopter additive noise only 69
7.3 Word error rates (%) of the baseline and standard PMC-compensated model sets: NOISEX-92 Lynx Helicopter additive and convolutional noise 69
7.4 Word error rates (%) against number of general speech model components used for estimating the convolutional noise: NOISEX-92 Lynx Helicopter additive and convolutional noise 71
7.5 Word error rates (%) against number of digits in the tilt estimation data: NOISEX-92 Lynx Helicopter additive and convolutional noise 71

8.1 Average SNR for the RM test sets adding Lynx Helicopter noise attenuated by 20dB 74
8.2 Word error rates (%) on clean RM Feb'89 test set for various parameterisations 75
8.3 Word error rates (%) of a six component model set on clean RM 75
8.4 Word error rates (%) compensating various parameter groups using single-pass training on additive Lynx noise-corrupted RM at 18dB 76
8.5 Word error rates (%) using single-pass and multi-pass training on Lynx Helicopter additive noise-corrupted RM at 18dB 77
8.6 Word error rates (%) using single-pass training on Lynx Helicopter noise-corrupted additive noise-corrupted RM at 10dB 78
8.7 Word error rates (%) of DPMC-compensated model sets with various numbers of "observations" on Feb'89 test set with Lynx Helicopter additive noise-corrupted RM at 18dB 80
8.8 Word error rates (%) of various PMC-compensated model sets on Lynx Helicopter additive noise-corrupted RM at 18dB 80
8.9 Word error rates (%) compensating means and variance, and just means, using DPMC on the Lynx Helicopter additive noise-corrupted RM Feb'89 test set 82
8.10 Word error rates (%) using various DPMC covariance approximations on Lynx Helicopter additive noise-corrupted RM at 18dB 83

8.11 Word error rates (%) using various DPMC covariance approximations on Lynx Helicopter additive noise-corrupted RM at 10dB 84

8.12 Word error rates (%) using iterative PMC and the Extended covariance approximation on Lynx Helicopter additive noise-corrupted RM Feb'89 test set 84

8.13 Word error rates (%) using various DPMC covariance approximations on Operations Room additive noise-corrupted RM at 18dB, * indicates some sentences ran out of hypotheses 86

8.14 Word error rates (%) of DPMC mean-compensated and mean-and-variance-compensated model sets on Operations Room additive noise-corrupted RM at 18dB 86

8.15 Word error rates (%) of clean and DPMC-compensated model sets on Machine Gun noise-corrupted RM at 4dB, *indicates some sentences ran out of hypotheses 87

8.16 Word error rates (%) of iterative DPMC- compensated model sets on Machine Gun noise-corrupted RM at 4dB. 87

8.17 Word error rates (%) of DPMC-compensated model sets on Lynx Helicopter additive and convolutional noise-corrupted RM at 18dB. 89

8.18 Word error rates (%) with per-speaker convolutional noise estimation on Lynx Helicopter additive and convolutional noise-corrupted RM at 18dB, * indicates that tilt compensation was performed on a speaker level and † indicates the use of the iterative scheme. 90

8.19 Word error rates (%) of DPMC-compensated model sets on Lynx Helicopter additive and convolutional noise-corrupted RM at 10dB, * indicates that tilt compensation was performed on a speaker level and † indicates the use of the iterative scheme. 91

8.20 Word error rates (%) of iterative DPMC-compensated model sets with the Extended covariance approximation on Lynx Helicopter additive and convolutional noise-corrupted RM at 10dB, * indicates that tilt compensation was performed on a speaker level and † indicates the use of the iterative scheme. 91

9.1 Word error rates (%) of PMC-compensated model sets on Car3 ARPA 1994 CSRNAB Spoke 10 development data, * indicates that the results were generated by re-scoring the Diagonal mean-compensated lattices. 95

9.2 Word error rates (%) of the IBM system on Car3 ARPA 1994 CSRNAB Spoke 10 development data. 96

9.3 Word error rates (%) of PMC-compensated model sets on ARPA 1994 Spoke 10 evaluation data, * indicates that the results were generated by re-scoring the Diagonal mean-compensated lattices. 96

9.4 Word error rates (%) of PMC-compensated and matched model sets on the ARPA 1994 CSRNAB Spoke 10 evaluation data and adaptation-noise-corrupted evaluation data. 98

9.5 Word error rates (%) of PMC-compensated model sets with simple speaker adaptation, using either 10 or 1 adaptation sentences, on ARPA 1994 CSRNAB Spoke 10 evaluation data, * indicates that tilt compensation was performed on a speaker level and † indicates the use of the iterative scheme. 99

9.6 Word error rates (%) of other sites on Car1 WSJ Spoke 10 1994 evaluation data, * indicates that one sentence failed to give a result and resulted in all deletions for that sentence. 100

Notation

The following notation has been used throughout this work.

$\mathbf{y}(\tau)$	a general speech observation vector at time τ.
$\mathbf{Y}_T$	a series of T speech observation vectors.
$\mathbf{O}^c(\tau)$	a corrupted-speech observation in the Cepstral domain at time τ.
$\mathbf{S}^c(\tau)$	a speech observation in the Cepstral domain at time τ.
$\mathbf{N}^c(\tau)$	a noise observation in the Cepstral domain at time τ.
$\mathbf{H}^c$	the time invariant convolutional noise in the Cepstral domain.
$S_i^c(\tau)$	the i^{th} element of the speech observation in the Cepstral domain.

The delta and delta-delta parameters will be denoted by Δ and Δ^2 respectively, thus

$\mathbf{\Delta O}^c(\tau)$	a corrupted-speech delta observation in the Cepstral domain at time τ.
$\mathbf{\Delta}^2\mathbf{O}^c(\tau)$	a corrupted-speech delta-delta observation in the Cepstral domain at time τ.

The superscript is used to denote the domain, thus

$\mathbf{O}^l(\tau)$	a corrupted-speech observation in the Log-Spectral domain at time τ.
$\mathbf{O}(\tau)$	a corrupted-speech observation in the Linear-Spectral domain at time τ.

Related to each "observation" is a random variable (RV) thus

$\mathbf{O}^c$	the RV associated with the corrupted-speech observation in the Cepstral domain.
$\mathbf{S}^c$	the RV associated with the speech observation in the Cepstral domain.
$\mathbf{N}^c$	the RV associated with the noise observation in the Cepstral domain.

In addition associated with each "observation" there will be a distribution

μ^c	the mean of the clean speech.
$\mathbf{\Sigma}^c$	the covariance matrix of the clean speech.
$\tilde{\mu}^c$	the mean of the noise.
$\tilde{\mathbf{\Sigma}}^c$	the covariance matrix of the noise.
$\hat{\mu}^c$	the estimated mean of the corrupted speech.
$\hat{\mathbf{\Sigma}}^c$	the estimated covariance matrix of the corrupted speech.

Distributions may be collected at time $\tau - w$, not τ. This is denoted by $'$, thus

μ'^c	the mean of the clean speech at time $\tau - w$.
$\mathbf{\Sigma}'^c$	the covariance matrix of the clean speech at time $\tau - w$.

Chapter 1

Introduction

Increasingly, as they are applied in real world applications, speech recognition systems must operate in situations where it is not possible to control the acoustic environment. This may result in a serious mismatch between the training and test conditions, which often causes a dramatic degradation in performance of these systems. The aim of the work presented in this thesis is to make automatic speech recognition systems robust to these environmental differences.

Speech can be characterised by a slowly changing spectral envelope. This spectral envelope is perceived by humans and translated into words and their associated meaning. Automatic speech recognition attempts to emulate part of this task, that of mapping the spectral envelope into a series of words. There are many problems associated with this process. Not all people speak the same. The spectral envelope will vary due to regional accents and differences in the individual, for example whether male or female and their height. Furthermore, an individual speaker may utter a given sentence in a wide variety of ways. This may be due to speaker stress, such as when shouting, or by the speaker deliberately emphasising words to alter the meaning. Humans have little difficulty coping with all these variations, however, the design of an automated system that mimics this process is a major challenge.

Most state-of-the-art speech recognition systems use statistical approaches to cope with the many variations. The performance of these recognition systems has gradually increased over the years to a stage where today, on an unlimited vocabulary speaker-independent task, performance figures of over 90% word accuracy can be obtained, and on smaller vocabulary tasks this figure rises to above 95%. At this level of performance the systems are usable, however the majority of them are both trained and tested in identical, quiet environments. In practical systems, it is rarely quiet and there is usually little control over the acoustic environment. There may be differences in both the background noise, for example fans running or cars passing by, and the channel conditions, that is the microphone or telephone channel being used. As the background noise changes and the channel conditions vary, so the speech spectra are altered and there is a resulting, often dramatic, deterioration in the performance of the system. This problem of environmental robustness must be addressed before any practical, easy to use, systems can be developed.

1.1 Speech Recognition

Given a speech waveform, the task of a speech recognition system is to produce an estimate of the word string associated with that waveform. Over the years a variety of approaches to designing speech recognisers have been adopted [94]. They may be split into four broad classes:

1. Template-based. A set of reference templates of the speech pattern are stored. Recognition is carried out by matching an unknown utterance to one of these templates. These techniques make use of Dynamic Programming (DP) algorithms and have been applied to both speaker-dependent and speaker-independent tasks [79]. However, the use of templates limits the ability of the system to model the wide variabilities present in the speech signal. Over the years this has led to the decline of these techniques in the field of speech recognition.

2. Knowledge-based. The use of templates for speech recognition does not make use of linguistic and phonetic knowledge about speech. In knowledge based approaches the aim is to incorporate knowledge of human speech perception into the recognition process [103]. This knowledge is derived from spectrograms and is incorporated in the form of rules. These approaches have had limited success due to the problems of quantifying expert knowledge, and of integrating the various knowledge levels together.

3. Stochastic. Probabilistic models are used to model the uncertainty of the speech signal. These approaches are described in more detail later in this section.

4. Connectionist. Here, a set of simple computing units are used to store knowledge about the speech signal. A wide range of topologies have been examined for speech recognition, ranging from simple multi-layer perceptrons to time delay neural nets to recurrent neural nets. They have also been used, with some success, in conjunction with Hidden Markov Models (HMMs) in hybrid schemes [11].

At present the most popular approach is a stochastic one, in particular the use of HMMs. This is the approach considered in this work.

Having chosen a stochastic framework the basic decision rule used is

$$\hat{\mathbf{W}} = \arg\max_{\mathbf{W}} \left[p(\mathbf{W}|\mathbf{Y}_T)\right] = \arg\max_{\mathbf{W}} \left[\frac{p(\mathbf{W})p(\mathbf{Y}_T|\mathbf{W})}{p(\mathbf{Y}_T)}\right] \tag{1.1}$$

where $\mathbf{Y}_T$ is the observed acoustic data, $p(\mathbf{W})$ is the a-priori probability of a particular word string, $p(\mathbf{Y}_T|\mathbf{W})$ is the probability of the observed acoustic data given the acoustic models, and $\hat{\mathbf{W}}$ is the hypothesised word string. There are three main sources of information being used; the language model, the lexicon and the acoustic model. These are shown in figure 1.1.

As previously mentioned, the acoustic model considered in this thesis is the HMM and is described in detail in Chapter 2.

The lexicon is used to map whatever abstract form the acoustic models have been chosen to represent into the words present in the vocabulary and language model. For small vocabulary tasks the lexicon is usually very simple with a one-to-one mapping from

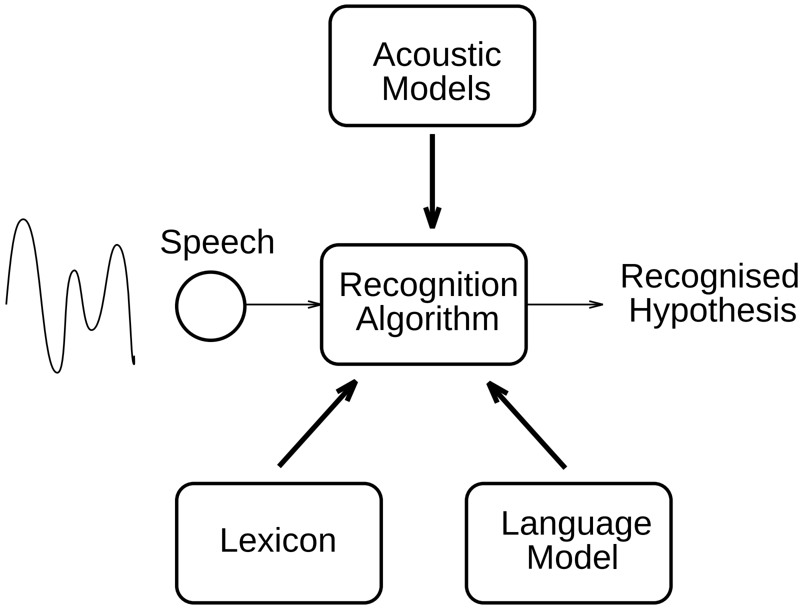


Figure 1.1: General speech recognition system

acoustic model to word. In larger vocabulary tasks, where the acoustic models may represent sub-word units, the lexicon dictates how these sub-word units are linked together to form individual words. For example, the word *speech* may be broken up as

$$\text{speech} = \{\mathbf{s\ p\ iy\ ch}\}$$

where phones[1] are the sub-word unit.

The language model contains the information about which word sequences are allowable. This may include the probability of such sequences. It can dramatically reduce the computational load and improve the accuracy of the recogniser by limiting the word search space. Various language models may be used, for example *word-pair* grammars, which simply limit the set of which words may follow the current word, or stochastic language models, such as *N-grams*, which associate probabilities with each word sequence.

N-gram language models [69] are commonly used in HMM-based speech recognition systems. The probability of a particular word sequence may be written as

$$p(\mathbf{W}_1^n) = p(W_1)p(W_2|W_1)\ldots p(W_n|\mathbf{W}_1^{n-1}) \tag{1.2}$$

where $\mathbf{W}_1^n$ is the partial string of words from W_1 to W_n. It is not feasible to train all probabilities for all possible partial word sequences. A simplifying assumption is therefore made that the probability of the current word is only dependent on the previous $N-1$ words, not the complete history. These are N-gram language models and the language model probability is given by

$$p(\mathbf{W}_1^n) \approx \prod_{k=1}^{n} p(W_k|\mathbf{W}_{k-N+1}^{k-1}) \tag{1.3}$$

[1]The term *phone* is used here to avoid heated debate about what constitutes a *phoneme*, the Concise Oxford Dictionary (1982) definition of which is "a unit of significant sound in a specified language."

The most commonly used forms of N-grams are trigram ($N = 3$), bigram ($N = 2$) and unigram ($N = 1$) language models.

1.2 Speech Recognition in Adverse Environments

It has been well documented that variations in acoustic environment and stress related variability seriously degrade current speech recognition systems [22]. There are many "noise" sources that may affect the speech signal.

Firstly, there is the problem of the acoustic environment in which the system is operating. There are a wide range of possible noise sources that are uncorrelated with the speech. These may be short lived and non-stationary, such as cars passing, doors slamming, competing speech; or continuous such as fans and air-conditioning units. Speech correlated noises will also impact upon the system performance. These result from reflection and reverberation. In addition to the direct effects of the acoustic environment, speakers will alter their speech pattern in response to stress or background noise, the Lombard effect [44]. Finally, it is necessary to consider the input equipment. The type of microphone and transmission channel, particularly if limited to telephone bandwidth, will affect the speech signal.

A general framework based on the system described in [32] considers all the noise sources independently. The observed corrupted-speech signal, $y(t)$, is related to the "normal" noise-free distortionless speech signal, $s(t)$, by

$$y(t) = \left[\left\{\left(\left[s(t)|^{Stress}_{Lombard}\right]_{n_1(t)} + n_1(t)\right) * h_{mike}(t) + n_2(t)\right\} * h_{chan}(t)\right] + n_3(t) \qquad (1.4)$$

where $n_1(t)$ is the background noise, $h_{mike}(t)$ is the impulse response of the microphone, $n_2(t)$ and $h_{chan}(t)$ are the additive noise and impulse response, respectively, of the transmission channel, and $n_3(t)$ is the noise present at the receiver.

The performance of speech recognition systems is known to degrade as a talker's speech characteristics change. This change may result from the speaker being in a high noise environment, the Lombard effect, or from psychological stress. There are various forms of psychological stress. For example, the user may be performing a highly demanding problem, leaving speech as a secondary task, or may be tired. In equation 1.4 this is indicated by conditioning the speech signal on stress, workload and the additive noise environment. For the work presented in this thesis, the Lombard effect and stress are not taken into account, as in normal office conditions they are not usually the dominant effects [14]. A description of some work related to the Lombard effect and stress compensation, and further references, are given in [33].

By ignoring the effects of speaker stress, it is possible to simplify equation 1.4. The various additive noise sources and channel conditions may be combined into composite non-specific sources. Figure 1.2 shows such a system and how the combined channel distortion, convolutional noise from the channel, and the composite additive noise are assumed to affect the clean speech. The additive noise is shown after the channel difference, as in practical systems the only noise estimate available is observed in the test channel condition. It is, therefore, not important at what stage the noise was generated, whether ambient noise, channel noise etc., nor what channel conditions it is transmitted over. The observed signal is now

$$y(t) = s(t) * h(t) + n(t) \qquad (1.5)$$

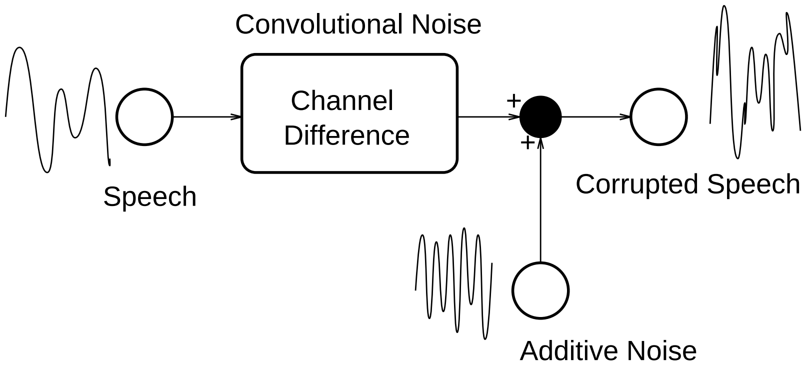


Figure 1.2: Simplified distortion framework

where $n(t)$ is the total additive noise and $h(t)$ is the channel mismatch. The additive noise component in equation 1.5 is also considered to be uncorrelated with the speech signal. The correlated aspects of the noise, due to reverberation etc., are subsumed in the convolutional noise component. For all the work presented here, this is the form of the acoustic environment considered.

1.3 Organisation of Thesis

This thesis is organised as follows. The next chapter, **Chapter 2**, describes the HMM, how it is trained and used in the speech recognition task. **Chapter 3** describes examples of each of the main classes of noise compensation techniques that have previously been employed. The following chapter, **Chapter 4**, illustrates the effects that additive noise has on the probability distributions associated with the corrupted speech in idealised circumstances. The main contribution of this thesis is described in **Chapter 5**, which details the aims and various implementations of Parallel Model Combination. **Chapter 6** describes the various noise sources considered in the experimental chapters, Chapters 7, 8 and 9. These chapters are organised in order of increasing vocabulary size. **Chapter 7** describes the NOISEX-92 experiments, **Chapter 8** the DARPA Resource Management based task, and **Chapter 9** the ARPA 1994 CSRNAB Spoke 10. Finally **Chapter 10** draws overall conclusions and describes possible future work.

Chapter 2

Hidden Markov Model Speech Recognition Systems

For all the work described in this thesis the HMM is used as the basic acoustic model. This chapter describes the basic HMM, how it is trained and applied to the speech recognition task, and the speech parameters commonly used with HMM-based speech recognition systems.

2.1 The Hidden Markov Model

The HMM is the most popular and successful stochastic approach to speech recognition in general use. This is due to the existence of elegant and efficient algorithms for both training and recognition. The HMM, the acoustic model, is required to determine, in conjunction with the language model, the most likely word sequence given some speech data. Specifically within this process, the acoustic model is required to give the probability of each possible word sequence.

The basic theory for HMMs is presented here. A more detailed discussion of HMM theory and application for speech recognition is given by Rabiner [78]. The use of HMMs for speech recognition is dependent on certain assumptions:

1. Speech may be split into segments, states, in which the speech waveform may be considered to be stationary. The transition between these states is assumed to be instantaneous.

2. The probability of a certain "observation" being generated is only dependent on the current state, not on any previously generated symbols. This is a first-order Markov assumption and is usually referred to as the *independence assumption*.

Neither of these assumptions are true for the speech signal. There has been much research into how to reduce the effects of these crude assumptions [6, 75], particularly the independence assumption. However to date, the standard HMM is still used in most current speech recognisers.

A three-emitting-state HMM is shown in figure 2.1. This HMM has a topology whereby transitions may only be made to the same state and the next state, that is no "skips" are allowed. This simple left-to-right topology is now virtually standard and it is the one used throughout this work, though the number of states in the model may vary.

HMMs are characterised by:

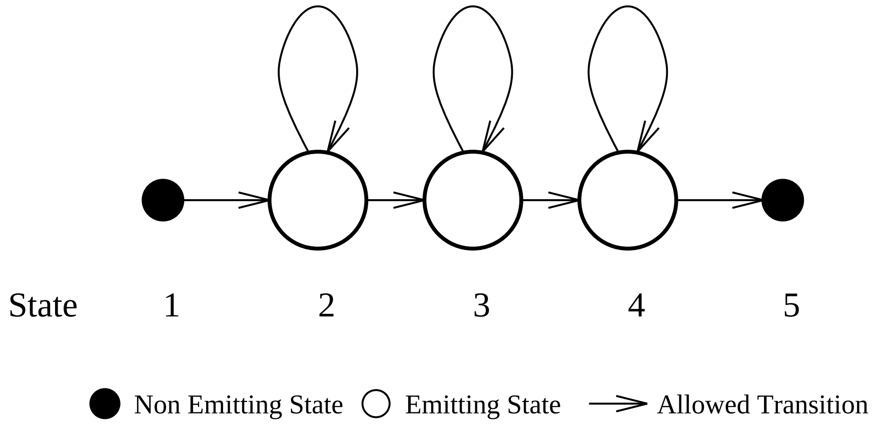


Figure 2.1: A 3-emitting state hidden Markov model

1. N, the number of states in the model. The states are denoted as $\mathbf{s} = \{s_1, s_2, \ldots, s_N\}$ and $q_i(\tau)$ indicates "being in" state s_i at time τ. For the multiple Gaussian mixture component case an extended form of this notation is used, $q_{im}(\tau)$, which indicates being in mixture component M_m of state s_i at time τ.

2. $\mathbf{A}$, the state probability transition matrix, where

$$a_{ij} = p(q_j(\tau+1)|q_i(\tau)) \tag{2.1}$$

   This matrix does not have to be full, whereby all transitions are permitted, as shown in figure 2.1. The transition matrix contains the only temporal information associated with the HMM and, as such, is normally constrained to be left-to-right. Given the definition of the transition matrix, its elements must obey the constraint that

$$\sum_{j=1}^{N} a_{ij} = 1 \tag{2.2}$$

3. $\mathbf{B}$, the output probability distribution associated with each emitting state, where

$$b_j(\mathbf{y}(\tau)) = p(\mathbf{y}(\tau)|q_j(\tau)) \tag{2.3}$$

   $\mathbf{y}(\tau)$ is the feature vector at time τ. There are two general types of HMM, split according to the form of their output distribution. If the output distribution is based on discrete elements then the models are called Discrete HMMs (DHMMs). Alternatively if the output distribution is continuous they are referred to as Continuous-Density HMMs (CDHMMs). CDHMMs are the only output distributions considered in this work.

4. π, the initial state distribution, where

$$\pi_i = p(q_i(1)) \tag{2.4}$$

All standard HMMs are described by the above parameters. However, the topology used in this thesis, in common with standard HMM Toolkit (HTK) terminology [99], utilises non-emitting states, states s_1 and s_N, as shown in figure 2.1. The HMM is now constrained to start in state s_1 and the traditional initial state distribution, π, simply becomes the first row of the transition matrix. Furthermore, the model must terminate in state s_N. These non-emitting states are included to make extensions from the isolated to the continuous speech recognition case simple.

Although only CDHMMs are considered in this work, some of the noise compensation techniques developed in this thesis, which don't use the variances, such as the Log-Add approximation described in section 5.6.3, could be applied to DHMMs. However to date, this has not been investigated.

Usually in CDHMMs, the output probabilities are modelled by multivariate Gaussian distributions or a mixture of multivariate Gaussian distributions. For the multiple Gaussian mixture component case the "probability"[1] is given by

$$b_j(\mathbf{y}(\tau)) = \sum_{m=1}^{M} c_{jm}\mathcal{N}(\mathbf{y}(\tau); \mu_{jm}, \mathbf{\Sigma}_{jm}) \tag{2.5}$$

where $\mathcal{N}(\mathbf{y}(\tau); \mu_{jm}, \mathbf{\Sigma}_{jm})$ is a multi-variate Gaussian distribution, with mean vector μ_{jm} and covariance matrix $\mathbf{\Sigma}_{jm}$, each mixture component having an associated weight c_{jm}. An extension to this notation indicates the probability of an observation given a particular mixture component, M_m, of a state s_j,

$$b_{jm}(\mathbf{y}(\tau)) = \mathcal{N}(\mathbf{y}(\tau); \mu_{jm}, \mathbf{\Sigma}_{jm}) \tag{2.6}$$

For the case of CDHMMs, $\mathbf{B}$ will consist of a set of means, variances and mixture component weights. Non-Gaussian distributions have been investigated [29] for the output probability distributions. Currently, Gaussians distributions are normally used in HMMs and are the output distributions considered in this work. However, many of the techniques described may be applied to non-Gaussian distributions.

For convenience the following compact notation will be used to represent a HMM

$$\mathcal{M} = (\mathbf{A}, \mathbf{B}) \tag{2.7}$$

2.2 Pattern Matching with HMMs

The HMM, as the acoustic model, is required to determine the probability of the observations data, $\mathbf{Y}_T$, given a hypothesised word string, $\mathbf{W}$. The word string is mapped to the appropriate set of models using the lexicon, so now the task is to find $p(\mathbf{Y}_T|\mathcal{M})$ where $\mathcal{M}$ is the set of HMMs linked with word string $\mathbf{W}$. Since CDHMMs are being considered, the term *likelihood*[2] will be used instead of probability, so the aim is to find $\mathcal{L}(\mathbf{Y}_T|\mathcal{M})$ where $\mathcal{L}(.)$ denotes the likelihood.

---

[1]The term "probability" is in quotes as CDHMMs use Probability Density Functions (PDFs), so the values are strictly densities not probabilities.

[2]In [66] the likelihood, as used by "careful writers", is defined as "the probability that a model with particular parameter values assigns to the data that has actually been observed". Thus

$$\mathcal{L}(\mathcal{M}|\mathbf{Y}_T) = \prod_{\tau=1}^{T} p(\mathbf{y}(\tau)|\mathcal{M}) \tag{2.8}$$

As this definition, particularly when language models are used, would add little to understanding of the work presented here and would confuse the simple statistical decision rule defined in equation 1.1, the term likelihood is far more loosely defined as the "probability" of the observed data given by the particular model parameter set.

Initially only the simplified case of a single model, $\mathcal{M}$, will be examined. The observation sequence is defined as

$$\mathbf{Y}_T = \mathbf{y}(1)\mathbf{y}(2)\ldots\mathbf{y}(T) \tag{2.9}$$

where each observation $\mathbf{y}(\tau)$ is an n-dimensional vector

$$\mathbf{y}(\tau) = \left[\begin{array}{cccc} y_1(\tau) & y_2(\tau) & \ldots & y_n(\tau) \end{array}\right]^T \tag{2.10}$$

The total likelihood is given by summing over all possible paths through the model that end at the appropriate final state, in this case s_N. Hence, the likelihood of the observation sequence and the particular model is given by

$$\begin{aligned} \mathcal{L}(\mathbf{Y}_T|\mathcal{M}) &= \sum_{\theta\in\mathbf{\Theta}} \mathcal{L}(\mathbf{Y}_T|\theta,\mathcal{M})\mathcal{L}(\theta|\mathcal{M}) \\ &= \sum_{\theta\in\mathbf{\Theta}} a_{\theta_T N} \prod_{\tau=1}^{T} a_{\theta_{\tau-1}\theta_\tau} b_{\theta_\tau}(\mathbf{y}(\tau)) \end{aligned} \tag{2.11}$$

where $\mathbf{\Theta}$ is the set of all K possible state sequences of length T in model $\mathcal{M}$,

$$\mathbf{\Theta} = \left\{\theta^{(1)},\ldots,\theta^{(K)}\right\} \tag{2.12}$$

and θ_τ is the state occupied at time τ in path θ. The model is initialised such that $\theta_0 = 1$. Equation 2.11 results directly from the definition of the HMM and the independence assumption previously described.

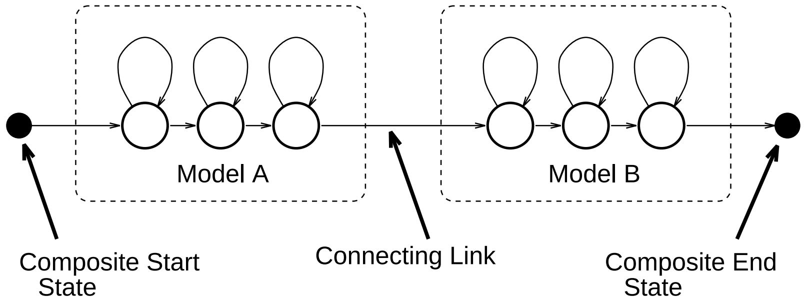


Figure 2.2: Composite hidden Markov model

So far only a single model has been considered. For continuous speech, or where subword models are used, many models may be required to be connected together to form the word string. This extension is achieved by linking models together to form a composite model, figure 2.2. The end, non emitting, state, of model A and the start state of model B have been removed and replaced by a connecting link. The start state of model A has now become the composite start state and the end state of model B the composite end state. The set of possible states, $\mathbf{\Theta}$, in equation 2.11, is now the set of all paths of length T in this composite model AB.

Computationally equation 2.11 is very expensive if implemented directly. Fortunately, an efficient algorithm exists for this calculation, the *forward-backward* algorithm.

2.3 The Forward-Backward Algorithm

The forward-backward algorithm is an efficient method for calculating the likelihood of an observation sequence being generated by a particular set of models. Given the first-order Markov assumption used in the standard HMM, that the probability of a particular observation is only dependent on the current state, it is unimportant, in terms of the likelihood, what the previous state sequence was. All that is important is the likelihood of generating an observation given a particular state and the likelihood of being in that state at the current time instance. This is the basis of the forward-backward algorithm.

Two new variables are introduced, the *forward probability* $\alpha_j(t)$ and the *backward probability* $\beta_j(t)$. These are defined as

$$\begin{aligned}\alpha_j(t) &= \mathcal{L}(\mathbf{y}(1), \ldots, \mathbf{y}(t), q_j(t)|\mathcal{M}) &(2.13)\\ \beta_j(t) &= \mathcal{L}(\mathbf{y}(t+1), \ldots, \mathbf{y}(T)|q_j(t), \mathcal{M}) &(2.14)\end{aligned}$$

Using these definitions and the previous HMM assumptions, it is possible to derive iterative formulae to obtain values for $\alpha_j(t)$ and $\beta_j(t)$. Firstly it is necessary to define the initial conditions. From the HMM definition, the start conditions for $\alpha_j(0)$ are

$$\begin{array}{l}\alpha_1(0) = 1\\ \alpha_j(0) = 0 \quad \text{if } j \neq 1\end{array} \tag{2.15}$$

Then for $1 \leq t \leq T$ and $2 \leq j \leq N-1$

$$\alpha_j(t) = \left[\sum_{i=1}^{N} \alpha_i(t-1)a_{ij}\right] b_j(\mathbf{y}(t)) \tag{2.16}$$

and terminates with

$$\alpha_N(T) = \sum_{i=2}^{N-1} \alpha_i(T)a_{iN} \tag{2.17}$$

A similar set of recursions may be defined for the backward probability. The initial conditions are now

$$\beta_j(T) = a_{jN} \quad \text{for } 1 \leq j \leq N \tag{2.18}$$

and the iterative scheme from $t = T-1$ to $t = 0$ is

$$\begin{array}{l}\beta_i(t) = \sum_{j=1}^{N} a_{ij}b_j(\mathbf{y}(t+1))\beta_j(t+1) \quad \text{for } 1 \leq j < N\\ \beta_N(t) = 0\end{array} \tag{2.19}$$

The likelihood of a particular observation sequence is given by

$$\mathcal{L}(\mathbf{Y}_T|\mathcal{M}) = \alpha_N(T) = \beta_1(0) = \sum_{j=1}^{N} \alpha_j(t)\beta_j(t) \tag{2.20}$$

In addition to yielding the likelihood of a particular utterance, these recursions are important in the re-estimation formulae for HMMs.

2.4 Estimation of HMM Parameters

The estimation of the parameters of the HMM set is a large multi-dimensional optimisation problem. The task is to obtain a set of models that, according to an appropriate criterion, matches the available training data well. There are many criteria that may be chosen to be optimised, however the most popular, due to the existence of very efficient training algorithms, is the Maximum Likelihood (ML) criterion. For a ML estimator to be good, in the sense of a consistent estimator with the lowest variance[3], it must satisfy certain criteria [13]:

1. The observations are from the assumed family of distributions.
2. The assumed family of distributions is "well behaved"[4].
3. The number of observations is large enough to use asymptotic theory.

If, in addition, the performance of the system cannot get worse as its parameters get closer to the true estimates, then a ML estimator will yield a "good" decoder. These assumptions are not all valid in the case of the standard HMM-based speech recogniser, however ML estimation has been found in practice to give consistently good performance. Other criteria have been examined, such as Maximum Mutual Information (MMI) [13], but are not considered in this work.

The aim in ML estimation is to obtain the set of HMMs, $\mathcal{M}$, such that they maximise $\mathcal{G}_{mle}(\mathcal{M})$ where

$$\mathcal{G}_{mle}(\mathcal{M}) = \mathcal{L}(\mathbf{Y}_T|\mathcal{M}) \tag{2.21}$$

This problem may be dealt with as a standard multi-dimensional optimisation task. However, the dimensionality of this problem (consider the total number of means, variances and weights in a large vocabulary system) is very large, which may result in slow training times. To overcome this problem an *Expectation-Maximisation* (EM) technique [17] is used. For the particular application of training HMMs, this is commonly referred to as the *Baum-Welch* algorithm [8].

The Baum-Welch algorithm is designed such that

$$\mathcal{G}_{mle}(\hat{\mathcal{M}}) \geq \mathcal{G}_{mle}(\mathcal{M}) \tag{2.22}$$

where $\hat{\mathcal{M}}$ is the new estimate of the model set. This is achieved by introducing an auxiliary function, $\mathcal{Q}(\mathcal{M}, \hat{\mathcal{M}})$, defined as

$$\mathcal{Q}(\mathcal{M}, \hat{\mathcal{M}}) = \sum_{\theta \in \mathbf{\Theta}} \mathcal{L}(\mathbf{Y}_T, \theta|\mathcal{M}) \log\left(\mathcal{L}(\mathbf{Y}_T, \theta|\hat{\mathcal{M}})\right) \tag{2.23}$$

It can be shown that maximising this auxiliary function ensures $\mathcal{G}_{mle}(\mathcal{M})$ is non-decreasing, satisfying equation 2.22. Equation 2.23 may be used many times, each time replacing the original model set by the new model set estimate. Each iteration is guaranteed not to

[3] In [84] a "good" estimator is described as one yielding a minimum variance unbiased estimate. However, ML estimates are not necessarily unbiased, for example the estimate of the variance in a Gaussian distribution, but are consistent provided that the observations are independent and identically distributed.

[4] Here, "well behaved" means that the distribution tails off rapidly enough and the range of the distribution, where it is non zero, is not a function of the distribution parameters (see [84] for details).

decrease the likelihood. This will eventually result in a local ML estimate of the parameters. Baum first proved the convergence of this, and it was later extended to mixture distributions and vector observations [40, 52].

For CDHMMs with Gaussian output distributions, simple re-estimation formulae exist to iteratively estimate the model parameters. Many references exist for the derivation of these formulae [40, 52], only the results are quoted here. The probability of being in a particular state mixture component is given by

$$\begin{aligned} L_{jm}(\tau) &= p(q_{jm}(\tau)|\mathbf{Y}_T, \mathcal{M}) \\ &= \frac{1}{\mathcal{L}(\mathbf{Y}_T|\mathcal{M})} U_j(\tau) c_{jm} b_{jm}(\mathbf{y}(\tau)) \beta_j(\tau) \end{aligned} \tag{2.24}$$

where

$$U_j(\tau) = \begin{cases} a_{1j}, & \text{if } \tau = 1 \\ \sum_{i=2}^{N-1} \alpha_i(\tau-1) a_{ij}, & \text{otherwise} \end{cases} \tag{2.25}$$

The extended dataset $\{\mathbf{y}(1), \ldots, \mathbf{y}(T), \mathbf{L}(1), \ldots, \mathbf{L}(T)\}$, where $\mathbf{L}(\tau)$ is the matrix whose elements $L_{jm}(\tau)$ are the probability of being in state s_j and component M_m at time τ, is called the complete dataset. Throughout the descriptions of the compensation techniques the terms *frame/state alignment* and *frame/state component alignment* will be used to denote $L_j(\tau)$, where $L_j(\tau) = p(q_j(\tau)|\mathbf{Y}_T, \mathcal{M})$, and $L_{jm}(\tau)$ respectively. Where just the frame/state alignment is assumed to be unaltered, the value of $L_j(\tau)$ remains the same. If the frame/state component is assumed unaltered then $L_{jm}(\tau)$ remains the same. Using this extended dataset, the parameter estimation has simple closed-form expressions. The mean, variances and mixture weights are given by

$$\hat{\mu}_{jm} = \frac{\sum_{\tau=1}^{T} L_{jm}(\tau) \mathbf{y}(\tau)}{\sum_{\tau=1}^{T} L_{jm}(\tau)} \tag{2.26}$$

$$\hat{\mathbf{\Sigma}}_{jm} = \frac{\sum_{\tau=1}^{T} L_{jm}(\tau)(\mathbf{y}(\tau) - \hat{\mu}_{jm})(\mathbf{y}(\tau) - \hat{\mu}_{jm})^T}{\sum_{\tau=1}^{T} L_{jm}(\tau)} \tag{2.27}$$

$$\hat{c}_{jm} = \frac{\sum_{\tau=1}^{T} L_{jm}(\tau)}{\sum_{\tau=1}^{T} L_j(\tau)} \tag{2.28}$$

The transition probabilities are re-estimated by

$$\hat{a}_{ij} = \frac{\sum_{\tau=1}^{T-1} \alpha_i(\tau) a_{ij} b_j(\mathbf{y}(\tau+1)) \beta_j(\tau+1)}{\sum_{\tau=1}^{T-1} \alpha_i(\tau) \beta_i(\tau)} \tag{2.29}$$

where $1 < i < N$ and $1 < j < N$. The transitions from and to the non-emitting states are re-estimated by

$$\hat{a}_{1j} = \frac{1}{\mathcal{L}(\mathbf{Y}_T|\mathcal{M})}\alpha_j(1)\beta_j(1) \tag{2.30}$$

and

$$\hat{a}_{iN} = \frac{\alpha_i(T)\beta_i(T)}{\sum_{\tau=1}^{T}\alpha_i(\tau)\beta_i(\tau)} \tag{2.31}$$

These re-estimation formulae are used to train all the HMMs used in this work[5]. Further details of how multiple component models are built are given in the HTK manual [102].

2.5 Speech Recognition using HMMs

The previous sections have described how to train a set of HMMs given an observation sequence and associated word string. For recognition it is necessary to obtain $\mathcal{L}(\mathbf{Y}_T|\mathcal{M})$ for each possible word string. The forward-backward algorithm could be used to obtain the most likely word sequence, however, in practice the most likely state sequence associated with $\mathbf{Y}_T$ is commonly used. This state sequence is usually obtained using the *Viterbi* algorithm [93].

A new variable $\phi_i(t)$ is introduced.

$$\phi_i(t) = \max_{\theta \in \mathbf{\Theta}_{t-1}} \left[\mathcal{L}(\mathbf{Y}_t, q_i(t), \theta|\mathcal{M})\right] \tag{2.32}$$

where $\mathbf{\Theta}_{t-1}$ is the set of all partial paths of length $t-1$. Values of $\phi_i(t)$ may be efficiently calculated using the following recursive equation.

$$\phi_j(t+1) \quad = \quad \max_{1 \leq i \leq N}\left[\phi_i(t)a_{ij}\right]b_j(\mathbf{y}(t+1)) \tag{2.33}$$

In continuous speech recognition where there is a large number of possible word strings it is not feasible to have a single concatenated model for each word string. Instead a *token passing* framework may be used [100]. In this scheme each state has a token associated with it. This token contains the history of the model sequence that was taken to get to that point and the current value of $\phi_i(t)$. When a new vector is observed these tokens are updated and propagated for each state within the model. The most likely token calculated at the end state of each model is then propagated to all connected models and the history of that token updated.

In most speech recognition systems, it is not practical to propagate all tokens. In order to reduce the computational load, if a path falls below a certain threshold it is removed, or pruned, from the list of possible paths. Many thresholding schemes are possible, the

[5] The re-estimation formulae described here are for a single observation sequence. The extension for multiple observation sequences is shown in the HTK manual [102], but adds little to the understanding of the problems addressed in this work. Furthermore, the transition probabilities are defined for the simple single model case. The extension to the concatenated model case only alters equations 2.30 and 2.31, but again adds little to the understanding of the problems addressed here and the nature of the re-estimation formulae.

most commonly used one is to set the threshold at some fixed value below the current most likely path. This can dramatically reduce the computational load associated with the recognition task, but may introduce *search errors*. These are partial paths that have been pruned at some earlier stage in the recognition process, but were in fact part of the most likely path.

In addition to *substitutions*, where the wrong word is hypothesised, errors in continuous speech recognition systems will result from *deletions*, words left out of the hypothesis, and *insertions*, words added. In order to minimise the total number of errors in the recognition system, a *fixed transition penalty* score is often added. This penalty score is used to set the level of insertions and deletions such that the performance is optimised. In addition to the transition penalty, a *grammar scale factor* is commonly incorporated in the language model. This scale factor is used to "balance" the information of the language model and the acoustic models. Thus, for the case of a bigram language model the likelihood of word W_j following word W_i with associated observed data $\mathbf{Y}_T$, the log-likelihood is given by

$$\log(\mathcal{L}(\mathbf{Y}_T, W_j|W_i)) = \log(\mathcal{L}(\mathbf{Y}_T|W_j)) + s\log(p(W_j|W_i)) + p \tag{2.34}$$

where s is the grammar scale factor and p is the fixed transition penalty.

2.6 HMM Parameter Tying

In small vocabulary systems the basic modelling unit for the HMM is the word. For medium to large vocabulary speech recognition tasks it is not possible to obtain sufficient examples of all the words in the vocabulary to obtain good whole-word models. Hence, it is necessary to *tie* sets of parameters together. Here the term tying means that, for example, two Gaussian mixture components from two different states share the same distribution. Tying may be applied at many levels, ranging from the model level, to the state level, to the Gaussian component level. Which components/states of the recognition system are to be tied is an important question. The most commonly adopted approach is to break the whole word models into sub-word, phone, models. All observations of the same phone are then tied together at the model level. The resulting set of models is referred to as a *monophone* set, or context-independent model set. One of the major advantages of using phones as the sub-word unit is that standard word to phone dictionaries and rules exist, allowing any word to be split into a series of phones.

For many tasks monophone systems do not give sufficiently good performance. The acoustic nature of the current phone is highly dependent on the preceeding and following phones. To model this effect, *triphone* models are commonly used. These are referred to as context-dependent models. A triphone is a model of a phone, given particular preceeding and following phones. For example, for the phone **p** a possible triphone is **s-p+iy**, where the preceeding phone is **s** and the following phone is **iy**. Thus for the example of the isolated word *speech* the triphones are

$$\text{speech} = \{\textbf{sil-s+p s-p+iy p-iy+ch iy-ch+sil}\}$$

Phones may be made dependent upon contexts of an arbitrary number of phones either side of the current phone, however triphones are the most commonly used models and the ones considered in this work.

An additional problem arises when these context-dependent models are used. There may be insufficient data to accurately estimate all the triphones observed in the training data. Moreover, triphones may appear in the test set which do not appear in the training data, particularly if cross-word triphones are used. Various clustering techniques have been applied to solve this problem. Here, triphones that are acoustically similar are tied together. These clustering schemes range from triphone-model clustering [101], which will not handle unseen triphones, to model-based [7] and state-based [99] decision tree clustering, which do.

2.7 Speech Parameterisation

For HMMs to work effectively the speech must be converted from its raw waveform to some feature vector. The first stage is to digitise the signal. This digitised waveform is then converted into a set of feature vectors. These feature vectors should retain all the useful discriminatory speech information, whilst reducing the data rate. Over the years, many forms of speech parameterisation have been considered. At present, most state-of-the-art speech recognition systems are based on Cepstral parameters. The general transformation to map from the digitised waveform to the Cepstral domain[6] is

$$o(t) \overset{\text{FFT}}{\Longrightarrow} \mathbf{o}(f,\tau) \overset{\sum|.|^{\gamma}}{\Longrightarrow} \mathbf{O}(\tau) \overset{\log()}{\Longrightarrow} \mathbf{O}^{l}(\tau) \overset{\mathbf{C}}{\Longrightarrow} \mathbf{O}^{c}(\tau)$$

where $o(t)$ is the sample at time t, $\mathbf{o}(f,\tau)$ is the FFT value at frequency f for vector τ, $\sum|.|^{\gamma}$ indicates taking the magnitude of the FFT values, raising them to a power γ and summing, and $\mathbf{C}$ is the matrix associated with the Discrete Cosine Transform (DCT). The three domains mentioned in this thesis are the Linear-Spectral, Log-Spectral and Cepstral domains.

Throughout this work superscripts are used to describe the domain of the observations or parameters. For example, $\mathbf{O}^{c}(\tau)$ denotes the Cepstral domain, $\mathbf{O}^{l}(\tau)$ the Log-Spectral domain and $\mathbf{O}(\tau)$ the Linear Spectral domain.

2.7.1 The Linear-Spectral and Log-Spectral Domains

The digitised waveform may be split into overlapping frames of speech, typically having period 25ms at 10ms intervals. Pre-emphasis, usually in the form of

$$\hat{o}(t) = o(t) - 0.97o(t-1) \tag{2.35}$$

and a Hamming window, in this work of the form

$$\hat{o}(t) = \left[0.54 - 0.46\cos\left(\frac{2\pi t}{T-1}\right)\right] o(t) \tag{2.36}$$

where T is the length of the frame in question, are applied to the sample waveform. An FFT is then taken of each frame of the digitised speech signal to map from the sample domain into the frequency domain. Normally the magnitude, or power, of each FFT bin is calculated. These are then combined together to reduce the size of the feature vector

[6]The analysis shown here is for filterbank-based Cepstral parameters. It is possible to obtain parameters in the Cepstral domain without going via filterbanks, however in this work, unless otherwise stated, the term Cepstral parameters will refer to those based on filterbank derived features.

using a set of frequency bins. These bins may take many forms, the most popular being to use *mel-spaced* bins. A typical mel-scale used is

$$\mathrm{Mel}(f) = 2595 \log_{10}\left(1 + \frac{f}{700}\right) \tag{2.37}$$

The use of these frequency bins will typically reduce the number of FFT channels from more than 256 to around 20 smoothed frequency estimates. The elements of the Linear-Spectral Domain feature vector are

$$O_b(\tau) = \sum_{f=1}^{F} w_b(f)^{\gamma} |\mathbf{o}(f,\tau)|^{\gamma} \tag{2.38}$$

where $w_b(f)$ is the frequency bin weight associated with bin b, F is the highest frequency component present, and γ determines the domain, normally either power or magnitude, of the mel-frequency bining. The frequency bin weights are normally triangular in shape and applied in the magnitude domain.

The Log-Spectral values are then obtained from

$$O^l_b(\tau) = \log\left(O_b(\tau)\right) \tag{2.39}$$

If within mel-frequency bin variations are ignored, altering the value of γ in equation 2.38 simply results in a scaling of the Log-Spectral value.

2.7.2 The Cepstral Domain

Models may be generated using observations in the Log-Spectral domain. However, in this domain the assumption that feature vector elements are independent of one another is poor. This means that full covariance matrices should be used in the multi-variate Gaussian mixture components, or the lack of correlation modelling will degrade performance. The use of full covariance matrices dramatically increases the number of parameters in the system. Alternatively, models may be generated from observations in the Cepstral domain. There are two main advantages in using Cepstral parameters. Though only approximate, the assumption that the elements of the feature vector are independent is better in the Cepstral domain than in the Log-Spectral domain. This allows diagonal covariance matrices to be used with less degradation in performance. Furthermore, Cepstral parameters are a more compact representation than Log-Spectral parameters [16]. This allows the feature vector to be truncated without degrading performance, thus reducing the number of parameters in the system.

Cepstral parameters are obtained by taking the inverse FFT of the parameters in the Log-Spectral domain. This transformation may be approximated by

$$O^c_i(\tau) = \sum_{b=1}^{B} O^l_b(\tau) \cos(i(b-0.5)\pi/B) \tag{2.40}$$

where B is the number of mel-spaced frequency bins and a DCT is used to perform the inverse fourier transform. This results in Mel-spaced Frequency Cepstral Coefficients (MFCCs). The transformation from the Log-Spectral domain to the Cepstral domain may be represented as a matrix transformation of the Log-Spectral feature vector. The elements of this matrix, $\mathbf{C}$, are given by

$$C_{ij} = \cos(i(j-0.5)\pi/B) \tag{2.41}$$

Data and models may also be mapped back from the Cepstral domain to the Log-Spectral domain using an inverse DCT, denoted by $\mathbf{C}^{-1}$.

Liftering [43] is often used in the Cepstral domain. Various liftering functions may be used, for this work the Cepstral parameters are weighted according to

$$\hat{O}_i^c(\tau) = \left(1 + \frac{L}{2} \sin\left(\frac{i\pi}{L}\right)\right) O_i^c(\tau) \tag{2.42}$$

where L is the liftering coefficient.

2.7.3 Linear Prediction Cepstral Domain

An alternative Cepstral representation of the speech may be obtained via Linear Predictive Coding (LPC) of the speech instead of using the Linear-Spectral domain. Various LPC methods [57] are available, a commonly used one is the least squares autocorrelation method. Here the predictor filter $A(z)$ defined as

$$A(z) = 1 - \sum_{k=1}^{p} a_k z^{-k} \tag{2.43}$$

is optimised to minimise the energy of the residual signal or error,

$$E = \sum_{n=-\infty}^{\infty} e^2(n) \tag{2.44}$$

where

$$e(n) = s(n) - \sum_{k=1}^{p} a_k s(n-k) \tag{2.45}$$

and the error is taken over the window of interest.

Linear Prediction Cepstral Coefficients (LPCCs) may be calculated from these directly [57] as

$$c_n = -a_n + \frac{1}{n} \sum_{i=1}^{n-1} (n-i) a_i c_{n-i} \tag{2.46}$$

where c_i is the i^{th} Cepstral parameter calculated from the LPC parameters.

For the work presented in this thesis LPCC parameters are not considered. However, the Log-Normal approximation, see section 5.6.2, has been applied to compensating these parameters [58].

2.7.4 Dynamic Coefficients

In the speech signal, each observation vector is dependent to some extent on the preceeding one. Thus, the first-order independence assumption that HMMs are usually based on is not valid. For simple tasks, for example small vocabulary speaker-independent tasks, this may not affect the performance of the system, as the words may be sufficiently distinct from one another. As the vocabulary size increases and speaker-independent recognition is required, it is more likely that the words will be confusable and the system performance will drop. To overcome this problem, dynamic coefficients may be incorporated into the feature vector [21]. These coefficients capture some of the correlation between vectors.

Many forms of dynamic coefficients may be used, indeed in the general case any form of digital filter may be applied to the feature vector. The most popular forms of dynamic coefficients are delta parameters, $\mathbf{\Delta O}^c(\tau)$, calculated as either simple differences or as a linear regression over a number of frames. Hence,

$$\mathbf{\Delta O}^c(\tau) = \frac{\sum_{t=d}^{D} t(\mathbf{O}^c(\tau+t) - \mathbf{O}^c(\tau-t))}{2\sum_{t=d}^{D} t^2} \tag{2.47}$$

where for regression parameters $d = 1$ and simple differences $d = D$, D is the width over which the dynamic coefficients are to be calculated.

The use of differential coefficients may be extended one more stage to acceleration, or delta-delta parameters, $\mathbf{\Delta}^2\mathbf{O}^c(\tau)$. Again, these may be calculated as simple differences or in a linear regression fashion.

Note that the use of dynamic coefficients, either calculated as a linear regression or simple difference, violates one of the assumptions behind the use of HMMs for speech recognition, that the speech waveform may be split into stationary segments with instantaneous transitions between them. Here, an observation in one state may depend on the previous and following states.

Chapter 3

Techniques for Noise Robustness

A substantial amount of work has been performed in the field of robust speech recognition. The various approaches adopted can be split into three general categories. Firstly, an inherently robust parameterisation of the speech may be used, in which case no additional processing is required. Alternatively, the clean speech may be estimated from the corrupted-speech signal, then the standard speech recogniser may be used. Finally, the problem of robust speech recognition may be transferred from the front-end processing stage to that of the pattern matching stage. Each of these approaches is further described below. This chapter describes a few techniques in detail, particularly those related to the schemes developed in this work. For an extensive survey of speech recognition in noise techniques see [30].

3.1 Inherently Robust Speech Parameters

The aim of this approach is to choose speech parameters so that the distortions due to additive and/or convolutional noise are minimal. Hence the recognition process can simply be based on $\mathcal{L}(\mathbf{O}^c(\tau)|\mathcal{M}_s)$[1], where $\mathcal{M}_s$ is the clean speech model, with no need to estimate the background noise. Of these techniques, some are primarily aimed at robustness to the effects of convolutional noise, channel distortion, others for additive noise, and a few attempt to handle the joint condition of additive and convolutional noise.

3.1.1 Cepstral Mean Normalisation

A popular technique for noise robustness in large vocabulary speech recognition is the use of Cepstral Mean Normalisation (CMN) [95]. Here, the speech is parameterised using Cepstral parameters with an additional stage, in which a mean value is subtracted from each Cepstral element. This mean is calculated either as a running smoothed value or on a per-sentence level. On a per-sentence level the observation is given by

$$\hat{\mathbf{O}}^c(\tau) = \mathbf{O}^c(\tau) - \frac{1}{T}\sum_{t=1}^{T}\mathbf{O}^c(t) \tag{3.1}$$

If purely time-invariant convolutional noise is present this set of speech parameters is unaffected by changes in the noise. When additive noise is present, subtracting the mean

[1] The likelihood shown here is based on static Cepstral parameters. It is not necessary to use Cepstral parameters, nor limit the feature vector to static elements.

is found to aid the robustness of the system. However, the system behaviour is hard to predict when both forms of noise are present.

3.1.2 Short-term Modified Coherence

Short-term Modified Coherence (SMC) [56] is an all-pole modelling of the autocorrelation sequence, followed by a spectral shaper. Modelling the autocorrelation function, which is less affected by noise than the original signal [36], has been found to be more robust to additive noise. In addition, it may be assumed that only the first few autocorrelation parameters are affected by the noise, a very good assumption for near white noise sources. Removing these from the parameter estimation process further improves the robustness. More recently, a similar approach, One-Sided Autocorrelation Linear Predictive Coding (OSALPC) [37], has been found to give even better performance than SMC with a variety of speech parameters in the car environment.

This form of speech parameters has been tested on alpha-digits, the alphabet and digits. All showed improvements using SMC Cepstra over LPCC. However, as with many techniques, it has not been tested on larger vocabulary systems, nor in comparison with other compensation techniques.

3.1.3 Linear Discriminant Analysis

It has been reported that using features selected with Integrated Mel-scale with Linear Discriminate Analysis (IMELDA) yields a more noise robust feature set than standard MFCCs [39]. However, the robustness of these features is still limited, particularly as the Signal-to-Noise Ratio (SNR) drops. To improve the robustness, Linear Discriminant Analysis (LDA), or IMELDA, may be applied using data collected in a particular noise environment [85]. A set of features is then obtained that optimises the standard LDA criterion for that SNR and noise source. This has been found to yield improved performance.

This approach has drawbacks when the transform is optimised for a particular noise condition. The transformation is sensitive to mismatches in both SNR and noise source. Thus, for a new source or noise level a new transformation is required. This must be either calculated on data collected in the new acoustic environment or artificially generated data, assuming that the additive and convolutional noise sources are known.

3.1.4 Generalised Cepstral Analysis

In most state-of-the-art speech recognition systems Cepstral parameters are used in the speech feature vector. These have been found to perform well in clean conditions, however, their performance in noise-corrupted environments has been poor. People have investigated modifications of the Cepstral parameters to achieve improved noise robustness by using functions other than *log()* for homomorphic deconvolution [80]. The root Cepstral domain [50] uses a power-based operator instead of the standard *log()* operator, this is called root homomorphic deconvolution. Thus

$$O_i^{l,\gamma}(\tau) = O_i(\tau)^\gamma \tag{3.2}$$

A further generalisation, that is a superset of the standard Cepstral domain, is the generalised logarithmic function [47, 87] where

$$O_i^{l,\gamma}(\tau) = \begin{cases} (O_i(\tau)^\gamma - 1)/\gamma, & 0 < |\gamma| \leq 1 \\ \log(O_i(\tau)), & \gamma = 0 \end{cases} \tag{3.3}$$

The mel-generalised Cepstral parameters $O^{c,\gamma}(\tau)$ are the inverse Fourier transform of this generalised Log-Spectral parameter, calculated according to some warped frequency scale.

Root homomorphic deconvolution has been investigated in the context of noise robustness [4, 54]. Using a power of $\frac{2}{3}$ was found to achieve significant improvements in noise robustness over standard Cepstral parameters on a small vocabulary task. However, these techniques have not been tested in medium or large vocabulary tasks. On these harder tasks, where the performance is more sensitive to mismatches in the training and test conditions, the same level of robustness may not be achieved.

3.1.5 RASTA-PLP

Perceptual Linear Predictive (PLP) analysis [34] is similar to linear prediction techniques, but uses transformations that are believed to be more consistent with human hearing. In PLP critical band spectral resolution, a Bark-hertz transformation, is used. In addition, equal loudness pre-emphasis and an intensity-loudness power law are applied. These operations are usually applied in the power spectral domain, so prior to calculating the all-pole model an inverse fourier transform is taken.

RelAtive SpecTrAl (RASTA) [48] processing makes use of the fact that convolutional noise effects may be assumed to be approximately stationary. Hence, by either high-pass or band-pass filtering the parameters these effects will be removed. The RASTA process involves applying a compressing static non-linear transformation, commonly a *log()*. This is then temporally filtered, and finally mapped back through an expanding static non-linear transformation. When applied to PLP this operation is performed prior to the application of the intensity-loudness law. In a similar way to CMN described above, when a *log()* operation is used for the compression, RASTA is insensitive to convolutional noise.

RASTA has been extended to accommodate the situations when additive noise and convolutional noise is present [48, 64], J-RASTA. Here, the *log()* compression is replaced by

$$y_i = \log\left(1 + Jx_i\right) \tag{3.4}$$

The inverse function is not calculated exactly, as this could result in negative values for the power domain, in much the same way as spectral subtraction, thus

$$x_i = \frac{\exp(y_i)}{J} \tag{3.5}$$

This is guaranteed to be positive, but is less accurate for small values of J. It has been noted [35] that there is a distinct optimal value for J for each particular noise level. Various methods for estimating the value of J and compensating for differences between J_{ref} and J_{test}, the optimal training and test values of J respectively, have been investigated [48].

RASTA is not necessarily linked with PLP coding, however it is the form that it is normally implemented.

Results for RASTA-PLP in both convolutional noise [74] and additive and convolutional noise [35, 48, 64] show improvement over standard PLP, LPC or MFCC parameters. As noted the performance of J-RASTA is dependent on the choice of J, which may result in mismatches between training and test speech parameters. Though techniques for handling this problem have been proposed, it is not clear how robust they are to widely varying noise levels and conditions.

3.2 Clean Speech Estimation

Instead of relying on some inherently noise-robust set of speech parameters, it is possible to estimate the clean speech from the noisy observations. Thus the likelihood will be of the form $\mathcal{L}(\hat{\mathbf{S}}^c(\tau)|\mathcal{M}_s)^2$, where

$$\hat{\mathbf{S}}^c(\tau) = \mathcal{H}(\mathbf{O}^c(\tau), \mathcal{M}_s, \mathcal{M}_n) \tag{3.6}$$

There has been much research into the exact form the mapping function, $\mathcal{H}()$, should take and what parameters it should depend on, here the speech and noise models, $\mathcal{M}_s$ and $\mathcal{M}_n$, are used. For some of the approaches described it is necessary to make some estimate of the background noise, either explicitly on periods of silence or estimated along with the clean speech.

3.2.1 Spectral Subtraction

The simplest technique for estimating the clean speech is to use spectral subtraction. There have been many forms of spectral subtraction [51], however most fit into the general form

$$\hat{S}_i(\tau) = \begin{cases} \left[\frac{|O_i(\tau)|^\gamma - \alpha|\hat{N}_i(\tau)|^\gamma}{|O_i(\tau)|^\gamma}\right]^\beta O_i(\tau), & (|O_i(\tau)|^\gamma - \alpha|\hat{N}_i(\tau)|^\gamma) > k^{\frac{1}{\beta}}|O_i(\tau)|^\gamma \\ kO_i(\tau), & \text{otherwise} \end{cases} \tag{3.7}$$

where k is the maximum allowable attenuation of the corrupted signal [88]. There are four variables for this expression, α, β, k and γ. These correspond to various forms of spectral subtraction. In recent years much work has concentrated on the selection of optimal values for α. The use of neural nets [97] and functions of the instantaneous SNR [55] have both been studied.

The basic assumption underlying spectral subtraction is that the noise is stationary with zero variance and there is no variability in the estimation of the speech and noise spectra. Since neither is valid it is necessary to incorporate a maximum attenuation constant, k, to prevent the speech estimate becoming negative. For low energy speech this may result in poor estimates. Furthermore, if dynamic coefficients are to be incorporated into the feature vector, these must to be calculated as the differences of clean static estimates. Thus, any errors in the estimation scheme will be accentuated when the dynamic coefficients are estimated.

3.2.2 Probabilistic Optimal Filtering

The mapping function for Probabilistic Optimal Filtering (POF) [67] is given by a piecewise linear transformation. This transformation is learnt by quantising the feature space into a set of distinct regions and minimising the squared error between clean and noise-corrupted observations. These noise-corrupted observations may be obtained using stereo recordings, or by artificially adding noise to the clean data. Thus for a set of N pairs of data points $\{\mathbf{S}^c(\tau), \mathbf{O}^c(\tau)\}$, the aim is to minimise for each of the I regions, g_i,

$$E_i = \sum_{\tau=p}^{N-1-p} p(g_i|\mathbf{Z}_\tau)\mathbf{e}_{i\tau}^T\mathbf{e}_{i\tau} \tag{3.8}$$

[2] The estimation, as will be seen, need not necessarily be performed in the Cepstral domain, nor Cepstral parameters used.

where

$$\mathbf{e}_{i\tau} = \mathbf{S}^c(\tau) - \left(\mathbf{b}_i + \sum_{k=-p}^{p} \mathbf{A}_{ik}\mathbf{O}^c(\tau+k)\right) \tag{3.9}$$

p is the maximum filter delay and $\mathbf{Z}_i$ is some conditioning noise vector, which in addition to the noisy observations may contain information such as the instantaneous SNR. If a stereo database is used then $p(g_i|\mathbf{Z}_\tau)$ may be obtained using the clean data. The training of $\mathbf{A}_{ik}$ and $\mathbf{b}_i$ is a standard linear-algebra problem. At run time, the estimate of the clean speech is given by

$$\hat{\mathbf{S}}^c(\tau) = \sum_{k=-p}^{p}\left[\left(\sum_{i=0}^{I-1} p(g_i|\mathbf{Z}_\tau)\mathbf{A}_{ik}\right)\mathbf{O}^c(\tau+k) + \sum_{i=0}^{I-1} p(g_i|\mathbf{Z}_\tau)\mathbf{b}_i\right] \tag{3.10}$$

Dynamic coefficients may either be calculated as if the estimates of the clean speech were correct, or separate transforms used.

The main disadvantage of this work is that the environment must be effectively learnt. If a new environment is encountered a new set of mapping functions must be obtained. This requires either stereo data to be present, or the acoustic environment to be known and the clean speech data adapted accordingly.

3.2.3 Code-book Dependent Cepstral Normalisation

Like the POF work described above, the variants of Code-book Dependent Cepstral Normalisation (CDCN) [2, 3, 53] attempt to learn a mapping from the noise-corrupted speech to clean speech. In its standard form CDCN can learn its new environment with no additional information, such as stereo data, being available. The estimate of the speech is given by

$$\hat{\mathbf{S}}^c(\tau) = \mathbf{O}^c(\tau) - \hat{\mathbf{H}} - \sum_{i=0}^{I-1} p(g_i|\mathbf{O}^c(\tau))\hat{\mathbf{r}}_i \tag{3.11}$$

where

$$\hat{\mathbf{r}}_i = \mathbf{C}(\log(1 + \exp(\mathbf{C}^{-1}(\hat{\mathbf{N}}^c - \hat{\mathbf{H}} - \mathbf{c}_i)))) \tag{3.12}$$

and $\mathbf{c}_i$ is the "mean" of the vector quantized region g_i, I is the number of such regions, $\hat{\mathbf{H}}$ is an estimate of the convolutional noise and $\hat{\mathbf{N}}^c$ is an estimate of the additive noise. These estimates are obtained in a ML fashion using an iterative approach. Unfortunately, they are often too computationally expensive to obtain, so simpler approximations to this scheme are often used. The simplest approximation is to make the correction factor a function of the instantaneous SNR, SNR_i, thus

$$\hat{\mathbf{S}}^c(\tau) = \mathbf{O}^c(\tau) - \mathbf{w}(\mathrm{SNR}_i) \tag{3.13}$$

The values of $\mathbf{w}(\mathrm{SNR}_i)$ are learnt from stereo data. This approach has been extended to make the compensation vector a function of a code-book entry, thus the feature space will again be split into I regions. The correction vector, or mapping, is again learnt from stereo data.

As the computational load of estimating the noise and channel distortion on-line is high, the various approximations are normally used. These must be learnt for each new noise condition, with the same problems as the POF technique. Additionally, this technique was primarily developed for convolutional noise with a low additive noise component. Its performance in high additive noise conditions has not been evaluated.

3.2.4 State-Based Speech Enhancement

An improved estimate of the clean speech should be possible if statistics, or estimates thereof, are known about the underlying clean speech and the interfering additive noise. Assuming that a model of the interfering additive noise may be obtained from the ambient noise conditions, it is only necessary to obtain statistics about the underlying clean speech. One approach adopted to obtain these statistics is to perform an initial recognition pass using a set of HMMs. This yields a frame/state alignment. Using this alignment, the statistics for the clean speech are available. However, the task of obtaining good estimates for the frame/state alignment in the presence of additive interfering noise is itself a speech recognition in noise task. Two approaches that have been investigated are the use of parallel model combination to compensate a set of clean models to be representative of corrupted models in the Cepstral domain [83], and a version when the models are generated in the Linear-Spectral domain [23].

Having obtained the statistics for the speech, an appropriate estimator may be employed. In both investigations the use of an inhomogeneous affine estimator was used, where the estimate in the Linear-Spectral domain is given by

$$\hat{S}_i(\tau) = \mu_i + \frac{\Sigma_{ii}}{\Sigma_{ii} + \tilde{\Sigma}_{ii}} \left(O_i(\tau) - \mu_i - \tilde{\mu}_i \right) \tag{3.14}$$

where μ and $\boldsymbol{\Sigma}$ are the mean and variance of the clean speech state distribution, and $\tilde{\mu}$ and $\tilde{\boldsymbol{\Sigma}}$ are those associated with the noise.

These techniques have only been applied, within the context of speech recognition tasks to small-vocabulary, digit, systems.

A different approach to this state based enhancement task [19, 20] uses HMMs with Auto-Regressive (AR) output probability density functions [41] to model the clean speech. As AR output probability density functions are used to model the time-domain waveforms, the effect of the noise on the speech may be assumed to be additive. By also modelling the interfering noise source by a second AR model, both Minimum Mean Square Error (MMSE) and Maximum A-Posteriori (MAP) estimates for the clean speech may be obtained, given the frame/state component alignment. This alignment may be obtained in an iterative fashion, by using each successive clean speech signal estimate to obtain a new frame/state component alignment, or by directly using a model of the corrupted speech. When used as a MMSE clean speech estimator this technique becomes a state-specific Wiener filter. However, presently most state-of-the-art speech recognition systems use HMMs built on Cepstral parameters. Thus the simple additive assumption used for AR HMMs is not applicable.

3.3 Model-Based Techniques

An alternative to the schemes previously described is to modify the acoustic models, the pattern matching stage, instead of the parameterisation stage. This has the advantage that no decisions or hypotheses about the speech are necessary and the observed data is unaltered. The likelihood of the observation is of the form $\mathcal{L}(\mathbf{O}^c(\tau)|\mathcal{M}_s, \mathcal{M}_n)$. Two different approaches may be adopted to model the corrupted-speech distribution. Firstly, standard Gaussian probability distributions may be used to model the noise corrupted observations. Alternatively a non-Gaussian probability distribution may be used. Both styles of modelling have previously been examined.

3.3.1 Linear Regression Adaptation

The task of adapting a set of models to a new acoustic environment may be related to speaker adaptation. Here instead of adapting the means, and possibly variances, of a set of speaker independent models to a particular speaker, the models are adapted to a particular speaker, or group of speakers, in a new acoustic environment. One currently popular method for speaker adaptation is Maximum Likelihood Linear Regression (MLLR) [49]. If only the means are considered then

$$\hat{\mu}_s^c = \mathbf{A}_s \xi_s^c \tag{3.15}$$

where $\hat{\mu}_s^c$ is the estimate of the noise-corrupted mean, $\mathbf{A}_s$ is an $n \times (n+1)$ transformation matrix associated with class s, and ξ_s^c is the extended mean vector

$$\xi_s^c = \begin{bmatrix} w & (\mu^c)^T \end{bmatrix}^T \tag{3.16}$$

where w is an offset term. For no offset w is set to zero, otherwise it is set to one. $\mathbf{A}_s$ is trained in a ML fashion using speech in the new acoustic environment. This form of linear regression model adaptation has also been applied to adapting the variances of the models [18].

One of the major drawbacks of these techniques when applied to noise-corrupted environments is that the transformation matrices are estimated given the frame/state alignment. If running in an unsupervised mode, that is the transcription of the adaptation speech is unknown, and to a lesser extent in supervised mode, the initial estimate of the frame/state alignment may be poor. This will yield a poor transformation matrix and, hence, poor performance. To improve the frame/state alignment estimate this adaptation technique may be applied to alignments obtained from noise compensated models, for example using parallel model combination or hypothesised Wiener filtering to initially compensate the models. A second problem arises if the technique is used to transform clean model sets. Due to the non-linearities of the noise compensation process, it is necessary to have enough regression classes to model this transformation well. Unfortunately, there is not always sufficient adaptation material to allow this.

3.3.2 Model-Based Stochastic Matching

In stochastic pattern matching[3] [82] the aim is to modify the clean model parameters to reflect the model parameters in a new acoustic environment, given a test utterance in the new environment, $\mathbf{O}_T^c$. The matching is performed in a ML fashion. The aim is thus to learn the mapping

$$\hat{\mathcal{M}}_s = \mathcal{G}_\eta(\mathcal{M}_s) \tag{3.17}$$

where the parameters, η, of the transformation, $\mathcal{G}_\eta()$, are estimated such that

$$(\hat{\eta}, \hat{\mathbf{W}}) = \arg \max_{(\eta, \mathbf{W})} [\mathcal{L}(\mathbf{O}_T^c, \mathbf{W} | \eta, \mathcal{M}_s)] \tag{3.18}$$

and $\mathbf{W}$ is a possible word string. For the channel distortion case the mismatch is assumed to be a random additive bias, $\mathbf{H}^c$, so the parameters of the transformation are

$$\eta = \{\mu_h^c, \mathbf{\Sigma}_h^c\} \tag{3.19}$$

[3] Here, the technique is being considered as a model-based compensation technique. The same form has also been applied as an enhancement-based technique.

where μ_h^c is the mean and $\mathbf{\Sigma}_h^c$ is the diagonal covariance matrix of the bias. The parameters are estimated in an iterative fashion as the actual word sequence $\mathbf{W}$ is unknown. The technique has been applied to telephone speech where the convolutional noise mismatch will dominate the degradation in performance.

A similar approach has been adopted for the case of significant additive noise [63]. Here systems using stereo databases, RATZ, and non-stereo databases, blind RATZ, have been investigated.

This form of model adaptation is closely linked with the linear regression work in the previous section. The main difference between the two techniques is that for the linear regression based techniques the new corrupted mean is a function of both the current mean vector and a bias. If variance compensation is used, the same transformation matrix as used for the means is applied. Here on the other hand, the mismatch is modelled as a bias with an additional variance estimation. This reduces the number of parameters to estimate, but is less powerful. This technique will have the same problems as the MLLR scheme.

3.3.3 Noise Masking

The basic assumption behind noise masking is that either the noise or the speech will dominate in the Linear-Spectral or, more normally, the Log-Spectral domain. The form of the observation likelihood is determined by the corrupted-speech signal, the mean of the noise signal, and the particular clean speech distribution. Three possible noise masking schemes are the Klatt algorithm [46], the Bridle et al. algorithm [12] and the Holmes and Sedgewick algorithm [38]. These have all been examined within the same HMM framework [89] from which the following tables are taken. For this work a fully automatic median noise tracking algorithm was used to obtain $\hat{N}_i^l(\tau)$.

Condition	Relationship	
a_1	$O_i^l(\tau) < \mu_i^l$	$\hat{N}_i^l(\tau) < O_i^l(\tau) < \mu_i^l$
a_2		$O_i^l(\tau) < \hat{N}_i^l(\tau) < \mu_i^l$
a_3		$O_i^l(\tau) < \mu_i^l < \hat{N}_i^l(\tau)$
b_1	$O_i^l(\tau) > \mu_i^l$	$\hat{N}_i^l(\tau) < \mu_i^l < O_i^l(\tau)$
b_2		$\mu_i^l < \hat{N}_i^l(\tau) < O_i^l(\tau)$
b_3		$\mu_i^l < O_i^l(\tau) < \hat{N}_i^l(\tau)$

Table 3.1: Different noise condition regions for noise masking function

Depending on the relationship between the observed noisy signal, $O_i^l(\tau)$, the mean of the speech model, μ_i^l and the estimate of the noise at that time instance, $\hat{N}_i^l$, for each channel, one of the conditions, $\{a_1, a_2, a_3, b_1, b_2, b_3\}$, is selected.

Having selected one of the conditions, the exact form of the likelihood is determined by the particular masking scheme to be used. The various options for $\mathcal{L}(\mathbf{O}^l(\tau)|\mathcal{M}_s, \mathcal{M}_n)$ using the masking schemes mentioned are given in the table 3.2. In the table, d_i^l is set to some empirically chosen non-zero value and $\mathcal{C}(\hat{N}_i^l(\tau), \mu_i^l, \Sigma_{ii}^l)$ is the cumulative probability density function. The form of the likelihood calculation is dependent on the assumptions made. For the Klatt algorithm, if either the model or the observation are below the noise estimate they are masked by the noise estimate. This technique does not make best use of

Cond.	Klatt	Bridle et al.	Holmes
a_1	$\mathcal{N}(O_i^l(\tau), \mu_i^l, \Sigma_{ii}^l)$	$\mathcal{N}(O_i^l(\tau), \mu_i^l, \Sigma_{ii}^l)$	$\mathcal{N}(O_i^l(\tau), \mu_i^l, \Sigma_{ii}^l)$
a_2	$\mathcal{N}(\hat{N}_i^l(\tau), \mu_i^l, \Sigma_{ii}^l)$	$\min(\mathcal{N}(d_i^l, 0, \Sigma_{ii}^l)), \mathcal{N}(O_i^l(\tau), \mu_i^l, \Sigma_{ii}^l)$	$\mathcal{C}(\hat{N}_i^l(\tau), \mu_i^l, \Sigma_{ii}^l)$
a_3	$\mathcal{N}(\hat{N}_i^l(\tau), \hat{N}_i^l, \Sigma_{ii}^l)$	$\mathcal{N}(d_i^l, 0, \Sigma_{ii}^l)$	$\mathcal{C}(\hat{N}_i^l(\tau), \mu_i^l, \Sigma_{ii}^l)$
b_1	$\mathcal{N}(O_i^l(\tau), \mu_i^l, \Sigma_{ii}^l)$	$\mathcal{N}(O_i^l(\tau), \mu_i^l, \Sigma_{ii}^l)$	$\mathcal{N}(O_i^l(\tau), \mu_i^l, \Sigma_{ii}^l)$
b_2	$\mathcal{N}(O_i^l(\tau), \hat{N}_i^l(\tau), \Sigma_{ii}^l)$	$\mathcal{N}(O_i^l(\tau), \mu_i^l, \Sigma_{ii}^l)$	$\mathcal{N}(O_i^l(\tau), \mu_i^l, \Sigma_{ii}^l)$
b_3	$\mathcal{N}(\hat{N}_i^l(\tau), \hat{N}_i^l(\tau), \Sigma_{ii}^l)$	$\mathcal{N}(d_i^l, 0, \Sigma_{ii}^l)$	$\mathcal{C}(\hat{N}_i^l(\tau), \mu_i^l, \Sigma_{ii}^l)$

Table 3.2: Probability distributions assumed for various noise conditions and masking functions

the information available in the speech, however it was found to give good performance, particularly for low energy speech regions. Alternatively, in the Bridle et al algorithm, if the observation and the model are below the noise level then nothing can be deduced about the distance between the two, so some minimum distance, d_i^l, is used. Finally in the Holmes and Sedgewick algorithm where the observation falls beneath the noise level, nothing may be deduced about the speech other than it is below the noise, so a cumulative probability is used.

These techniques are applied in the Log-Spectral domain. A masking scheme in the MFCC domain [59] has been investigated with improved recognition performance.

The masking techniques utilise a very simple model of the interfering noise source, the mean of the noise. It would be preferable to use a more powerful model for the noise source.

3.3.4 Speech and Noise Decomposition

If a HMM is used to model the background interfering noise, instead of using a simple mean or related value, it is unnecessary to assume that the noise has zero variance, or the same variance as the speech. The probability of a noisy observation is then given by

$$\mathcal{L}(O_i^l(\tau)|q_j(\tau), q_v^n(\tau), \mathcal{M}_s, \mathcal{M}_n) = \int \mathcal{L}(S_i^l(\tau), N_i^l(\tau)|q_j(\tau), q_v^n(\tau), \mathcal{M}_s, \mathcal{M}_n) \tag{3.20}$$

where s_j is the current speech state, $q_v^n(\tau)$ indicates being in state s_v^n of the noise model at time τ, and the integral is over all couples such that $O_i^l(\tau) = S_i^l(\tau) \otimes N_i^l(\tau)$. If the speech and noise are assumed to be additive and independent, then the relationship is

$$O_i^l(\tau) = \log(\exp(S_i^l(\tau)) + \exp(N_i^l(\tau))) \tag{3.21}$$

Unfortunately there is no simple expression for the integral over all couples when the noisy observation has this form. Hence a simplification is adopted, whereby it is assumed that

$$O_i^l(\tau) \approx \max(S_i^l(\tau), N_i^l(\tau)) \tag{3.22}$$

Under this additional assumption it can be shown that

$$\begin{aligned} &\mathcal{L}(O_i^l(\tau)|q_j(\tau), q_v^n(\tau), \mathcal{M}_s, \mathcal{M}_n) = \\ &\quad \mathcal{N}(O_i^l(\tau); \mu_i^l, \Sigma_{ii}^l)\mathcal{C}(O_i^l(\tau); \tilde{\mu}_i^l, \tilde{\Sigma}_{ii}^l) + \mathcal{C}(O_i^l(\tau); \mu_i^l, \Sigma_{ii}^l)\mathcal{N}(O_i^l(\tau); \tilde{\mu}_i^l, \tilde{\Sigma}_{ii}^l) \end{aligned} \tag{3.23}$$

This is referred to as Speech and Noise Decomposition [90] (SND) or Adaptive Noise Prototypes [65]. The form of the output probability function has now been altered from the standard Gaussian distributions to a weighted sum of two Gaussian distributions.

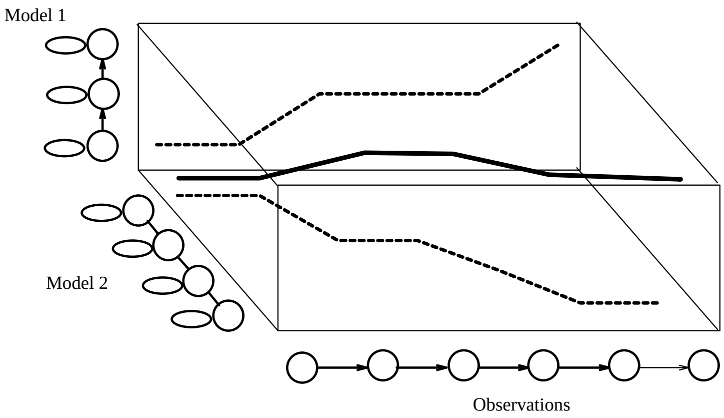


Figure 3.1: 3-dimensional Viterbi decoder

As it is not possible to know a-priori which state of the noise model and which state of the speech model is most likely at recognition time, it is necessary to introduce the concept of a 3-dimensional Viterbi decoder [90]. Here, a new variable $\phi_{j,v}(t)$ is introduced, which is defined as

$$\phi_{j,v}(t) = \max_{\theta \in \mathbf{\Theta}_{t-1}} \left[\mathcal{L}(\mathbf{O}^l_t, q_j(t), q^n_v(t), \theta | \mathcal{M}) \right] \tag{3.24}$$

where $\mathbf{\Theta}_{t-1}$ is the set of all partial paths of length $t-1$, θ is the path through both the speech and the noise model. The following recursion is used

$$\phi_{j,v}(t) = \max_{i,u} \phi_{i,u}(t-1) a_{ij} \tilde{a}_{uv} b_{jv}(\mathbf{O}^l(t)) \tag{3.25}$$

where $\phi_{j,v}(t)$ is the maximum joint probability of being in state s_j of the speech model and state s^n_v of the noise model at time t and observing the sequence $\mathbf{O}^l(1)$ to $\mathbf{O}^l(t)$. This allows a multiple state noise model to be traversed.

The major drawback of this technique is that again, owing to the max approximation, the data must be modelled in the Log-Spectral domain, which is less compact than the Cepstral domain. Moreover, each of the elements of the feature vector is assumed to be independent in the Log-Spectral domain.

3.3.5 Hypothesised Wiener Filtering

Most state-of-the-art speech recognition systems use Cepstral parameters, as these are more compact and the diagonal covariance assumption more appropriate. SND and the majority of the masking techniques are based on Log-Spectral parameters. It would be preferable to use a compensation scheme that is applicable in the Cepstral domain. One such model-based scheme is referred to as hypothesised Wiener filtering and has been used with both Dynamic-Time-Warping (DTW) based [10] and HMM-based [9] recognisers.

The aim of this technique is to obtain an "optimal" filter to convert the corrupted speech into clean speech. Here the optimal filter is calculated in the Linear-Spectral domain and is optimal in a least squares sense. For the case of independent additive noise the filter is

$$w_i = \frac{\mu_i}{\mu_i + \tilde{\mu}_i} \tag{3.26}$$

Transforming this filter into the Log-Spectral and then Cepstral domain yields,

$$\hat{\mathbf{S}}^c(\tau) = \mathbf{C}(\mathbf{O}^l(\tau) + \log(\mathbf{w})) = \mathbf{O}^c(\tau) + \mathbf{C}\log(\mathbf{w}) \tag{3.27}$$

The choice is either to offset the data as it enters a particular state, or alternatively to offset the means of the model distributions. It is simpler to offset the means of the clean speech distributions, so

$$\hat{\mu}^c = \mu^c - \mathbf{w}^c \tag{3.28}$$

where

$$\mathbf{w}^c = \mathbf{C}\left[\log(\mu) - \log(\mu + \tilde{\mu})\right] \tag{3.29}$$

For both the DTW and the HMM work the Linear-Spectral domain estimates, μ_i and $\tilde{\mu}_i$ are explicitly estimated from the training data. This technique has the advantage that it is applicable to Cepstral parameters. However, it makes an implicit assumption that for a particular state or component pairing the corrupted-speech distribution is approximately Gaussian. Furthermore, it is unable to compensate the variances of the system.

It would be preferable to be able to update all the parameters, not just the means of the static parameters of the model set in the Cepstral domain. This is the aim of parallel model combination, the scheme developed in this work.

3.4 Combination of Approaches

Various combinations of approaches for robust speech recognition have been examined with some success. IMELDA has been combined with Non-linear Spectral Subtraction (NSS) [86]. NSS has also been combined with root homomorphic parameterisation [54] giving improved noise robustness. The POF technique has been combined with a version of linear adaptation [18] to correct for the errors in the linear partitioning of the feature space [68]. This was the system used by SRI on the ARPA 1994 CSRNAB Spoke 10 evaluation. By combining the two methods the system was found to be more robust to noise. The parallel model combination technique described in this work has been combined with spectral subtraction [70, 71]. Here the model adaptation is used to compensate for the approximations used in the spectral subtraction process. On the small task examined improvements in recognition rates were obtained.

The general approach of enhancing the speech then using some adaptation technique to compensate for errors in the enhancement process has various attractions associated with it. The corrupted-speech distribution in the Log-Spectral domain is known to be non-Gaussian, even if Gaussian speech and noise sources are used. Thus, to accurately model this corrupted distribution may required additional components in the corrupted-speech model, or the use of non-Gaussian distributions. By firstly enhancing the speech, the distribution will now, hopefully, be approximately Gaussian. Unfortunately, the use of an enhancement technique, particularly when low energy speech is completely masked by noise, is unlikely to give good estimates of the clean speech. Another advantage in using an enhancement based scheme is that the best performance is now the clean performance figure, in contrast to the model-based techniques where the best performance will be a matched system. Though, again, there is the problem of “reconstructing” the low energy clean speech in periods where it is masked by noise.

Chapter 4

The Effects of Additive Noise

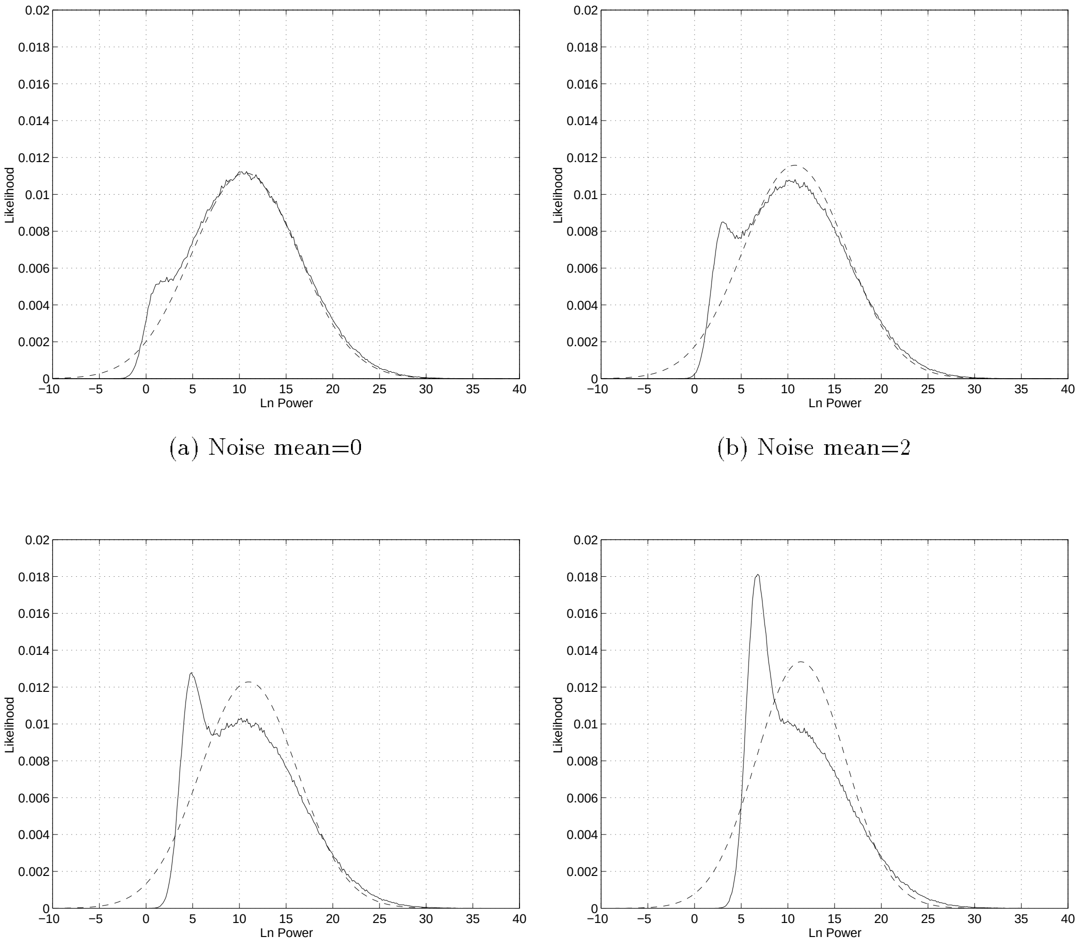
If a model-based approach is to be used, it is important to know what form the corrupted-speech distribution will take. To this end a simplified system was investigated, where only one dimension in the Log-Spectral domain was considered. Both the speech and noise sources were assumed to be well modelled by Gaussian distributions in the Log-Spectral domain and the noise to be additive in the Linear-Spectral domain. Thus, the corrupted-speech probability will be given by equation 3.20. Since there is no simple closed form for this expression, a Monte-Carlo simulation was used to generate an estimate of the probability density function.

Figure 4.1 shows the estimate of the corrupted-speech distribution and the ML Gaussian distribution estimate of the data[1]. The ML estimate of a Gaussian distribution for each case is shown, as it illustrates the mean and variance of the corrupted-speech distribution. For all the plots, the "speech" was generated from a Gaussian distribution with a mean of 10.5 and standard deviation of 6. The noise means were set to 0, 2, 4 and 6, all with a standard deviation of 1.

Even at relatively low noise levels the corrupted-speech distribution was non-Gaussian. As the noise level increases, the distribution became increasingly bimodal. Eventually, as the noise starts to dominate, the corrupted-speech distribution again became unimodal, however still very distinctly non-Gaussian. The plots also show that for model-based schemes it is not the SNR that is important, but the distance of the speech mean to the noise mean measured in terms of the standard deviations of the speech and noise distributions. Some general trends of the ML estimated Gaussian distribution associated with the corrupted speech were observed. Firstly as the SNR decreased, so the variance of the corrupted speech gradually moved towards that of the noise. Thus in the case illustrated, the variance of the corrupted speech became less than that of the clean speech. Of course in the limit the distribution of the corrupted speech will be the same as that of the interfering noise. In addition to the change in the variance, the mean of the distribution increased as the SNR decreased. These general trends in the Log-Spectral domain have been noted elsewhere [63].

Despite the fact that these trends are in the Log-Spectral domain, they indicate what may happen in the Cepstral domain. The non-Gaussian nature of these distributions in the Log-Spectral domain will result in non-Gaussian distributions in the Cepstral domain, as there is a linear transformation between the two domains. The bimodality of the

[1] These are not actual probability density functions, but were generated using histograms. The areas under the graph have been scaled to be equal.



(a) Noise mean=0

(b) Noise mean=2

(c) Noise mean=4

(d) Noise mean=6

Figure 4.1: Plots of "corrupted-speech" distribution (solid), and maximum likelihood Gaussian distribution (dashed)

corrupted-speech distribution in the Cepstral domain has previously been noted [73].

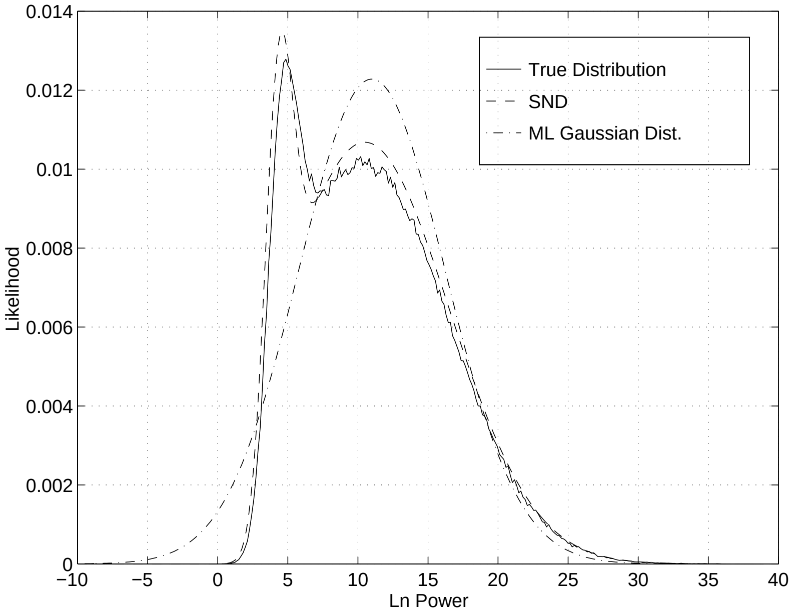
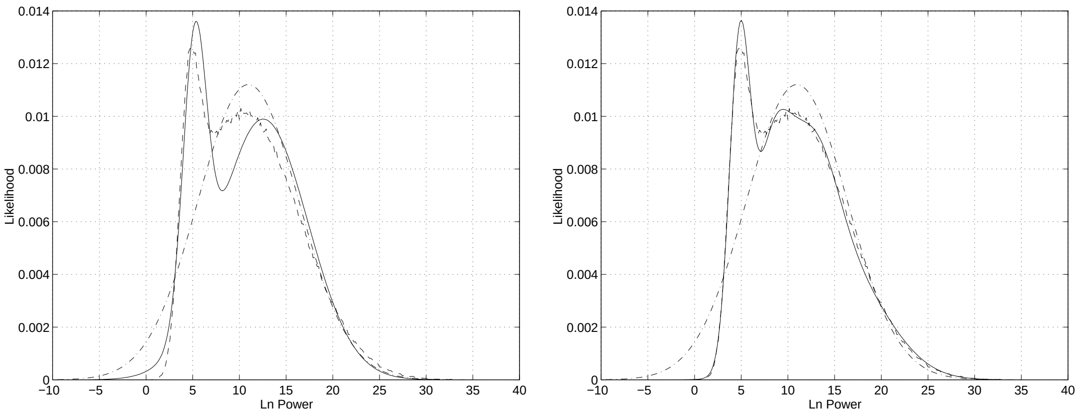


Figure 4.2: Plots of "corrupted-speech" distribution, maximum likelihood Gaussian distribution, and speech and noise decomposition distribution, noise mean=4

For a particular noise level, a noise mean of 4, the SND *max()* approximation was compared to the estimated corrupted-speech distribution. The plots are shown in figure 4.2. Using the simple *max()* approximation yielded a far closer approximation than using a Gaussian distribution. Indeed, it can model the bimodality observed in the corrupted-speech distribution. Unfortunately, as previously mentioned, this approximation is unsuitable for models in the Cepstral domain, whereas mapping to and from the Log-Spectral and Cepstral domains for Gaussian, and multi-variate Gaussian, distributions is simple.

In the same fashion as standard HMMs use multiple mixture components to model complex state distributions in clean conditions, they may be used to model this noise-corrupted distribution. Figure 4.3(a) shows a two component and figure 4.3(b) a four mixture component Gaussian distribution modelling the corrupted-speech distribution, with the estimated distribution and the single Gaussian plotted for reference. By increasing the number of Gaussian mixture components the non-Gaussian distribution was better modelled. However, there are problems associated with adding components. Firstly, there is the computational overhead associated with the increase in the number of components at recognition time[2]. Also, as there is no closed form for estimating multiple component distributions given a set of data, it is necessary to perform the optimisation in an iterative fashion. This increases the computational load for compensating the model set. On the other hand, an advantage of using multiple component Gaussian distributions to model the complexities of the corrupted-speech distribution is that it is simple to transform these distributions to, and from, the Cepstral domain. Again, the example given here is a single

[2] In practice more accurate modelling of the PDFs leads to more effective pruning and the actual computational load may not increase proportionally.



(a) Two mixture components

(b) Four mixture components

Figure 4.3: Plots of "corrupted-speech" distribution (dashed), maximum likelihood Gaussian distribution (dash-dot), and multiple mixture component distributions (solid), noise mean=4

dimension in the Log-Spectral domain. For the actual speech recognisers utilised in this work multi-dimensional, up to 39 dimensions for the medium to large vocabulary tasks, feature vectors, which are correlated in the Log-Spectral domain, are used.

If multiple components are used to model the states of the clean speech models, then it may not be necessary to increase the total number of components per state. By considering the corrupted-speech distribution on a per-state level, the same number of components as the clean system may be used to model the corrupted-speech data. This will give good results provided the number of modes in the corrupted distribution has not increased.

This chapter has only considered the effects of additive noise on the speech signal. However, when convolutional noise is present, under certain simplifying assumptions described in section 5.3.3, it may be treated as a simple shift in the Log-Spectral domain. Thus, the "corrupted-speech" distributions for this case will still have the same form as the one shown in figure 4.1.

Chapter 5

Parallel Model Combination

This chapter presents the theory behind Parallel Model Combination (PMC). The technique approximates the "best" model-based compensation scheme which is to train on noise-corrupted speech data. Techniques for compensating static and dynamic coefficients in both additive and convolutional noise are described.

Sections 5.1 to 5.7 describe the basic principles behind PMC, the mismatch functions used, and the iterative and non-iterative implementations of PMC. The following section, section 5.8, describes how a convolutional noise component can be estimated in the presence of additive noise. Complex-noise environments and how they may be handled within the PMC framework are described in section 5.9. Section 5.10 describes the various modelling approximations used in implementations of PMC. In order to assess the performance of PMC it is necessary to build matched systems. How such systems may be built, and the performance assessment criteria used, are described in sections 5.11 and 5.12.

5.1 Basic Parallel Model Combination Description

The first decision to be made in a model-based scheme is the form of the corrupted-speech model. For SND, previously described, the model chosen was a HMM with non-Gaussian output probability distributions. In PMC, the model used is a standard HMM with Gaussian output probability distributions. This model therefore requires no modification of the recognition software and allows the standard HMM estimation formulae to be applied. Thus, the basic assumption is that

$$\mathcal{L}(\mathbf{O}^c(\tau)|q_j(\tau), q_v^n(\tau), \mathcal{M}_s, \mathcal{M}_n) = \sum_{m=1}^{M} \hat{c}_m \mathcal{N}(\mathbf{O}^c(\tau); \hat{\mu}_m^c, \hat{\mathbf{\Sigma}}_m^c) \tag{5.1}$$

There are various approximations that may be used in estimating the set of weights, $\hat{c}_m$, means, $\hat{\mu}_m^c$, and variances, $\hat{\mathbf{\Sigma}}_m^c$. These are discussed later in this chapter. It is also necessary to decide, in exactly the same way as in standard HMM training, how many mixture components, M, are to be used to model a particular speech and noise state pairing. This need not necessarily be the same as the original number of components in the clean speech state [28], particularly given the complex nature of the corrupted-speech distribution, as described in Chapter 4.

Having decided on the form of the models, it is necessary to choose an approach to estimate the new model parameters. The best technique for additive noise would be to add samples of the background noise to the clean training data at the waveform level.

A new training database, matched to the test environment, could then be generated and a new set of models trained. However, it should be noted that in order to perform this training it is required that:

1. The whole training database is available on line.
2. Sufficient noise samples are available to add to the clean data.
3. Computational power is available for performing the noise addition, and training the model parameters, whenever the background noise changes.

Given these conditions, it will normally be impractical to perform this form of compensation. However, if the clean speech models are assumed to contain sufficient information about the statistics of the training data, they may be used in the compensation scheme in place of the data itself. Moreover, a model of the background noise can be generated using whatever noise samples are available to represent the background noise conditions. The problem then is to find a method of combining the two models to accurately estimate the corrupted-speech models.

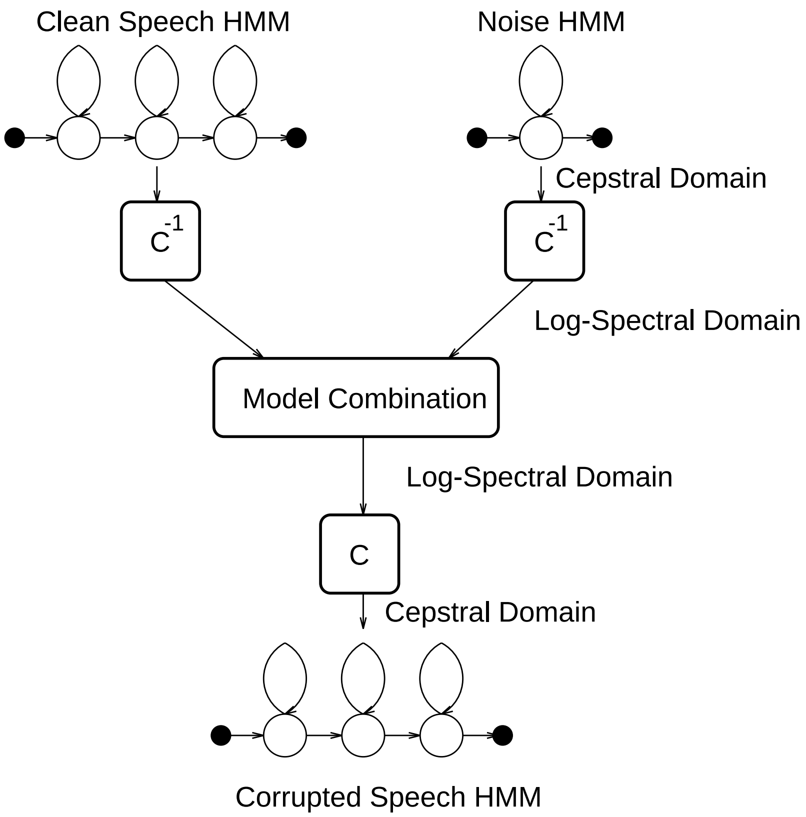


Figure 5.1: The basic parallel model combination process

The basic PMC process is illustrated in figure 5.1. The inputs to the scheme are clean speech models and a noise model. Since the combination of the speech and noise is most naturally expressed in the Linear-Spectral or Log-Spectral domains, it is simplest to model the effects of the additive noise on the speech parameters in one of these domains. The function that approximates this will be referred to as the *mismatch function*. If Cepstral parameters are used, the model parameters must be transformed into the appropriate combination domain (Log-Spectral in figure 5.1). The clean speech models and the noise model are then combined according to this mismatch function. After the models have been combined, the estimate of the corrupted-speech model is transformed into the Cepstral domain, if required.

5.2 Mapping from the Cepstral to the Log-Spectral Domain

As previously stated, if Cepstral parameters are used in the speech and noise models, it is necessary to map the models from the Cepstral domain to the Log-Spectral domain. This is achieved using the inverse DCT. For the clean speech static parameters, the mapping to the Log-Spectral domain is given by

$$\mu^l = \mathbf{C}^{-1}\mu^c \tag{5.2}$$

$$\mathbf{\Sigma}^l = \mathbf{C}^{-1}\mathbf{\Sigma}^c(\mathbf{C}^{-1})^T \tag{5.3}$$

Similarly, the noise parameters $\{\tilde{\mu}^c, \tilde{\mathbf{\Sigma}}^c\}$ may be mapped to $\{\tilde{\mu}^l, \tilde{\mathbf{\Sigma}}^l\}$. $\mathbf{C}$ is the matrix representing the DCT, the elements of which are given in equation 2.41. It is worth noting that even if the Cepstral domain covariance matrix is diagonal, the Log-Spectral domain covariance matrix will be full.

If delta parameters are appended to the feature vector, then these must also be mapped into the Log-Spectral domain. The mean feature vector in the Log-Spectral domain is

$$\mu^{\Delta l} = \left[\begin{array}{cc} (\mathbf{C}^{-1}\mu^c)^T & (\mathbf{C}^{-1}\mathbf{\Delta}\mu^c)^T \end{array}\right]^T \tag{5.4}$$

where $\mathbf{\Delta}\mu^c$ is the delta parameter mean in the Cepstral domain. The mapping of the covariance matrix is slightly more complex as there are covariance terms relating the static and delta coefficients. If a full Cepstral domain covariance matrix is used then

$$\mathbf{\Sigma}^{\Delta l} = \left[\begin{array}{cc} \mathbf{C}^{-1}\mathbf{\Sigma}^c(\mathbf{C}^{-1})^T & \mathbf{C}^{-1}\delta\mathbf{\Sigma}^c(\mathbf{C}^{-1})^T \\ \mathbf{C}^{-1}(\delta\mathbf{\Sigma}^c)^T(\mathbf{C}^{-1})^T & \mathbf{C}^{-1}\mathbf{\Delta\Sigma}^c(\mathbf{C}^{-1})^T \end{array}\right] \tag{5.5}$$

where $\mathbf{\Delta\Sigma}^c$ is the covariance matrix of the delta parameters and $\delta\mathbf{\Sigma}^c$ is the covariance matrix representing the correlation between the static and delta coefficients. A similar mapping converts the noise parameters in the Cepstral domain, $\{\tilde{\mu}^{\Delta c}, \tilde{\mathbf{\Sigma}}^{\Delta c}\}$, to the Log-Spectral domain, $\{\tilde{\mu}^{\Delta l}, \tilde{\mathbf{\Sigma}}^{\Delta l}\}$. If diagonal covariance matrices are used for the speech and noise models, the cross correlation terms between the static and delta parameters may be ignored. The extension of this mapping to incorporate delta-delta parameters is straightforward.

Mapping from the Log-Spectral domain back to the Cepstral domain is simply the reverse operation. Thus the inverse DCT, $\mathbf{C}^{-1}$, is replaced by the DCT, $\mathbf{C}$.

5.3 The Mismatch Function

The first stage in a model combination process is to produce some function describing how noise affects each of the speech parameters. In order to describe the effects of the noise on the clean speech parameters, a series of assumptions are required:

1. The speech and noise are independent.
2. The speech and noise are additive in the time domain. In addition, it is assumed that there is sufficient smoothing on the spectral estimate so that the speech and noise may be assumed to be additive at the power spectrum level.
3. A single Gaussian or a multiple Gaussian mixture component model contains sufficient information to represent the distribution of the observation vectors in the Cepstral or Log-Spectral domain.
4. The frame/state alignment, used to generate the speech models from the clean speech data, is not altered by the addition of noise.

Having made these assumptions it is possible to write expressions for the effects of the additive noise on each of the parameters of the feature vector. In order to use these expressions in a compensation process, it is necessary to have statistics for all the variables and the various correlations between the statistics must be adequately described.

5.3.1 Mismatch Function for the Static Parameters in Additive Noise

The static noise-corrupted speech "observations" in the Log-Spectral domain are given by the mismatch function

$$\begin{aligned} O_i^l(\tau) &= \mathcal{F}(S_i^l(\tau), N_i^l(\tau)) \\ &= \log(g\exp(S_i^l(\tau)) + \exp(N_i^l(\tau))) \end{aligned} \tag{5.6}$$

where g is a gain matching term introduced to account for level differences between the clean speech and the noisy speech.

5.3.2 Mismatch Function for Dynamic Parameters in Additive Noise

For medium to large vocabulary speech recognition it is necessary to incorporate dynamic coefficients in the parameter set to achieve good performance. These parameters are added in an attempt to model the correlation between successive frames. If the dynamic coefficients are calculated using simple differences over a given window width, w for the deltas and w_a for the delta-delta parameters, then the observation feature vector will be

$$\mathbf{O}^{\Delta^2 c}(\tau)^T = \left[\begin{array}{ccc} \mathbf{O}^c(\tau)^T & \mathbf{\Delta O}^c(\tau)^T & \mathbf{\Delta}^2\mathbf{O}^c(\tau)^T \end{array} \right]^T \tag{5.7}$$

where

$$\mathbf{\Delta O}^c(\tau) = (\mathbf{O}^c(\tau + w) - \mathbf{O}^c(\tau - w)) \tag{5.8}$$

and

$$\mathbf{\Delta}^2\mathbf{O}^c(\tau) = \mathbf{\Delta O}^c(\tau + w_a) - \mathbf{\Delta O}^c(\tau - w_a) \tag{5.9}$$

are the delta and delta-delta parameters respectively. If statistics about the correlation between successive frames are known, for example $\mathcal{E}\{\mathbf{S}^c(\tau+w)\mathbf{S}^c(\tau-w)^T\}$, then statistics, including correlations, exist for the above expressions and the corrupted delta and delta-delta parameters may be estimated using the standard static parameter mismatch function. However, in order to determine the correlation between successive frames it is necessary to use full, or partial full, covariance matrices for the clean speech, with an associated increase in model storage requirements. This is further discussed in section 5.4.

An alternative to the very direct modelling given above is to re-express the delta parameters in terms of parameters whose statistics are known and, if necessary, may be assumed to be independent [26]. In Appendix A it is shown that

$$\begin{aligned}
\Delta O_i^l(\tau) &= \mathcal{F}^{\Delta}(S_i^l(\tau-w), N_i^l(\tau-w), \Delta S_i^l(\tau), \Delta N_i^l(\tau)) \\
&= \log\left(\exp(\Delta S_i^l(\tau) + S_i^l(\tau-w) + g^l) + \exp(\Delta N_i^l(\tau) + N_i^l(\tau-w))\right) \\
&\quad - \log\left(\exp(S_i^l(\tau-w) + g^l) + \exp(N_i^l(\tau-w))\right) \qquad (5.10)
\end{aligned}$$

where $g^l = \log(g)$. The corrupted-speech Cepstral delta coefficients have been rewritten in terms of the static and delta coefficients of the clean speech and interfering noise. Note that the expression for the delta coefficient at time τ is dependent on the static coefficients at time $\tau - w$.

The mismatch function given in equation 5.10 has been derived for delta coefficients. The same form of analysis may be applied to delta-delta parameters. Here, simple differences of delta parameters are calculated. An added constraint is also introduced that $w = w_a$. The mismatch function is of the form

$$\Delta^2 O_i^l(\tau) = \mathcal{F}^{\Delta^2}(S_i^l(\tau), N_i^l(\tau), \Delta S_i^l(\tau-w), \Delta N_i^l(\tau-w), \Delta^2 S_i^l(\tau), \Delta^2 N_i^l(\tau)) \qquad (5.11)$$

and in Appendix A this is shown to be[1]

$$\begin{aligned}
\Delta^2 O_i^l(\tau) = \log(&\exp(\Delta^2 S_i^l(\tau) + 2(S_i^l(\tau) - O_i^l(\tau))) + \exp(\Delta^2 N_i^l(\tau) + 2(N_i^l(\tau) - O_i^l(\tau))) + \\
&\exp(\Delta^2 N_i^l(\tau) + \Delta N_i^l(\tau-w) - \Delta S_i^l(\tau-w) + S_i^l(\tau) + N_i^l(\tau) - 2O_i^l(\tau)) + \\
&\exp(\Delta^2 S_i^l(\tau) + \Delta S_i^l(\tau-w) - \Delta N_i^l(\tau-w) + S_i^l(\tau) + N_i^l(\tau) - 2O_i^l(\tau))) \qquad (5.12)
\end{aligned}$$

where $O_i^l(\tau)$ is described by the static parameter mismatch function described in equation 5.6.

If linear regression coefficients are used to calculate the delta and delta-delta parameters no simple mismatch function is possible and the technique described here cannot be used to compensate the delta and delta-delta parameters.

5.3.3 Mismatch Function for Additive and Convolutional Noise

In situations where there is channel distortion as well as additive noise, the corrupted-speech observations are described by

$$O_i(\tau) = H_i S_i(\tau) + N_i(\tau) \qquad (5.13)$$

where **H** represents the channel difference between training and testing. Note that **H** now subsumes the gain term g used in equation 5.6. There are some additional assumptions made in stating this additive and convolutional noise mismatch function from that in equation 1.5. The convolutional noise is assumed to be constant over time and independent of the level of the incoming signal. Thus, non-linear gain distortions due to, for example, the type of microphone being used, may not be compensated for by using this equation.

[1] The use of g has been dropped in this expression to simplify it.

Furthermore, variations of the convolutional noise within a frequency bin are also ignored. Expressing this equation in the same form as equation 5.6 yields

$$\begin{aligned} O_i^l(\tau) &= \mathcal{F}^H(S_i^l(\tau), N_i^l(\tau), H_i^l) \\ &= \log(\exp(S_i^l(\tau) + H_i^l) + \exp(N_i^l(\tau))) \end{aligned} \tag{5.14}$$

This assumes that the noise model statistics are obtained in the test channel conditions as shown in figure 1.2. There is now an additional problem, as there are no statistics for the convolutional noise component **H**. It is necessary to estimate these statistics in the new test condition if they are not available a-priori. This estimation process is described in section 5.8.

The additive and convolutional noise mismatch function can be related to the additive noise mismatch functions. For the static case

$$\mathcal{F}^H(S_i^l(\tau), N_i^l(\tau), H_i^l) = \mathcal{F}(S_i^l(\tau) + H_i^l, N_i^l(\tau)) \tag{5.15}$$

For the clean speech the dynamic parameters are not altered by the addition of convolutional noise, provided the noise is not varying with time. Thus, for example the delta parameter mismatch function when convolutional noise is present is

$$\Delta O_i^l(\tau) = \mathcal{F}^\Delta(S_i^l(\tau - w) + H_i^l, N_i^l(\tau - w), \Delta S_i^l(\tau), \Delta N_i^l(\tau)) \tag{5.16}$$

and similarly for the delta-delta parameters. It is interesting to note that the dynamic coefficients of the corrupted-speech distributions are dependent on the convolutional noise.

It is easy to see that once **H** has been estimated, the static speech parameters may be shifted and the same estimation techniques used in the additive and convolutional noise case as for the simpler additive noise one.

5.4 Variations on the Mismatch Function

This section describes alternative forms of the mismatch functions described above that may be used within the PMC framework.

5.4.1 Domain of Noise Addition

In the same vein as the generalised spectral subtraction technique, it is not necessary to combine the speech and noise in the power spectral domain, or indeed to obtain statistics for the speech and noise in this domain. For many implementations of MFCC parameters, the combining of the FFT values is performed in the magnitude domain. Thus, statistics about the speech and noise are only available in the magnitude, not power, domain. However, scaling these parameters appropriately, assuming that the variation of the parameters within each of the mel-spaced frequency bins is small, will yield the power, or any other higher-order, domain statistics. The mismatch function for the static parameters could then be rewritten as

$$O_i^l(\tau) = \frac{1}{\gamma}\log(\exp(\gamma S_i^l(\tau)) + \alpha\exp(\gamma N_i^l(\tau))) \tag{5.17}$$

Delta and delta-delta compensation will be similarly altered. Here there are two additional variables α and γ. If, for example, the statistics are generated in the magnitude domain, then $\gamma = 1$ represents addition in the linear magnitude domain, $\gamma = 2$ addition in the linear power domain and so on. By setting γ to a large value it is also possible to approximate the max assumption used in some masking schemes. Normally α will be set to one, however, it could be varied in a similar fashion to the over-estimation factor sometimes used in spectral subtraction.

Neither α nor γ have been optimised for the work here. In all systems the noise is assumed additive in the power domain and no over-estimation factor used.

5.4.2 Matrix-Based Dynamic Coefficients

There are also alternatives to the mismatch function for the dynamic parameters. Instead of generating the delta and delta-delta parameters using differences of the static and delta parameters respectively, they may be obtained using the matrix operation

$$\begin{bmatrix} \mathbf{O}^c(\tau) \\ \boldsymbol{\Delta}\mathbf{O}^c(\tau) \\ \boldsymbol{\Delta}^2\mathbf{O}^c(\tau) \end{bmatrix} = \begin{bmatrix} \mathbf{0}^n & \mathbf{I}^n & \mathbf{0}^n \\ -\frac{1}{2w}\mathbf{I}^n & \mathbf{0}^n & \frac{1}{2w}\mathbf{I}^n \\ \frac{1}{w^2}\mathbf{I}^n & -\frac{2}{w^2}\mathbf{I}^n & \frac{1}{w^2}\mathbf{I}^n \end{bmatrix} \begin{bmatrix} \mathbf{O}^c(\tau - w) \\ \mathbf{O}^c(\tau) \\ \mathbf{O}^c(\tau + w) \end{bmatrix} \tag{5.18}$$

where $\mathbf{I}^n$ is the n dimensional identity matrix and $\mathbf{0}^n$ is the n dimensional zero matrix. This transformation matrix is invertible, so given the statistics about $\mathbf{O}^c(\tau)$, $\boldsymbol{\Delta}\mathbf{O}^c(\tau)$ and $\boldsymbol{\Delta}^2\mathbf{O}^c(\tau)$, it is possible to obtain statistics and correlations for $\mathbf{O}^c(\tau - w)$, $\mathbf{O}^c(\tau)$ and $\mathbf{O}^c(\tau + w)$, even if diagonal covariance matrices are used. These may then be compensated using the standard static parameter compensation.

There are a couple of problems with this matrix transformation. Correlations exist between the static and the delta-delta parameters and the delta and delta-delta parameters, even when the observations are assumed independent in time, so the diagonal covariance approximation is poor. In addition, the noise-corrupted speech dynamic parameters are not directly calculated, they are estimated from differences of compensated static parameters. Thus, any errors in the static parameter compensation will be accentuated in the dynamic parameters estimated.

The first problem may be partially overcome by altering the form of the matrix.

$$\begin{bmatrix} \mathbf{O}^c(\tau) \\ \boldsymbol{\Delta}\mathbf{O}^c(\tau) \\ \boldsymbol{\Delta}^2\mathbf{O}^c(\tau) \end{bmatrix} = \begin{bmatrix} \mathbf{0}^n & \mathbf{0}^n & \mathbf{I}^n & \mathbf{0}^n & \mathbf{0}^n \\ \mathbf{0}^n & -\frac{1}{2w}\mathbf{I}^n & \mathbf{0}^n & \frac{1}{2w}\mathbf{I}^n & \mathbf{0}^n \\ \frac{1}{4w^2}\mathbf{I}^n & \mathbf{0}^n & -\frac{1}{2w^2}\mathbf{I}^n & \mathbf{0}^n & \frac{1}{4w^2}\mathbf{I}^n \end{bmatrix} \begin{bmatrix} \mathbf{O}^c(\tau - 2w) \\ \mathbf{O}^c(\tau - w) \\ \mathbf{O}^c(\tau) \\ \mathbf{O}^c(\tau + w) \\ \mathbf{O}^c(\tau + 2w) \end{bmatrix} \tag{5.19}$$

Here it is necessary to model the observations at time $\tau + w$, $\tau + 2w$, τ, $\tau - w$, $\tau - 2w$, as the matrix used to obtain the dynamic coefficients is not invertible. The correlations between the delta and delta-delta parameters in the previous matrix transformation have been reduced, however there is still a strong correlation between the static and the delta-delta parameters. This correlation exists in most HMM-based speech recognition systems incorporating delta-delta parameters, whether they are based on simple differences or linear regression dynamic coefficients.

Using the matrix transform in equation 5.19 to obtain the dynamic coefficients is identical to the simple difference dynamic coefficients described in the standard mismatch functions. Unfortunately it does have an additional drawback. In any noise compensation technique it is desirable that when no interfering noise is present the models generated are identical to the standard clean system. Using the matrix based mismatch functions this is only possible if the correlations between all the parameters are modelled in the case of estimating the covariance matrices. This need not necessarily be a full covariance matrix, but may just model the correlations between $O_i^c(\tau)$ and $\Delta O_i^c(\tau)$ etc. In section 5.10.1 this is referred to as the *Extended* covariance approximation. If only the means of the models are to be updated then the use of these matrix style mismatch functions will yield the correct performance in clean conditions.

With this form of modelling it is possible to obtain mismatch functions for the linear regression parameters popular in state-of-the-art speech recognition systems. In this case all the observations from time $\tau - 2w$ to $\tau + 2w$ must be modelled. The appropriate matrix to obtain the dynamic coefficients may then be easily derived.

A slightly different version of this dynamic coefficient modelling has previously been proposed [5, 92]. In this approach the aim is not to obtain a set of dynamic features related to delta parameters and delta-delta parameters, but to use Two-Dimensional Cepstral (TDC) time matrices as the speech parameter set. This technique has been used in DTW-based systems [5] and HMM-based systems for both clean [92] and noise-corrupted environments [60]. There are some important differences to the matrices described here. In the TDC both the higher-order time and spatial Cepstral parameters are truncated, thus forcing smoothing in time and frequency. Whereas in the work described here, the aim is to modify the standard parameterisation as little as possible, since the standard Cepstral parameterisation, with delta and delta-delta parameters, has been shown to work well for large vocabulary systems.

5.4.3 Continuous-Time Dynamic Coefficients

The dynamic parameters may be calculated using a Continuous-Time approximation [31]. By considering the delta parameters to be an approximation to the first derivative of the static parameters

$$\Delta O_i^l \approx \frac{\partial O_i^l}{\partial t} = \left(\frac{\exp(S_i^l)}{\exp(S_i^l) + \exp(N_i^l)}\right)\frac{\partial S_i^l}{\partial t} + \left(\frac{\exp(N_i^l)}{\exp(S_i^l) + \exp(N_i^l)}\right)\frac{\partial N_i^l}{\partial t} \tag{5.20}$$

For the compensation process, the effects of the variances of the speech and the noise parameters on the non-linear combination of the speech and noise shown in equation 5.20 are assumed to be small. In addition, the means of the noise delta parameters are assumed to be approximately zero. Hence, the observation values may simply be replaced by the means and equation 5.20 may be rewritten as

$$\Delta\hat{\mu}_i^l \approx \left(\frac{\exp(\mu_i^l)}{\exp(\mu_i^l) + \exp(\tilde{\mu}_i^l)}\right)\Delta\mu_i^l \tag{5.21}$$

In a similar fashion, the delta-delta parameters may be considered to be approximately the second derivative, hence

$$\begin{aligned}\Delta^2 O_i^l \approx \frac{\partial^2 O_i^l}{\partial t^2} \quad &= \quad \left(\frac{\exp(S_i^l)}{\exp(S_i^l) + \exp(N_i^l)}\right)\frac{\partial^2 S_i^l}{\partial t^2} + \left(\frac{\exp(N_i^l)}{\exp(S_i^l) + \exp(N_i^l)}\right)\frac{\partial^2 N_i^l}{\partial t^2} \\ &+ \left(\frac{\exp(S_i^l + N_i^l)}{(\exp(S_i^l) + \exp(N_i^l))^2}\right)\left[\left(\frac{\partial S_i^l}{\partial t}\right)^2 + \left(\frac{\partial N_i^l}{\partial t}\right)^2 - 2\frac{\partial S_i^l}{\partial t}\frac{\partial N_i^l}{\partial t}\right]\end{aligned} \tag{5.22}$$

Again, if the mean of the second derivative of the noise is assumed to be zero[2] and the non-linearities ignored, then

$$\begin{aligned}\Delta^2\hat{\mu}_i^l &\approx \left(\frac{\exp(\mu_i^l)}{\exp(\mu_i^l)+\exp(\tilde{\mu}_i^l)}\right)\Delta^2\mu_i^l \\ &\quad + \left(\frac{\exp(\mu_i^l+\tilde{\mu}_i^l)}{(\exp(\mu_i^l)+\exp(\tilde{\mu}_i^l))^2}\right)\left(\Delta\Sigma_{ii}^l + (\Delta\mu_i^l)^2 + \Delta\tilde{\Sigma}_{ii}^l\right) \end{aligned} \tag{5.23}$$

This style of compensation has the advantage that the delta parameters are not limited to being simple differences, the more standard linear regression based dynamic coefficients may be compensated and no additional parameters need to be modelled.

5.5 Training HMMs on Statistical Data

Having derived expressions for the effects of noise on the various parameters of the feature vector, it is necessary to obtain expressions for estimating the parameters of the corrupted-speech HMMs. Instead of having observations, as in the case of standard HMM training, only statistics about the speech and noise are available. It is therefore necessary to modify the standard HMM training algorithms to allow for the training of models on statistical data. For this work a ML estimate will be used.

The standard re-estimation formula of the new mean of mixture M_m of state s_j of the corrupted-speech model can be expressed in terms of the expected values of observations instead of summations[3].

$$\hat{\mu}_{jm} = \frac{\mathcal{E}\left\{L_{jm}(\tau)\mathbf{O}^c(\tau)\right\}}{\mathcal{E}\left\{L_{jm}(\tau)\right\}} \tag{5.24}$$

Similar expressions for the variance and mixture weights in terms of expected values may be obtained.

As HMMs are used to model the training data there are no longer explicit observations at each time interval τ. Instead there are PDFs describing the observations for a particular state. This PDF will be a function of known statistics, represented here in the case of additive noise by the clean speech models and the noise model. Let $\mathbf{O}^c$ be the random variable associated with actual observations $\mathbf{O}^c(\tau)$. Since HMMs are being used to model the training data, it is necessary to assume that the frame/state alignment is not altered, as the state transition matrix contains the temporal information from the training dataset. The complete dataset is now the alignment of a particular "observation" with a particular component, M_m, of the current state. Thus for one state, s_j, of the corrupted-speech model, $\hat{\mathcal{M}}$,

$$p(M_m|\mathbf{O}^c, s_j, \hat{\mathcal{M}}) = \frac{c_m\mathcal{N}(\mathbf{O}^c;\mu_{jm},\mathbf{\Sigma}_{jm})}{\sum_{i=1}^{M} c_i\mathcal{N}(\mathbf{O}^c;\mu_{ji},\mathbf{\Sigma}_{ji})} = \mathcal{K}_m(\mathbf{O}^c) \tag{5.25}$$

[2] In the original publication [31] the variance of the noise source is also assumed to be small compared to that of the speech.

[3] The case shown here has only static parameters in the feature vector. Identical expressions apply if dynamic coefficients are used.

For multiple Gaussian mixture component HMMs the re-estimation formula becomes[4]

$$\hat{\mu}_{jm} = \frac{\int_{\mathcal{R}^n} \mathcal{K}_m(\mathbf{O}^c)\mathbf{O}^c\mathcal{L}(\mathbf{O}^c)\, d\mathbf{O}^c}{\int_{\mathcal{R}^n} \mathcal{K}_m(\mathbf{O}^c)\mathcal{L}(\mathbf{O}^c)\, d\mathbf{O}^c} \tag{5.26}$$

Similarly the re-estimation for the covariance matrix becomes

$$\begin{aligned}\hat{\mathbf{\Sigma}}_{jm} &= \frac{\int_{\mathcal{R}^n} \mathcal{K}_m(\mathbf{O}^c)(\mathbf{O}^c - \hat{\mu}_{jm})(\mathbf{O}^c - \hat{\mu}_{jm})^T\mathcal{L}(\mathbf{O}^c)\, d\mathbf{O}^c}{\int_{\mathcal{R}^n} \mathcal{K}_m(\mathbf{O}^c)\mathcal{L}(\mathbf{O}^c)\, d\mathbf{O}^c} \\ &= \left(\frac{\int_{\mathcal{R}^n} \mathcal{K}_m(\mathbf{O}^c)\mathbf{O}^c\mathbf{O}^{cT}\mathcal{L}(\mathbf{O}^c)\, d\mathbf{O}^c}{\int_{\mathcal{R}^n} \mathcal{K}_m(\mathbf{O}^c)\mathcal{L}(\mathbf{O}^c)\, d\mathbf{O}^c}\right) - \hat{\mu}_{jm}\hat{\mu}_{jm}^T \end{aligned} \tag{5.27}$$

and for the weights

$$\hat{c}_{jm} = \int_{\mathcal{R}^n} \mathcal{K}_m(\mathbf{O}^c)\mathcal{L}(\mathbf{O}^c)\, d\mathbf{O}^c \tag{5.28}$$

The transition matrix will remain the same as the frame/state alignment is assumed to be unaltered. All the above expressions may be seen to be functions of

$$\mathcal{E}\left\{\mathcal{K}_m(\mathbf{O}^c)\right\} = \int_{\mathcal{R}^n} \mathcal{K}_m(\mathbf{O}^c)\mathcal{L}(\mathbf{O}^c)\, d\mathbf{O}^c \tag{5.29}$$

$$\mathcal{E}\left\{\mathbf{O}^c\mathcal{K}_m(\mathbf{O}^c)\right\} = \int_{\mathcal{R}^n} \mathbf{O}^c\mathcal{K}_m(\mathbf{O}^c)\mathcal{L}(\mathbf{O}^c)\, d\mathbf{O}^c \tag{5.30}$$

$$\mathcal{E}\left\{\mathbf{O}^c\mathbf{O}^{cT}\mathcal{K}_m(\mathbf{O}^c)\right\} = \int_{\mathcal{R}^n} \mathbf{O}^c\mathbf{O}^{cT}\mathcal{K}_m(\mathbf{O}^c)\mathcal{L}(\mathbf{O}^c)\, d\mathbf{O}^c \tag{5.31}$$

If a single Gaussian mixture model is to be estimated, $M = 1$, then

$$\mathcal{K}_m(\mathbf{O}^c) = \mathcal{K}(\mathbf{O}^c) = 1 \tag{5.32}$$

and the formulae may be simplified accordingly.

The above scheme is an iterative one in the same way as standard HMM training. However, if the frame/state component alignment is also assumed to be unaltered, then single-pass, non-iterative, estimation is possible. In this single-pass case, all the mixtures are treated independently, as if they were each in a distinct state and the simplification shown in equation 5.32 is applied. Note that in this case, the mixture weights will remain unaltered. This assumption that the frame/state component alignment does not alter is implicit when using hypothesised Wiener filtering or linear regression adaptation. This is referred to as *non-iterative PMC*.

If the frame/state component alignment within a state is allowed to vary, it is necessary to keep $\mathcal{K}_m(\mathbf{O}^c)$ within the integration. For the case of additive noise, this introduces additional problems in the compensation process, as $\mathcal{K}_m(\mathbf{O}^c)$ is a function of all the elements of the feature vector. This is discussed in more detail in section 5.7. Where this iterative estimation scheme is used, it is referred to as *iterative PMC*.

[4]The conditioning of the likelihood on the speech and noise models and states has been dropped for clarity.

5.6 Non-Iterative PMC

The simplest, and most computationally efficient, versions of PMC are non-iterative. By that, it is meant that the frame/state component alignment is assumed unaltered. Thus, the simplifying assumptions given in the previous section apply and it is not necessary to re-estimate the transition probabilities or mixture component weights.

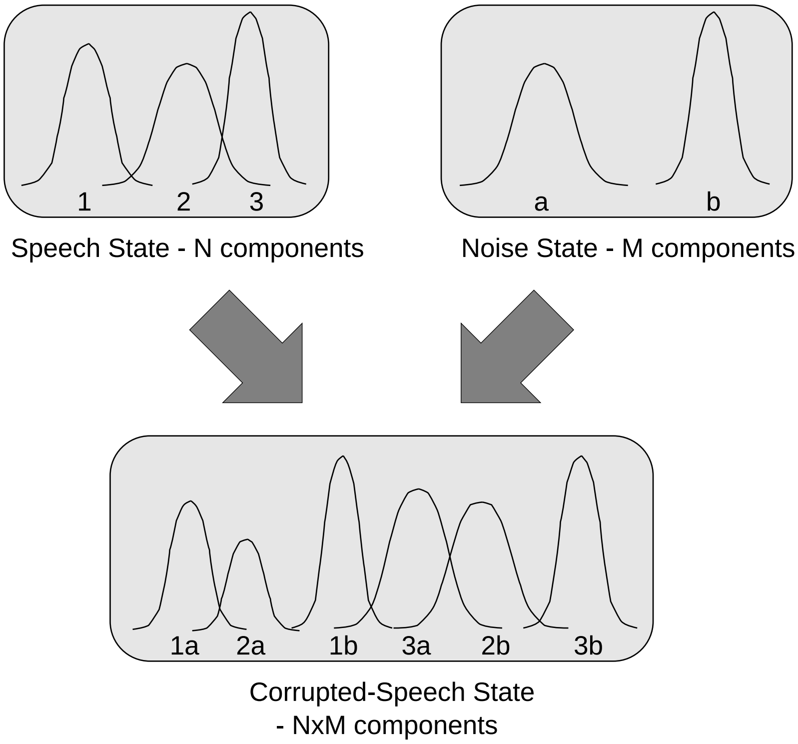


Figure 5.2: Non-iterative parallel model combination

Figure 5.2 shows how the components combine together using this assumption for a particular speech state and noise state. In the example given, the speech state has three components, labelled "1", "2", and "3", and the noise state two components "a" and "b". The resulting corrupted-speech state has six components associated with it, marked according to the speech and noise components that were used to generate it. Thus "2a" indicates speech component "2" and noise component "a". It is, of course, possible to merge corrupted-speech distributions to reduce the number of components, however this does not alter the fundamental frame alignments. For simplicity of notation, a particular speech and noise component pairing is chosen to describe the compensation processes.

Applying the equations from the previous section to the additive noise mismatch functions yields the following expressions for the mean and variance for the static parameters

$$\hat{\mu}_i^l = \mathcal{E}\left\{\mathcal{F}(S_i^l, N_i^l)\right\} \tag{5.33}$$

and

$$\hat{\Sigma}_{ij}^l = \mathcal{E}\left\{\mathcal{F}(S_i^l, N_i^l)\mathcal{F}(S_j^l, N_j^l)\right\} - \hat{\mu}_i^l\hat{\mu}_j^l \tag{5.34}$$

For the delta parameters the mean is given by

$$\Delta\hat{\mu}_i^l = \mathcal{E}\left\{\mathcal{F}^{\Delta}(S_i^{\prime l}, N_i^{\prime l}, \Delta S_i^l, \Delta N_i^l)\right\} \tag{5.35}$$

and the covariance by

$$\Delta\hat{\Sigma}_i^l = \mathcal{E}\left\{\mathcal{F}^{\Delta}(S_i^{\prime l}, N_i^{\prime l}, \Delta S_i^l, \Delta N_i^l)\mathcal{F}^{\Delta}(S_j^{\prime l}, N_j^{\prime l}, \Delta S_j^l, \Delta N_j^l)\right\} - \Delta\hat{\mu}_i^l\Delta\hat{\mu}_j^l \tag{5.36}$$

where $'$ indicates that the statistics are generated at time $\tau - w$. Finally, the delta-delta parameters have a similar form where

$$\Delta^2\hat{\mu}_i^l = \mathcal{E}\left\{\mathcal{F}^{\Delta^2}(S_i^l, N_i^l, \Delta S_i^{\prime l}, \Delta N_i^{\prime l}, \Delta^2 S_i^l, \Delta^2 N_i^l)\right\} \tag{5.37}$$

and

$$\begin{aligned}\Delta^2\hat{\Sigma}_{ii}^l &= \mathcal{E}\left\{\mathcal{F}^{\Delta^2}(S_i^l, N_i^l, \Delta S_i^{\prime l}, \Delta N_i^{\prime l}, \Delta^2 S_i^l, \Delta^2 N_i^l)\mathcal{F}^{\Delta^2}(S_j^l, N_j^l, \Delta S_j^{\prime l}, \Delta N_j^{\prime l}, \Delta^2 S_j^l, \Delta^2 N_j^l)\right\} \\ &\quad -\Delta^2\hat{\mu}_i^l\Delta^2\hat{\mu}_j^l \end{aligned} \tag{5.38}$$

There are no closed-form expressions for either the compensated mean, or covariance matrix for the static, delta or delta-delta parameters. To obtain values for the above expressions, it is necessary to make approximations about the forms of the distribution, or to use numerical integration techniques. Both of these approaches are investigated.

5.6.1 Numerical Integration

The estimation of the static parameters may be performed using numerical integration. If implemented directly this would involve two-dimensional integration to estimate the mean, and four-dimensional integration to estimate the covariance matrix for the static parameters, and even higher numbers of dimension to integrate over for the delta and delta-delta parameters. This would be computationally very expensive. A more efficient form for the integration is therefore required.

The estimate of the corrupted mean of the static parameters may be written as

$$\hat{\mu}_i^l = \mu_i^l + \mathcal{E}\{\log(\exp(N_i^l - S_i^l) + 1)\} \tag{5.39}$$

The sum or difference of two Gaussian distributed variables is itself Gaussian distributed. Thus, single-dimensional numerical integration may be used for the means. After some simple manipulation, this expression can be transformed into a standard Gaussian integration problem.

$$\int_{-\infty}^{\infty} f(x)\exp(-x^2)\,dx \approx \sum_{i=1}^{I} w_i f(x_i) \tag{5.40}$$

where x_i and w_i are given by the Gauss-Hermite abscissa and weights respectively [1]. For the covariance matrix estimation it is necessary to rotate the coordinate frame, so that the two dimensions may be integrated independently. Further details of the implementation of the numerical integration are given in Appendix B.

A similar approach to that used for the numerical integration of the static parameters is used to estimate the delta parameters. Rewriting equation 5.35, the expression for the corrupted delta mean is

$$\Delta\hat{\mu}_i^l = \Delta\tilde{\mu}_i^l - \mathcal{E}\{\log(\exp(S_i'^l - N_i'^l) + 1)\} + \mathcal{E}\{\log(\exp(S_i'^l + \Delta S_i^l - N_i'^l - \Delta N_i^l) + 1)\} \tag{5.41}$$

Again, by noting that the sum of two Gaussian distributed variables is itself Gaussian distributed, the above expression may be cast in terms of single-dimension numerical integrations. The details of this implementation are also given in Appendix B.

Delta-delta parameters may be compensated in a similar fashion, however owing to the nature of the mismatch function, the overhead associated with such a task is very high.

Estimating the parameters using this form of numerical integration places no constraints on the variances estimated. Given the finite number of points used in the numerical integration, this may yield negative variances. For the standard systems developed, however, this was found not to be a problem.

5.6.2 Log-Normal Approximation

The use of numerical integration will yield good estimates of the corrupted-speech distributions, however the computational load of such a technique is high.

If it is assumed that the sum of two Log-normally distributed variables is itself approximately Log-normally distributed then it is only necessary to calculate the mean and the variance of the corrupted-speech model in the Linear-Spectral domain to be able to derive the corrupted-speech model in the Cepstral domain. Given the assumptions that the speech and noise are independent and additive in the Linear-Spectral domain, the corrupted-speech static parameters in the Linear-Spectral domain are

$$\hat{\mu} = g\mu + \tilde{\mu} \tag{5.42}$$

$$\hat{\mathbf{\Sigma}} = g^2\mathbf{\Sigma} + \tilde{\mathbf{\Sigma}} \tag{5.43}$$

where the parameters μ and $\mathbf{\Sigma}$ are the mean and covariance matrix respectively of the Log-normal distribution associated with the Gaussian distribution $\{\mu^l, \mathbf{\Sigma}^l\}$. Similarly for the noise, $\tilde{\mu}$ and $\tilde{\mathbf{\Sigma}}$ are related to $\tilde{\mu}^l$ and $\tilde{\mathbf{\Sigma}}^l$. The parameters of the clean speech in the Linear-Spectral and Log-Spectral domains are related by

$$\mu_i = \exp(\mu_i^l + \Sigma_{ii}^l/2) \tag{5.44}$$

$$\mathbf{\Sigma}_{ij} = \mu_i\mu_j\left[\exp(\Sigma_{ij}^l) - 1\right] \tag{5.45}$$

and similarly for the noise. See Appendix C for derivation. Since the corrupted speech is assumed to be Log-Normally distributed in the Linear-Spectral domain, the required distribution in the Log-Spectral domain, $\{\hat{\mu}^l, \hat{\mathbf{\Sigma}}^l\}$, may be obtained using the inverse of the above expressions. Hence

$$\hat{\mu}_i^l \approx \log(\hat{\mu}_i) - \frac{1}{2}\log\left(\frac{\hat{\Sigma}_{ii}}{\hat{\mu}_i^2} + 1\right) \tag{5.46}$$

$$\hat{\Sigma}_{ij}^l \approx \log\left(\frac{\hat{\Sigma}_{ij}}{\hat{\mu}_i\hat{\mu}_j} + 1\right) \tag{5.47}$$

Note that this technique enforces the constraint that the estimate of the variance is positive.

The sum of two Log-normally distributed variables is not itself Log-normally distributed, as illustrated in Chapter 4. This technique only matches the first two moments of the corrupted-speech distribution in the Linear-Spectral domain. Due to the higher-order statistics in the Linear-Spectral domain and the non-linearity of the *log()* transformation, the resulting distribution in the Log-Spectral domain will not have the same first two moments as the true corrupted-speech distribution.

The mismatch function for the delta parameters, equation 5.10, may be considered as the combination of two terms. Each of these terms has the same form as the static parameter compensation, i.e. $\log(\exp(a)+\exp(b))$. Indeed the second term is the same as the static compensation using statistics at time $\tau - w$. The Log-Normal approximation can therefore be used to estimate a "corrupted-speech" distribution for each of the terms. However, it is then necessary to combine the two distributions. This is not a problem for the means and a good estimate of the corrupted mean will be obtained. For the covariance matrix, it is not possible to simply sum the two terms since the variances are not independent of one another, $S_i^l(\tau - w)$ and $N_i^l(\tau - w)$ appear in both expressions. To calculate the exact correlations between the two terms is a complicated function of the speech and noise parameters. Thus the Log-Normal approximation cannot simply be applied to compensating the delta parameter covariance matrix. This is also true for the delta-delta parameters.

This Log-Normal approximation was used in the original implementation of PMC [24].

5.6.3 Log-Add Approximation

The simplest approximation is the Log-Add approximation. Here, the variances are assumed to be small, so for the static parameters it is possible to write

$$\hat{\mu}_i^l \approx \mathcal{F}(\mu_i^l, \tilde{\mu}_i^l) = \log(\exp(\mu_i^l) + \exp(\tilde{\mu}_i^l)) \tag{5.48}$$

This may be viewed as a simplification of the Log-Normal approximation for the static parameters [25]. An analogy with the hypothesised Wiener filtering [9, 10] for these parameters may easily be seen by rewriting the expression

$$\hat{\mu}^c \approx \mu^c + \mathbf{C}\left[\log(\exp(\mu^l) + \exp(\tilde{\mu}^l)) - \mu^l\right] \tag{5.49}$$

Thus the Log-Add approximation is a simple and efficient implementation of hypothesised Wiener filtering, where the means of the speech and noise are obtained from HMMs instead of having to be stored explicitly.

The Log-Add approximation may also be applied to the dynamic coefficients. Again by ignoring the effect of the variance of the data on the non-linearity of the mismatch functions, the mean of the corrupted-speech delta parameters may be calculated as

$$\Delta\hat{\mu}_i^l \approx \mathcal{F}^{\Delta}(\mu_i'^l, \tilde{\mu}_i'^l, \Delta\mu_i^l, \Delta\tilde{\mu}_i^l) \tag{5.50}$$

The delta-delta parameters may be written as

$$\Delta^2\hat{\mu}_i^l(\tau) \approx \mathcal{F}^{\Delta^2}(\mu_i^l, \tilde{\mu}_i^l, \Delta\mu_i'^l, \Delta\tilde{\mu}_i'^l, \Delta^2\mu_i^l, \Delta^2\tilde{\mu}_i^l) \tag{5.51}$$

For the static parameters this technique has previously been used for speech recognition [60, 62] on small vocabulary tasks.

There is an additional advantage in using the Log-Add approximation. Since the covariance matrix is not used in the compensation it is unnecessary to map it from the Cepstral domain to the Log-Spectral domain and vice-versa. This is a major reduction in computational load. Furthermore, if the means are stored in the Log-Spectral domain, only a single DCT to map the estimated corrupted-speech means from the Log-Spectral to the Cepstral domain is required. Where truncated Cepstral parameters are used, storing the means in the Log-Spectral domain results in a slight increase in the number of parameters stored.

5.7 Iterative PMC

Though in some cases computationally efficient, the non-iterative PMC techniques are not expected to be "true" ML estimates of the corrupted-speech distributions. Chapter 4 illustrated that combining the speech and noise yields non-Gaussian distributions. The implicit assumption in the non-iterative PMC schemes is that for a particular component pairing the corrupted-speech distribution is approximately Gaussian. Improved modelling of these complex distributions may be achieved using iterative PMC, where the frame/state component alignment within a state is allowed to vary. As mentioned in section 5.5, the problem with an iterative PMC scheme is that it is necessary to calculate $\mathcal{K}_m(\mathbf{O}^c)$ which is a function of all the elements in the feature vector. A feasible iterative PMC scheme which allows this value to be calculated at a reasonable computational cost is Data-driven PMC (DPMC).

DPMC is a simple method of implementing iterative PMC. Examining the statistical re-estimation formulae, the value of $\mathcal{K}_m(\mathbf{O}^c)$ is a function of all dimensions of the feature vector. Hence, if numerical integration is to be performed it must be performed in all n dimensions of the observation vector, an intractable task when n is any reasonably large value. In DPMC, the integration is performed by generating corrupted speech observations. These are obtained by generating a speech "observation" and a noise "observation" for a particular pair of speech and noise states and combining them according to the appropriate mismatch function. Having generated a set of "observations" for a particular state pair, standard multiple mixture component single-emitting-state HMM training can be used. The only problem is now to get an initial set of Gaussians to start the iterative process. This initial set may take many forms depending on the number of components to be estimated, for example, the components from non-iterative PMC may be used. The basic concept of DPMC is shown in figure 5.3. Note here, in contrast to figure 5.2, there is no explicit mapping from the speech and noise components to those in the corrupted-speech model.

In addition to giving added flexibility in the number of components to be estimated, DPMC is faster, particularly if run in a non-iterative fashion, than the Numerical Integration scheme described and still allows compensation of all the model parameters. In the Numerical Integration scheme, the calculation of the covariance matrix must be performed in the Log-Spectral domain. For a 24 dimensional Log-Spectral domain vector, this results in 300 separate numerical integrations, which, in the case of the dynamic coefficients, may each require a number of Gaussian integrations. On the other hand by generating corrupted-speech "observations", it is only necessary to accumulate the appropriate statistics. The computational overhead is dependent on the number of points generated. DPMC, when used in a non-iterative fashion, will give approximately the same set of models as the Numerical Integration approximation.

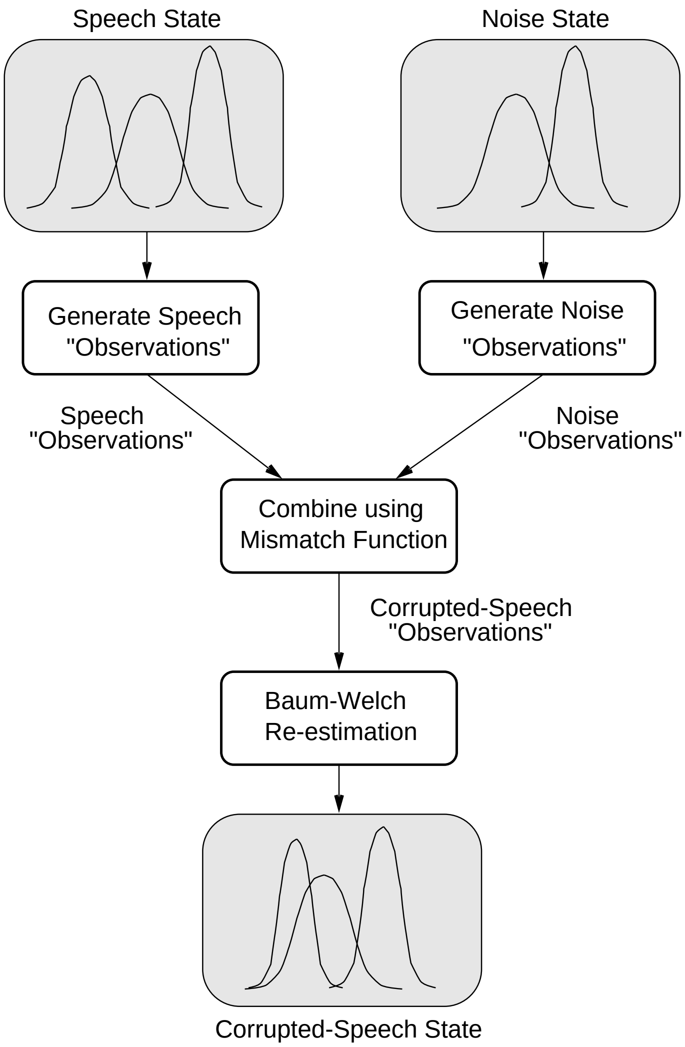


Figure 5.3: Data-driven parallel model combination

The Log-Add approximation may be viewed as a non-iterative, simplified and faster version of DPMC, which just updates the means. Instead of generating many "observations", the Log-Add approximation generates a single "observation" for each of the speech and noise components, at the mean of their respective distributions. These are then combined according to the appropriate mismatch function to yield the corrupted-speech "observation", which is an estimate of the corrupted-speech mean.

5.8 Estimation of the Convolutional Noise Component

This section describes methods to obtain the statistics for the channel difference $\mathbf{H}^c$, where $\mathbf{H}^c = \{H_i^c\}$. Since ML estimation is used as the optimisation criterion in estimating the corrupted-speech parameters, it is preferable to use ML as the criterion for estimating $\mathbf{H}^c$ also. Thus

$$\hat{\mathbf{H}}^c = \arg\max_{\mathbf{H}^c} \left[\mathcal{L}(\mathbf{O}_T^c|\mathbf{H}^c, \mathcal{M}_s, \mathcal{M}_n, \mathcal{V})\right] \tag{5.52}$$

where $\mathbf{O}_T^c$ is the data available to estimate the spectral tilt, $\mathcal{M}_s$ is the set of clean speech models, and $\mathcal{V}$ is the language model used for recognition.

This optimisation may be solved using an EM based approach, however this is computationally very expensive. In order to maximise the probability, it is necessary to compensate every component in all possible state sequences using the current estimate of $\mathbf{H}^c$. In large systems this is impractical. Moreover, there is no closed form for the maximisation, as the estimation of $\mathbf{H}^c$ is tied over all components. Thus, the maximisation must be performed using standard optimisation techniques. Recently this form of optimisation has been combined with the Log-Normal approximation, described in section 5.6.2, to estimate the spectral tilt using steepest descent methods [61].

To lower the computational overhead, the number of models in the system may be reduced. To represent the effects of convolutional noise, only the general structure of the speech needs to be modelled. Thus instead of using the large complex models typically used for recognition, simpler models could be used. In fact, for the purpose of estimating $\mathbf{H}^c$, a single-state multiple-component general speech model, $\mathcal{M}_g$, is sufficient to represent the statistics of the clean speech training data. This of course does not allow the use of the language model, $\mathcal{V}$, but greatly reduces the number of components in the system. It is still necessary to perform a multi-dimensional optimisation task. However, if the corrupted-speech distribution of all the data is modelled by a single-component model, then by combining all the individual corrupted-speech distributions into one, such that

$$\hat{\mu}^c(\mathbf{H}^c) = \sum_{m=1}^{M} c_m \hat{\mu}_m^c(\mathbf{H}^c) \tag{5.53}$$

and

$$\hat{\mathbf{\Sigma}}^c(\mathbf{H}^c) = \sum_{m=1}^{M} c_m \left(\hat{\mathbf{\Sigma}}_m^c(\mathbf{H}^c) + \hat{\mu}_m^c(\mathbf{H}^c)\hat{\mu}_m^c(\mathbf{H}^c)^T\right) - \hat{\mu}^c(\mathbf{H}^c)\hat{\mu}^c(\mathbf{H}^c)^T \tag{5.54}$$

the problems of having to estimate $\mathbf{H}^c$ over many components, and the complete dataset associated with these components, is removed, since all observations now belong to the same distribution. The optimisation task will now be

$$\hat{\mathbf{H}}^c = \arg\max_{\mathbf{H}^c} \left[k - \frac{T}{2}\log(|\hat{\boldsymbol{\Sigma}}^c(\mathbf{H}^c)|) \right.$$
$$\left. -\frac{1}{2}\sum_{t=1}^{T} (\mathbf{O}^c(t) - \hat{\mu}^c(\mathbf{H}^c))^T \, \hat{\boldsymbol{\Sigma}}^c(\mathbf{H}^c)^{-1} \, (\mathbf{O}^c(t) - \hat{\mu}^c(\mathbf{H}^c)) \right] \quad (5.55)$$

where k is the standard normalising constant. Ignoring the variation of the variance with the spectral tilt, this expression is simplified to

$$\hat{\mathbf{H}}^c = \arg\min_{\mathbf{H}^c} \left[\left| \hat{\mu}^c(\mathbf{H}^c) - \frac{1}{T}\sum_{t=1}^{T} \mathbf{O}^c(t) \right| \right] \quad (5.56)$$

Alternatively the optimisation may be performed in the Log-Spectral domain for which the analogous equation is

$$\hat{\mathbf{H}}^l = \arg\min_{\mathbf{H}^l} \left[\left| \hat{\mu}^l(\mathbf{H}^l) - \frac{1}{T}\sum_{t=1}^{T} \mathbf{O}^l(t) \right| \right] \quad (5.57)$$

The advantage of optimising the expression in the Log-Spectral domain is that this is the domain in which the convolutional noise simply affects the clean speech. If no constraints, such as smoothness, are applied, each element, H_i^l of $\mathbf{H}^l$, may be independently optimised, greatly simplifying and speeding up the estimation process.

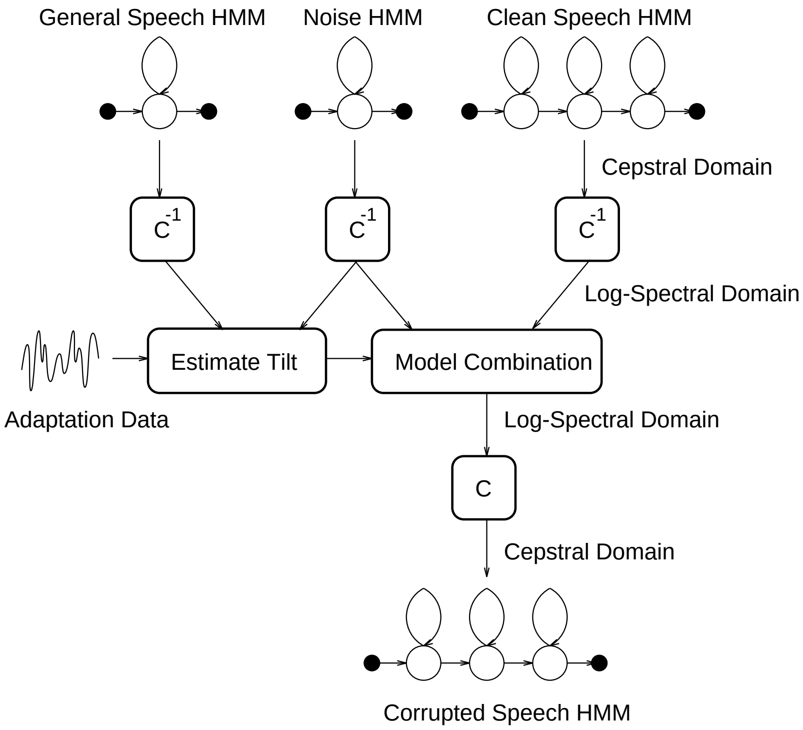


Figure 5.4: Parallel model combination for additive and convolutional noise

The PMC process, when additive and convolutional noise is present, is illustrated in figure 5.4. The additional inputs to the scheme are a general speech model and some adaptation data used to estimate the convolutional noise. As shown in section 5.3.3, once the convolutional noise is known all the standard iterative and non-iterative PMC techniques may be used to estimate the corrupted-speech models.

There are limitations to this approach. It has been reported [73] that a single multivariate Gaussian distribution is not a good model of the corrupted-speech parameters. However, the optimisation may be considered as matching the first moment of the observed data to that of the general speech model, so the higher-order moments may be ignored.

Another problem with this approach is that there is an implicit assumption that the adaptation data is representative of the training data. If there is little adaptation data available, then this assumption may be poor. A simple method to overcome this problem is to use a two-pass approach. Initially the tilt is estimated using equation 5.57. Then, using this estimate of the tilt, the weights of the general speech model, $\mathcal{M}_g$, are updated to reflect the component occupancy of the adaptation data. The optimisation in equation 5.57 is then repeated. This requires that the general speech model components are a "good" representation of the clean speech training database and that the frame/state component alignment obtained is reasonable. The overhead associated with this second pass is small due to the simple general speech model used.

The process of using the current estimate of the convolutional noise component to obtain a new alignment for the general speech model may be repeated many times. However, as the optimisation of the estimate of $\mathbf{H}^c$ is not a true maximisation in a ML sense, it is not guaranteed to converge, nor the likelihood of the adaptation sentence to increase. This problem is accentuated if dynamic coefficients are incorporated into the feature vector, as only the static parameters are used in the estimation of the spectral tilt. So, for the work presented here, if an iterative estimation of the convolutional noise is made, the general speech component weights are only updated once.

Although the techniques described in this section are approximate and do not achieve a true ML estimate, they are fast and may be applied to large vocabulary systems where delta and delta-delta parameters are appended to the feature vector.

5.9 Complex-Noise Environments

In order to handle "complex" or non-stationary noise conditions, such as Machine Gun noise, see Chapter 6, a multi-state noise model may be used. In this case the same estimation techniques apply but now it is no longer possible to know *a-priori*, which noise state to combine with each speech model state. Hence all combinations must be computed and the optimal sequence decoded at run time. The standard method of dealing with this is to use a 3-dimensional Viterbi decoding scheme [90] based on equation 3.25. The combined output probability $b_{jv}(\mathbf{O}^c(\tau))$ in the PMC case corresponds to the distribution obtained by combining state s_j of the speech model with state s_v^n of the noise model. Note that if the clean speech HMMs contain a total of M states and the noise model has N states, then the compensated recogniser will have $M \times N$ states if the direct implementation of PMC is used. Fortunately this is not normally a major problem as 2 or 3 states are usually sufficient for the noise model.

When the noise model is ergodic (i.e. fully connected), then the 3-D decoding scheme can be synthesised using a standard 2-D decoder operating on an expanded model, where each of the original speech states, s_i, is replaced by N compensated states, $s_{i1}, s_{i2}, \ldots, s_{iN}$, with transition probabilities given by

$$\hat{a}_{iu,jv} = a_{ij}\tilde{a}_{uv} \tag{5.58}$$

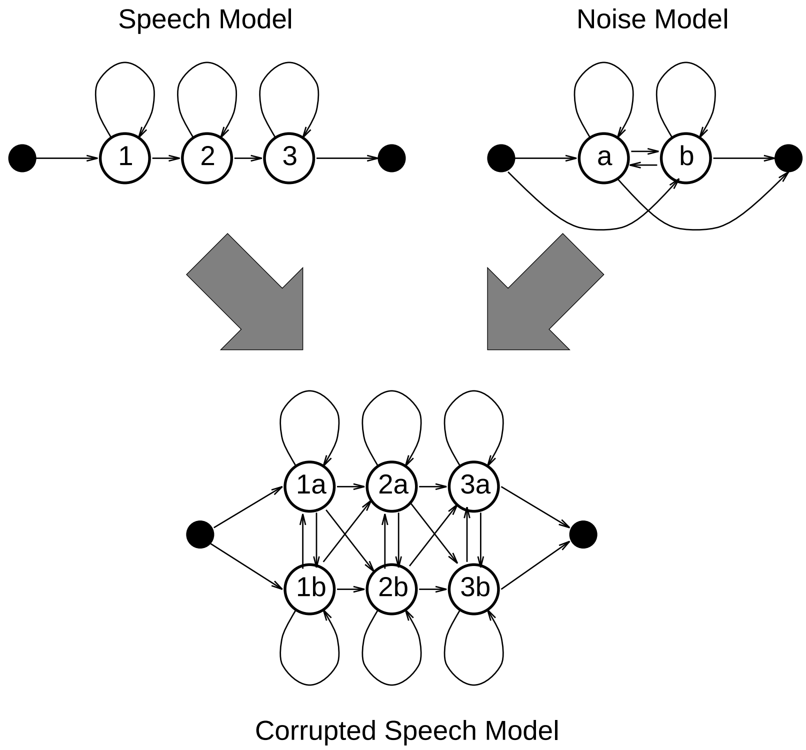


Figure 5.5: Mapping a 3-D Viterbi decoder to a 2-D Viterbi decoder

This is shown diagrammatically in figure 5.5.

The mapping of the 3-D decoder to this 2-D case has the obvious advantage that it requires no modifications to standard recognisers to allow them to work in these complex-noise environments[5]. However, there is a small loss of information at the model boundaries since the effective self-loop transition probabilities for the noise state cannot be preserved exactly. In practice this seems to make little difference.

A second alternative to increasing the number of states is to replace the multi-state ergodic noise HMM by a multiple-component single-state HMM. This discards all the temporal information in the noise. However, it does not require the number of states in the system to be increased. Furthermore, if the DPMC scheme is used to compensate the models, it is not necessary to increase the total number of components in the system. Of course, if the number of components in the system is not increased, there may be some loss of performance, since the number of modes in the corrupted-speech distribution may be greater than the number of components available.

5.10 Practical Implementations

In many practical systems full covariance matrices are not used to model the clean speech, nor are they desired in the corrupted-speech models for computational, and possibly memory limitation, reasons. Additionally, the Cepstral parameters are often truncated, particularly for more complex tasks, such as speaker independent systems. These practical

[5]Provided, of course, that the left-to-right topology is not "hardwired" in the recogniser.

implementation aspects are discussed in this section.

5.10.1 Speech and Noise Model Covariance Approximations

So far, nothing has been discussed about the actual statistics of the speech and noise that are stored. All the mismatch functions described, other than for the convolutional noise case, may be expressed in terms of 5 variables for each of the noise and speech sources. For the speech models these are

$$\mathbf{V}^c(\tau) = \left[\begin{array}{ccccc} \mathbf{S}^c(\tau)^T & \mathbf{\Delta S}^c(\tau)^T & \mathbf{\Delta}^2\mathbf{S}^c(\tau)^T & \mathbf{S}^c(\tau-w)^T & \mathbf{\Delta S}^c(\tau-w)^T \end{array}\right]^T \tag{5.59}$$

The first three are the standard elements stored in a HMM system. The remaining two are additional elements required for the dynamic coefficient compensation schemes. An important question is what correlations and statistics are required to achieve good compensation performance. The descriptions given in this section concentrate on schemes where static, delta and delta-delta parameters are required to be compensated. The number of Cepstral parameters for each of the static, delta and delta-delta parameters is assumed to be n.

The best approach is to use a full covariance matrix. Where this is the case, it will be referred to as the *Full* covariance approximation. To model all the correlations requires large amounts of memory, particularly with the additional parameters required for dynamic coefficient compensation. In medium to large vocabulary systems, this is not always feasible.

An alternative approximation is to assume that the DCT transformation sufficiently decorrelates the elements of the static coefficients so that the diagonal covariance approximation yields enough information for compensation. The covariance matrix could then be represented as

$$\mathrm{Covar}(\mathbf{V}^{\mathbf{r}c}(\tau), \mathbf{V}^{\mathbf{r}c}(\tau)) \approx \left[\begin{array}{cccc} \mathbf{\Sigma}_1 & \mathbf{0}^5 & \cdots & \mathbf{0}^5 \\ \mathbf{0}^5 & \mathbf{\Sigma}_2 & \cdots & \mathbf{0}^5 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}^5 & \mathbf{0}^5 & \cdots & \mathbf{\Sigma}_n \end{array}\right] \tag{5.60}$$

where

$$\mathbf{V}^{\mathbf{r}c}(\tau) = \left[\begin{array}{ccc} \mathbf{K}_1^c(\tau)^T & \ldots & \mathbf{K}_n^c(\tau)^T \end{array}\right]^T \tag{5.61}$$

$$\mathbf{\Sigma}_i = \mathrm{Covar}(\mathbf{K}_i^c(\tau), \mathbf{K}_i^c(\tau)) \tag{5.62}$$

$$\mathbf{K}_i^c(\tau) = \left[\begin{array}{ccccc} S_i^c(\tau) & \Delta S_i^c(\tau) & \Delta^2 S_i^c(\tau) & S_i{}^c(\tau-w) & \Delta S_i^c(\tau-w) \end{array}\right]^T \tag{5.63}$$

and the operation

$$\mathrm{Covar}(\mathbf{x},\mathbf{x}) = \mathcal{E}\left\{\mathbf{x}\mathbf{x}^T\right\} - \mathcal{E}\left\{\mathbf{x}\right\}\mathcal{E}\left\{\mathbf{x}\right\}^T \tag{5.64}$$

To store the correlations no longer requires a $5n \times 5n$ covariance matrix, but n sets of 5×5 covariance matrices. This is normally feasible to store, even for large model sets. This will be referred to as the *Extended* covariance approximation.

If not all these correlations are stored, thus reducing the memory requirements, it is possible, by assuming that

$$\mathrm{Covar}(\mathbf{S}^c(\tau), \mathbf{S}^c(\tau - w)) \approx \mathbf{0}^n \tag{5.65}$$

to obtain estimates for all the correlations. Thus[6]

$$\mathrm{Covar}(\mathbf{V}^c(\tau), \mathbf{V}^c(\tau)) \approx \begin{bmatrix} \boldsymbol{\Sigma}^c & \mathbf{0}^n & -2\boldsymbol{\Sigma}^c & \mathbf{0}^n & \boldsymbol{\Sigma}^c \\ \mathbf{0}^n & \boldsymbol{\Delta\Sigma}^c & \mathbf{0}^n & -\boldsymbol{\Sigma}'^c & \mathbf{0}^n \\ -2\boldsymbol{\Sigma}^c & \mathbf{0}^n & \boldsymbol{\Delta}^2\boldsymbol{\Sigma}^c & \mathbf{0}^n & -\boldsymbol{\Sigma}^c - \boldsymbol{\Delta\Sigma}'^c \\ \mathbf{0}^n & -\boldsymbol{\Sigma}'^c & \mathbf{0}^n & \boldsymbol{\Sigma}'^c & \mathbf{0}^n \\ \boldsymbol{\Sigma}^c & \mathbf{0}^n & -\boldsymbol{\Sigma}^c - \boldsymbol{\Delta\Sigma}'^c & \mathbf{0}^n & \boldsymbol{\Delta\Sigma}'^c \end{bmatrix} \tag{5.66}$$

where $\boldsymbol{\Sigma}'^c = \mathrm{Covar}(\mathbf{S}^c(\tau - w), \mathbf{S}^c(\tau - w))$. Unfortunately this matrix is not guaranteed to be positive definite, as required for a covariance matrix. To overcome this problem the off-diagonal terms, the approximate elements, may be scaled until the matrix becomes positive definite. This will be referred to as the *Pseudo* covariance approximation.

Alternatively, a simple diagonal covariance approximation may be used such that the covariance matrix $\mathrm{Covar}(\mathbf{V}^c(\tau), \mathbf{V}^c(\tau))$ is assumed to be diagonal. This form of covariance assumption will be referred to as the *Diagonal* covariance approximation.

A further simplification may be made by assuming that the statistics at time $\tau - w$ are the same as those at τ. Differences in these statistics come from a variety of sources. In multiple component systems, the multiple components partially model the trajectories of the Cepstral parameters. Thus, the mixture component occupancy may not be the same at time τ as at time $\tau - w$. Additionally, if the state occupancy is for a short duration then variations in the surrounding states will result in dramatically altered statistics. Ignoring these differences, however, it is possible to use the standard model set with no extensions to the compensation process. This will be referred to as the *Standard Diagonal* covariance approximation, since only the standard HMM parameters are required. Where this approximation is combined with a full covariance matrix, it will be referred to as the *Standard Full* covariance approximation.

An important measure, in terms of storage capacity, is the number of parameters required per-Gaussian distribution. This is shown for each of the modelling cases in table 5.1. In the table, $\mathbf{O}$ indicates the static parameters, $\boldsymbol{\Delta}\mathbf{O}$ and $\boldsymbol{\Delta}^2\mathbf{O}$ the delta and delta-delta parameters respectively.

5.10.2 Corrupted-Speech Model Covariance Approximation

The compensation procedures described assume that full covariance matrices are used for the corrupted-speech distributions. In practice it is common to assume that observation vector elements are independent, so that diagonal covariance matrices may be used. This reduces the run-time computational requirements. PMC, on the other hand, will yield a full covariance matrix, even if diagonal covariance matrices are used in the initial speech and noise models.

In order to avoid the run time computational overhead associated with the use of full covariance matrices and the large memory requirements for the model sets, the full

[6] If statistics exist for the correlation then the exact statistics are used. For example

$$\mathrm{Covar}(\boldsymbol{\Delta}^2\mathbf{S}^c(\tau), \boldsymbol{\Delta}\mathbf{S}^c(\tau - w)) = -\boldsymbol{\Sigma}^c - \boldsymbol{\Delta\Sigma}'^c$$

Covariance Approximation	Number of Parameters Needed to Compensate: $\mathbf{O}$	$\mathbf{O}$, $\Delta\mathbf{O}$	$\mathbf{O}$, $\Delta\mathbf{O}$ $\Delta^2\mathbf{O}$
Full	$n((n+1)/2+1)$	$3n((3n+1)/2+1)$	$5n((5n+1)/2+1)$
Extended	$2n$	$9n$	$20n$
Pseudo	$2n$	$6n$	$10n$
Diagonal	$2n$	$6n$	$10n$
Standard Full	$n((n+1)/2+1)$	$2n((2n+1)/2+1)$	$3n((3n+1)/2+1)$
Standard Pseudo	$2n$	$4n$	$6n$
Standard Diagonal	$2n$	$4n$	$6n$

Table 5.1: Number of parameters per-mixture component for various covariance approximations

covariance matrices can be made diagonal by simply setting the off-diagonal terms to zero. This is the approach adopted throughout this work, even when off-diagonal terms are modelled in the clean speech and noise models. For the case of static coefficients, little gain was obtained using full covariance matrices [25].

5.10.3 Cepstral Smoothing

In most large vocabulary speech recognition systems, only a subset of the Cepstral parameters are used. The feature vector is normally truncated, for example 24 Log-Spectral parameters will be mapped to the 13 lower-order Cepstral parameters. The unused truncated parameters are assumed to contain minimal discriminatory information and thus constitute an additional computational overhead at no gain. If truncated feature vectors are used in the clean speech and noise models, it is not possible to perform an inverse DCT to obtain the "correct" Log-Spectral domain distributions, as information has been lost. To get around this problem the means and variances are zero padded to the required number to allow the inverse DCT to be achieved. After model combining the mean and variances, which due to the non-linearity of the log compression will have full dimensionality, are truncated back to the original number.

Alternative approaches are possible. Where memory constraints allow, a full Cepstral vector model set may be generated in a single-pass fashion. This may then be used in the compensation process and the feature vector only truncated after the compensation process. Although this is more accurate, there is an overhead in terms of the number of parameters.

5.11 Training Models on Noise-Corrupted Data

The aim of model-based compensation schemes is to generate a set of corrupted speech models matched to the new noise environment. In order to compute an upper limit on the performance that can be achieved using this approach, models can be trained directly on noise-corrupted data. This noise-corrupted data may either be obtained from a stereo database, or samples of the interfering noise may be added to the clean training database. A new set of models can then be estimated from this "matched" database. The new models may be estimated in one of two ways, either using standard HMM training or in a single-pass, non-iterative, fashion.

5.11.1 Multi-Pass Training on the Noise-Corrupted Database

Given the corrupted training data, the noise-corrupted speech models may be generated in the standard way, for example using the techniques described in the HTK manual [102]. This will result in a set of noise-corrupted speech models. There will be some important differences between this model set and the clean set other than, of course, one models noise-corrupted speech and the other models clean speech. If any form of triphone tying is used in the system, as occurs in most medium to large vocabulary systems, the clustering, if performed on the noise-corrupted data, will naturally yield very different cluster sets. In particular, all low energy states will be mapped to the same state, as they will be masked by the interfering noise. Additionally during the training the frame/state alignments may be poor, as the noise will reduce the variations between the states. The estimated parameters may, therefore, not truly reflect a distinct region of speech. This form of training will be referred to as *multi-pass* training. It has previously been performed for a variety of noise conditions [15].

5.11.2 Single-Pass Training on the Noise-Corrupted Database

As a clean version of the noise-corrupted training data is available, it is possible to use single-pass training to obtain the noise-corrupted models. In single-pass training the clean model set is used to obtain the frame/state component alignment, $L^s_{jm}(\tau)$, on the clean speech database and observations, $\mathbf{O}^c(\tau)$, are taken from the noise-corrupted database. The re-estimation formulae of the standard HMM training may be rewritten as

$$\hat{\mu}_{jm} = \frac{\sum_{\tau=1}^{T} L^s_{jm}(\tau)\mathbf{O}^c(\tau)}{\sum_{\tau=1}^{T} L^s_{jm}(\tau)} \tag{5.67}$$

and

$$\hat{\mathbf{\Sigma}}_{jm} = \frac{\sum_{\tau=1}^{T} L^s_{jm}(\tau)(\mathbf{O}^c(\tau) - \hat{\mu}_{jm})(\mathbf{O}^c(\tau) - \hat{\mu}_{jm})^T}{\sum_{\tau=1}^{T} L^s_{jm}(\tau)} \tag{5.68}$$

where

$$L^s_{jm}(\tau) = p(q_{jm}(\tau)|\mathbf{S}^c_T, \mathcal{M}) \tag{5.69}$$

and

$$\mathbf{S}^c_T = \mathbf{S}^c(1)\mathbf{S}^c(2)\ldots\mathbf{S}^c(T) \tag{5.70}$$

A set of models may then be generated in a single pass. Moreover by using the frame/state component alignment associated with the clean model set and clean data, the frame/state component alignment of the training data is not altered. Thus the models obtained are the same as a "perfect" non-iterative model-based compensation scheme. This, of course, assumes that a stationary noise condition is being used. This style of training will be referred to as *single-pass* training. Having performed a single pass to obtain a reasonable set of models, it is possible to refine this set of models using standard HMM training.

In addition to training noise-corrupted speech models, it is possible to use single-pass training to alter the model parameter set. Provided that the complete dataset is not dramatically altered, the new single-pass trained model set will be a good estimate. For example, a set of models may be built using only static parameters. It is possible to append dynamic model parameters to these static parameters in a single pass. An additional smoothing pass may be incorporated to further improve this estimate.

5.12 Performance Criteria

There are two questions that need to be answered in assessing the performance of the compensation schemes described here. Firstly, how "close" is the estimated model set to those of the "best", single-pass trained, system? This assesses how good the approximations are in the compensation process. Secondly, given the crude nature of the modelling of the corrupted-speech distribution with a single multi-variate Gaussian distribution, or multiple mixture components, is this approximation good enough to achieve acceptable noise robustness and can it be improved by better modelling the corrupted-speech distribution?

5.12.1 Distance Measures between Model Sets

A variety of distance measures have previously been proposed for determining how "close" two HMMs are to one another [42]. These measures are either based on Euclidean distance or the Kullback-Leibler (KL) number between two models for discrete HMMs. This could be applied to CDHMMs, however, for this work, where the target HMMs are obtained using single-pass training, or non-iterative PMC, so the transition probabilities and component weights are unaltered, it is preferable to use a measure dependent only on the parameters being modified, in this case the means and variances. In addition, it is desirable to be able to examine each of the Cepstral features independently to assess the performance of the various schemes at a finer level. To this end, instead of considering the KL number between complete models, the KL number is averaged between pairs of Gaussians on a per-component-element level. An additional reason for using a criterion other than word error rate is that for a small test set (such as that provided in the NOISEX-92 database) where differences in recognition performance between the various schemes is small, the compensation schemes may be assessed in terms of how well they map a set of clean models to the matched corrupted-speech models.

The KL number is defined as

$$D_{KL}(p, q) = \mathcal{E}\left\{\log\left(\frac{p(O)}{q(O)}\right)\right\} \tag{5.71}$$

where $p(O)$ is the "true" distribution, in this case from the single-pass noise trained model set, and $q(O)$ is the estimated distribution. The KL number is greater than or equal to zero, with equality when the two distributions are identical. For Gaussian distributed variables the above expression has a closed form

$$D_{KL}(p, q) = \frac{1}{2}\left[\log\left(\frac{\sigma_q^2}{\sigma_p^2}\right) + \frac{(\mu_p - \mu_q)^2}{\sigma_q^2} + \left(\frac{\sigma_p^2}{\sigma_q^2} - 1\right)\right] \tag{5.72}$$

where μ_p and μ_q are the means of the true and estimated distributions, and σ_p^2 and σ_q^2 the respective variances. Since many Gaussian distributions are associated with each set of models, the average KL number is considered. Thus the distance measure used is

$$\overline{D}_{KL}(\mathcal{M}, \hat{\mathcal{M}}) = \frac{1}{\left(\sum\limits_{p=1}^{P} M(p)\right)} \sum_{p=1}^{P} \sum_{m=1}^{M(p)} D_{KL}(\mathcal{M}_p(m), \hat{\mathcal{M}}_p(m)) \tag{5.73}$$

where $\mathcal{M}_p(m)$ indicates mixture component M_m of the p^{th} model of the model set $\mathcal{M}$ and $M(p)$ is the total number of mixture components in that model. Note that no account is taken of the mixture weights or the transition probabilities. In the non-iterative PMC schemes these are assumed to be unaltered, as the frame/state alignment is not altered by the addition of noise.

The average KL number is not considered for the iterative PMC schemes as the frame state component alignment of the data is unknown. Additionally, the distance measure defined here is only applicable on a per-Cepstral-feature level to models with diagonal covariance matrices. The average KL number could be used on a per-mixture-component level for full covariance matrices.

5.12.2 Percentage Word Accuracy

For continuous speech recognition performance figures are usually quoted in terms of % word accuracy, or error rate. For N tokens, S substitution errors, D deletion errors and I insertion errors,

$$\% \text{ accuracy} = [(N - S - D - I)/N] \times 100\% \tag{5.74}$$

The error counts themselves are calculated by using a DP string matching algorithm between the recognised digit sequence and the reference transcription.

The word error rate is defined as $(100 - \%\text{ accuracy})$ and is the form used throughout this work.

Chapter 6

Interfering Noise Sources Considered

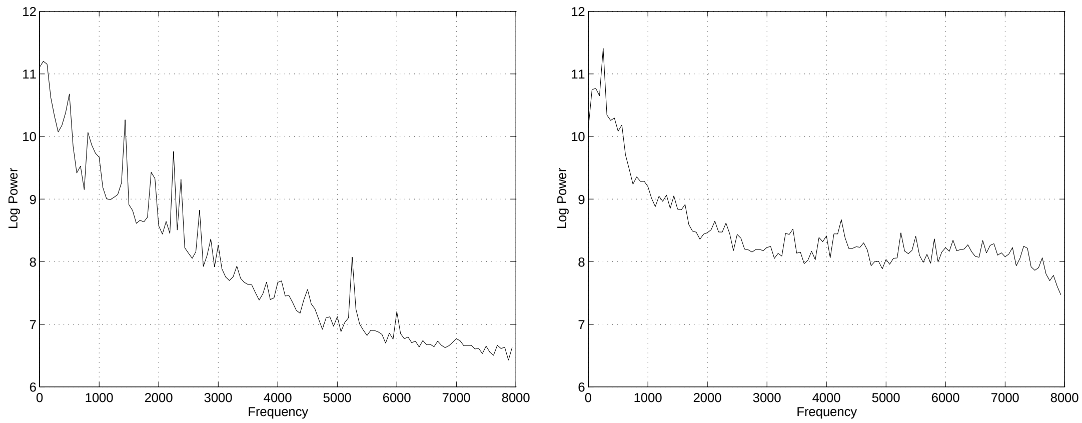
The work presented in this thesis concentrates solely on data where the noise has been artificially added to the clean speech. The majority of the noise sources considered were taken from the NOISEX-92 database [91]. A variety of noises are available on this database, ranging from fairly stationary, Lynx Helicopter noise, to some with temporal variability, Operations Room noise, to very distinct multi-state noise sources, the Machine Gun noise. The second set of additive noises considered were a variety of car noises supplied with the ARPA CSRNAB 1994 Spoke 10 development and evaluation data.

For the NOISEX-92 results the test data, with the noise added already at various signal-to-noise ratios, was supplied with the database. All the Resource Management experiments required noise to be added. Details of the attenuation on the noise sources are given in the relevant sections. Finally, the ARPA test data was distributed with noise already added by NIST.

6.1 Noises from the NOISEX-92 Database

The log power spectral densities of the approximately stationary noise conditions, the Lynx Helicopter noise and the Operations Room noise, were measured. Figure 6.1(a) shows the log power spectral density of the Lynx Helicopter noise. This contains a very significant low frequency component, which tails off rapidly. The log power spectral density of the Operations Room noise is also shown in figure 6.1(b). Here, there is again a significant low frequency noise component. However the spectra does not tail off as rapidly as the helicopter noise. In addition, this noise source has a far greater temporal variability.

The log power spectral density plots do not adequately describe the effects of the noise on the recognition system. For example, if a high pass filter is initially applied to the data as part of the parameterisation, then the large low frequency component of the Lynx Helicopter noise will not affect the system's performance. To gain an insight into the actual effects of the noise on the recognition system, single Gaussian mixture component single state noise models were built on the data, parameterised as in the NOISEX-92 experiments described in the next chapter. The means of these models then underwent an inverse DCT to convert them from the Cepstral to the Log-Spectral domain. Figure 6.2 shows the Log-Spectral plot against mel-spaced frequency bin for two noises, Lynx Helicopter and Operations Room. The pre-emphasis applied to the data, in conjunction with the mel-



(a) Lynx Helicopter noise (b) Operations Room noise

Figure 6.1: Log power spectral density against frequency for Lynx Helicopter noise and Operations Room noise taken from the NOISEX-92 database

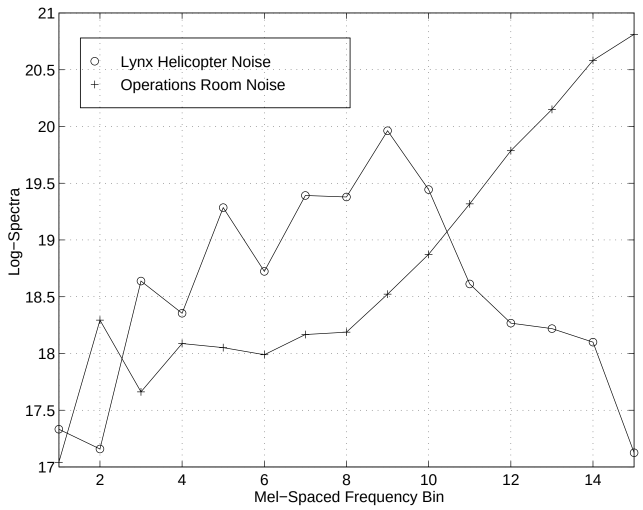


Figure 6.2: Log-Spectral values against mel-spaced frequency bin for noises from the NOISEX-92 database

spaced frequency bins, has dramatically reduced the low-frequency component of both noises.

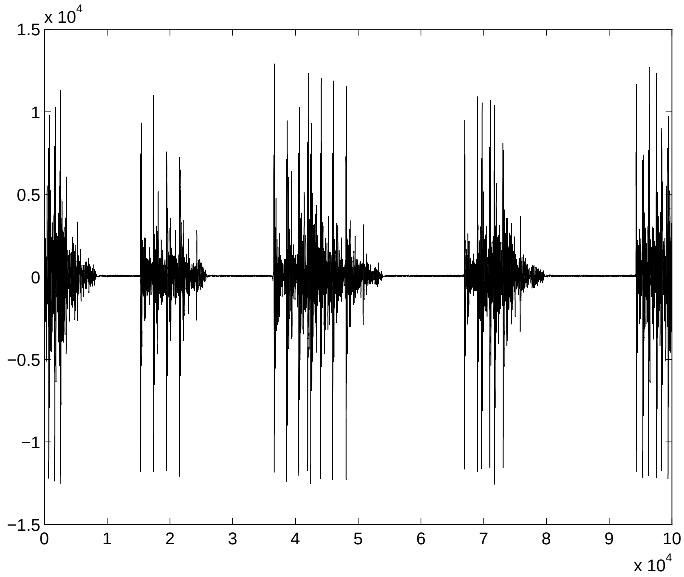


Figure 6.3: Time domain plot of the Machine Gun noise taken from the NOISEX-92 database

The NOISEX-92 database also contains a distinctly non-stationary noise source, the Machine Gun noise. A time domain plot of this noise is shown in figure 6.3, where the "silence" and "bang" periods are clearly visible. Two possible noise models associated

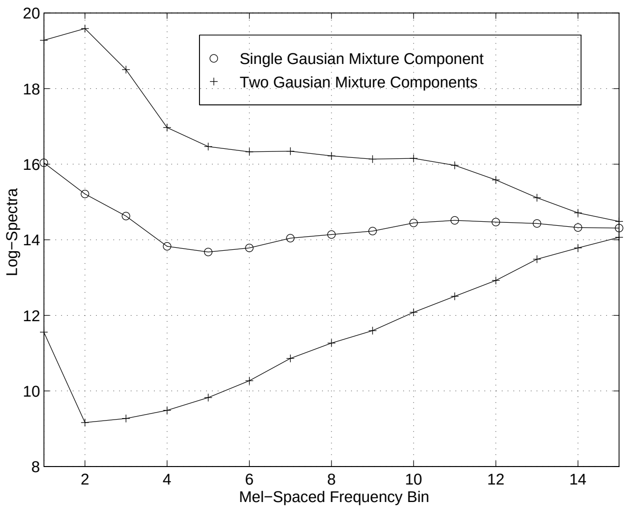


Figure 6.4: Log-Spectral value against mel-spaced frequency bin for a single and two Gaussian mixture component models from the NOISEX-92 database

with the machine gun noise are illustrated in figure 6.4. A single-component noise model is not a good representation for this noise and, as can be seen, increasing the number of

Gaussian components shows its distinct multi-state nature.

6.2 Noises from the ARPA 1994 Evaluation

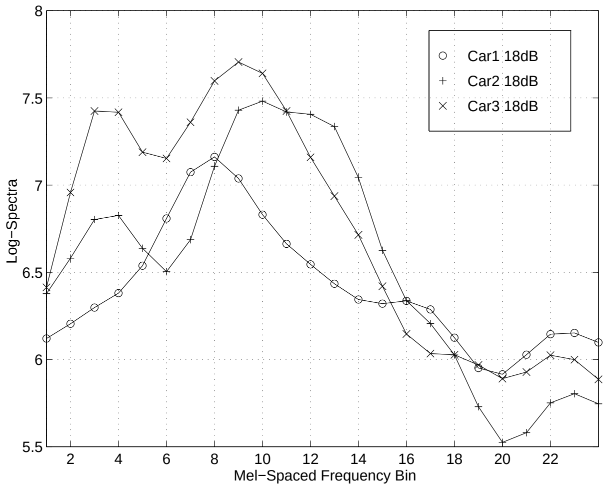


Figure 6.5: Log-Spectral value against mel-spaced frequency bin for the three ARPA development car noises at 18dB

Three different car noise sources were supplied with the ARPA 1994 development data. These were labelled Car 1, Car 2 and Car 3 and were supplied at 5 different noise levels, labelled 6dB, 12dB, 18dB, 24dB and 30dB. The Log-Spectral domain plots of the three car noises, as modelled by a single Gaussian HMM using the WSJ speech parameters described in Chapter 9, are shown in figure 6.5. All the noise sources show the same general pattern with a rapid tail off at high mel-spaced frequency bin values.

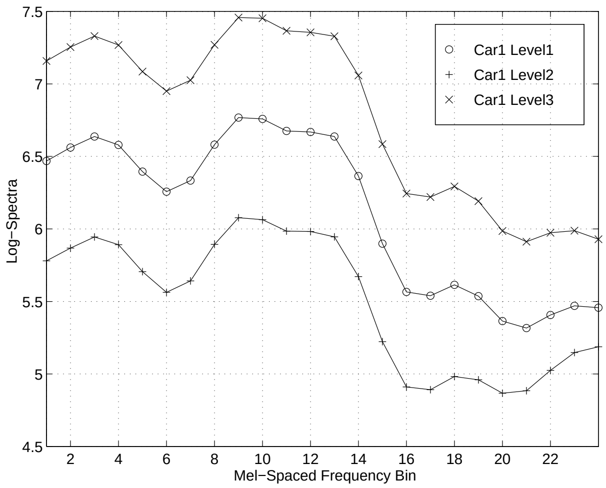


Figure 6.6: Log-Spectral value against mel-spaced frequency bin for the three ARPA evaluation car noise levels

For the evaluation a single car noise was supplied at 3 noise levels, labelled Level

1, Level 2 and Level 3. The model Log-Spectral plots for each of the levels are shown in figure 6.6. Again there is a rapid tail off at high mel-spaced frequency bin values. Comparing figure 6.6 with figure 6.5 indicates that the 18dB noise condition is similar to the Level 3 noise for the evaluation data.

Chapter 7

Evaluation on NOISEX-92

Initial evaluation of PMC as a technique for robustness to additive and convolutional noise was performed using a small vocabulary database in extreme noise conditions, the NOISEX-92 database [91]. NOISEX-92 is a publicly available database comprising of digits spoken by two speakers, one male, one female. Two test sets are available for each speaker. The first comprises the individual digits spoken in a random sequence with approximately one second gaps between them. In the second test, the digits are spoken in triplets with a one second gap between each triplet. Clean training data is supplied for both speakers. The noise was artificially added at a variety of SNRs ranging from -6dB to 18dB, both the corrupted and clean test data are supplied. For the convolutional noise experiments, spectral tilt having a flat frequency response up to a break point frequency of 250Hz followed by a +3dB/octave rise above 250Hz was added to the training data. For clarity, results quoted here are limited to the case of male isolated digits combined with Lynx Helicopter noise. Results for other noise sources have been published [27] and the general trends are consistent across all the noises tested.

7.1 The NOISEX-92 System

The data was preprocessed using a 25 msec Hamming window and a 10 msec frame period. For each frame a set of 15 MFCCs were computed. The zeroth Cepstral coefficient was computed and stored since it is needed in the PMC mapping process.

For each digit, a single Gaussian CDHMM with 8 emitting states was trained using the clean data only. The topology for all models was left-to-right with no skips and diagonal covariance matrices were used throughout. For each test condition, a single-state diagonal covariance noise HMM was trained using the "silence" intervals of the test files. 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 classification errors. All training and testing used version 1.4 of the portable HTK HMM toolkit [98] with suitable extensions to perform PMC. The Numerical Integration technique was used for the implementation of PMC unless otherwise stated. For the numerical integration ten points were used in each dimension. All results quoted are in terms of percentage word error rate.

7.2 Results on NOISEX-92 with Additive Noise

Model Set	SNR (dB)		
	-6	+0	+6
Clean	90	83	51
Single-Pass	46	4	0
Multi-Pass	54	10	1

Table 7.1: Word error rates (%) of the baseline and matched model sets: NOISEX-92 Lynx Helicopter additive noise only

Table 7.1 shows the baseline performance of the recogniser with no compensation compared to models trained on noise-corrupted data, using both single-pass and multi-pass training. Only the three noisiest conditions were considered, since most of the compensation techniques recorded no errors at higher SNRs. The addition of noise seriously degraded the performance, reducing the recognition rate from zero errors with no noise to almost random performance at the -6dB level for the clean speech models. Both matched systems performed better than the uncompensated clean system. It is interesting to note that the multi-pass trained system performed slightly worse than the single-pass system. This was observed in the other noise conditions of the NOISEX-92 database. The probable reason for this drop in performance is the poor frame/state alignment obtained when training at low SNRs.

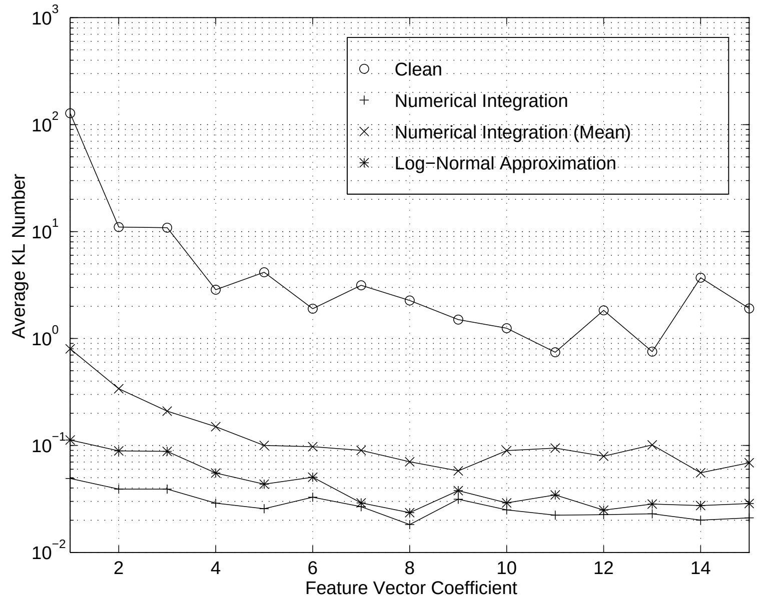


Figure 7.1: Average KL number for clean, Log-Normal compensated, mean-and-variance-compensated, and mean-compensated using the Numerical Integration approximation and Diagonal covariance approximation model sets on Lynx Helicopter noise at -6dB

The average KL number, as given by equation 5.73, between the single pass model set trained on noise-corrupted data at -6dB and various PMC approximations are shown in figure 7.1. Feature vector coefficients 1 to 15 represent C_0 to C_{14} respectively. Examining the clean model set average KL number, the low-order Cepstra were more affected by the addition of noise, with the first feature vector component, C_0, having the highest average KL number. The average KL number for both the mean-compensated and mean-and-variance-compensated model sets was far lower than for the uncompensated system. Compensating the variances of the model set further reduced the average KL number. This indicates that the noise altered both the means and the variances of the system. In addition to the Numerical Integration implementation of PMC, a Log-Normal approximation has been described, where both the means and variances are compensated. The average KL number of this model set is also illustrated in figure 7.1. As can be seen, the use of the Log-Normal approximation increased the average KL number of the system compared to the Numerical Integration approximation, but it was still lower than both the uncompensated system and only compensating the means.

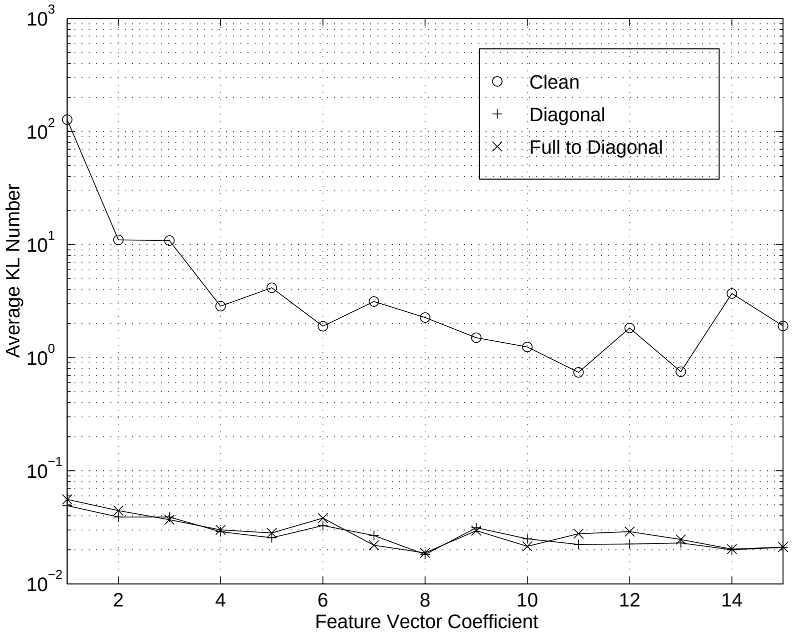


Figure 7.2: Average KL number for clean, Diagonal and Full covariance approximations on Lynx Helicopter noise at -6dB

A normal assumption made in a HMM system is that the elements of the feature vector are independent, this allows diagonal covariance matrices to be used and dramatically reduces the number of parameters in the system. However, this assumption is only partially valid for Cepstral parameters and may reduce the effectiveness of the compensation scheme. A full covariance HMM system was built using the same complete dataset as the diagonal covariance case. This model set was then compensated using a full covariance noise model. The off-diagonal terms of the corrupted-speech covariance matrix were then set to zero and the average KL number measured. The average KL number is shown in figure 7.2, labelled *Full to Diagonal*. The use of the full covariance matrix, at least in terms of the average KL number of the compensated system, gained little over diagonal covariance matrices[1]. Diagonal covariance matrices were therefore used in all the compensation schemes.

Model Set	SNR (dB) -6	+0	+6
Numerical Integration	33	2	1
Numerical Integration (Mean)	56	8	1
Log-Normal	42	6	1

Table 7.2: Word error rates (%) of the PMC-compensated model sets: NOISEX-92 Lynx Helicopter additive noise only

The word error rates for three different compensation schemes are shown in table 7.2. All three systems showed improved noise robustness compared to the clean system. The performance of the Numerical Integration with mean-and-variance compensation, *Numerical Integration*, was comparable with the single-pass matched system. Its performance at low SNR was better than the matched system, however on such a small database it is hard to draw conclusions from this result. The Numerical Integration technique was also used to modify just the means of the models, *Numerical Integration (Mean)*. This result should be approximately the same as that of hypothesised Wiener filtering [25]. Good performance was observed down to 0dB, although slightly degraded with respect to the variance compensated system. The Log-Normal approximation, *Log-Normal*, yielded performance between the two Numerical Integration systems, with a greatly reduced computational load.

7.3 Results on NOISEX-92 with Additive and Convolutional Noise

Model Set	SNR (dB) -6	+0	+6
Clean	90	76	58
Numerical Integration	61	52	28
CMN	83	82	56

Table 7.3: Word error rates (%) of the baseline and standard PMC-compensated model sets: NOISEX-92 Lynx Helicopter additive and convolutional noise

Spectral tilt in the form of a +3dB/octave rise above 250Hz was applied to the clean speech training data and the clean models retrained. The test data was identical to the data in the previous section, allowing direct comparison to the results described therein. Table 7.3 shows the performance of the new clean models with no compensation, *Clean*, and the standard PMC Numerical Integration, *Numerical Integration*, which assumed there was no spectral tilt present. Not surprisingly, the clean models performed poorly again.

[1]This was also reflected in recognition performance.

The standard PMC system with no tilt compensation also performed poorly, though not as badly as the clean system. This illustrates the degradation in performance that results from spectral tilt being present in the system. In addition, table 7.3 shows the performance of Cepstral Mean Normalisation (CMN), a standard technique used to make recognition systems robust to convolutional noise. However, due to the high levels of additive noise, CMN did not improve the performance in this case.

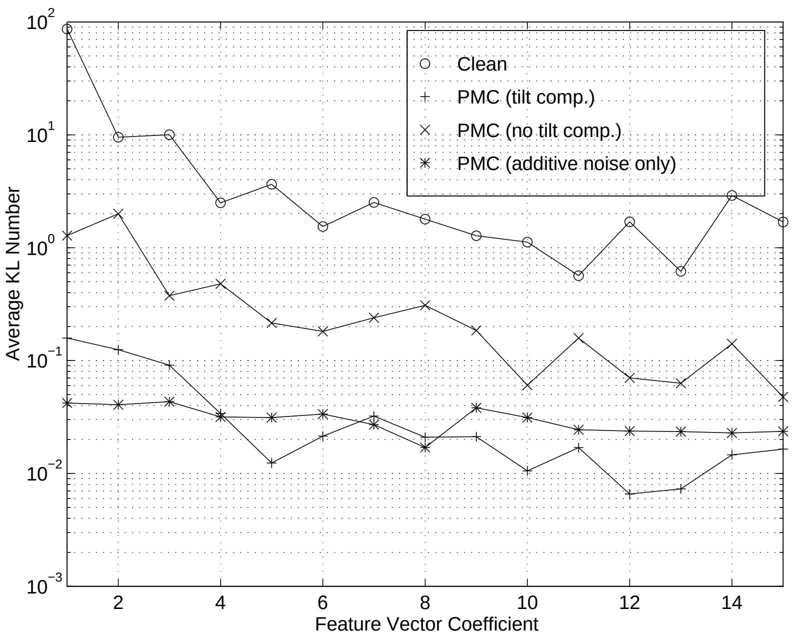


Figure 7.3: Average KL number for the PMC Numerical Integration approximation compensated models with and without convolutional noise compensation on Lynx Helicopter noise at 0dB

Since the target models in the new acoustic environment were identical to those in section 7.2, it was possible to measure the average KL number between the estimated models and single-pass trained models. Figure 7.3 shows the average KL number for PMC with no tilt compensation, PMC with tilt compensation and, for comparison, PMC in additive noise only for the 0dB Lynx Helicopter noise condition. The tilt compensation decreased the average KL number compared to the no tilt compensation case. Here, the tilt compensation was estimated using a 30-component general speech model and the first 20 digits of the test speech. Comparison of the tilt compensated case with the additive noise case shows, particularly for the lower Cepstra, that the tilt estimation had not perfectly compensated for the convolutional noise, but was better than the no tilt compensation case.

Of course, the average KL number is only an indicator of how well the system will actually perform. Table 7.4 shows the word error rate as a function of the number of components in the general speech model using the first 20 test digits as channel adaptation data. The -6dB case yielded little information about the tilt compensation. At this noise level, the estimate of the spectral tilt was poor as little of the speech exceeded the noise level, which is necessary to allow good estimates of the convolutional noise to be made. For the two higher SNR conditions the use of 5 components in the general model yielded good

Number Components	SNR (dB)		
	-6	+0	+6
1	55	20	5
2	52	8	1
5	50	5	0
10	55	5	0
30	57	5	0

Table 7.4: Word error rates (%) against number of general speech model components used for estimating the convolutional noise: NOISEX-92 Lynx Helicopter additive and convolutional noise

performance, comparable with that of the no spectral tilt case described in the previous section. The number of components in the general model is important as it impacts on the computational overhead of estimating the spectral tilt.

Simple Adaptation	Number Digits	SNR (dB)		
		-6	+0	+6
$\times$	1	75	62	73
$\surd$	1	68	56	44
$\times$	2	59	25	9
$\surd$	2	59	11	1
$\times$	5	53	2	1
$\surd$	5	49	2	1
$\times$	20	57	5	0
$\surd$	20	59	5	0

Table 7.5: Word error rates (%) against number of digits in the tilt estimation data: NOISEX-92 Lynx Helicopter additive and convolutional noise

It is preferable to start the recognition process as rapidly as possible. To this end the convolutional noise needs to be estimated on as few digits as possible. However, this may result in a large mismatch between the general speech model training data and the speech actually present in the test adaptation data. To partially overcome this the simple adaptation of the general model weights as described in section 5.8, labelled *Simple Adaptation*, was tested. For this work 30 components were used in the general speech model. Table 7.5 illustrates the variation of the performance as the number of digits used to estimate the spectral tilt is reduced with and without adaptation[2]. Again, poor performance was observed in the -6dB case. For the two higher SNR conditions, the use of the first 5 digits to estimate the tilt yielded good performance. Below this, the performance dropped. This degradation in performance was less dramatic when the simple adaptation scheme, described in section 5.8, was used to modify the general speech model weights.

[2]The first N test digits were taken in each case.

7.4 Discussion

This chapter has presented results on the evaluation of PMC using a small vocabulary system in low SNRs and in situations where convolutional noise may be present. Two questions were posed. The first, whether the PMC scheme modified the HMMs to be "close" to a matched system, was investigated using a model distance metric based on the average KL number between Gaussian components. All the compensation schemes reduced the average KL number, the lowest being achieved by the Numerical Integration approximation compensating both means and variances. The second question was that of whether using single multi-variate Gaussians to model the corrupted-speech distribution gave good noise robustness. The clean system showed poor robustness to noise, at 0dB the performance had degraded from no errors in the clean environment to 83% word error rate. Training a matched system, either in a single-pass fashion or a multi-pass fashion, improved the performance to under 10% word error rate at 0dB. All the PMC schemes dramatically improved the performance. In the case of the Numerical Integration approximation the performance was comparable to the single-pass trained system, the best matched system.

In the presence of additive and convolutional noise, it was found that, provided at least 5 digits were available to estimate the spectral tilt, the performance was comparable to that of the matched system. If fewer digits were available a simple adaptation scheme reduced the degradation in performance.

Chapter 8

Evaluation on a Noise-Corrupted Resource Management Task

The previous chapter describes the performance of PMC when compensating the static parameters on a small vocabulary speech recognition task in extreme SNRs. To evaluate the performance of the dynamic parameter compensation schemes it was necessary to move to a larger database. For this task the DARPA Resource Management (RM) database was chosen. To allow the results described in this chapter to be easily duplicated, the noises selected were taken from the NOISEX-92 database and the scaling factors used to generate the test data, in addition to the SNRs, are given.

8.1 The Resource Management System

The database used for the clean speech models was the RM database [77]. This is a 1000 word task with a vocabulary based on a naval resource management domain. There are 3990 training sentences and a set of four 300 sentence test sets, of which only three were considered in this work, the February 1989, October 1989 and February 1991 DARPA evaluation sets[1]. The recordings were made in a sound isolated recording booth, yielding a SNR of $>$ 40dB. For all tests performed in this chapter a word pair grammar, perplexity[2] 60, was used.

Noise sources were taken from the NOISEX-92 database [91]. The Lynx Helicopter (stationary), Operations Room (some temporal variability), and Machine Gun (distinctly multiple state) noises were used. These are described in Chapter 6.

The Lynx helicopter noise was added to the clean RM data to give a SNR of approximately 18dB[3]. To achieve this SNR, the noise was attenuated by 20dB. The exact SNRs for the three test sets, and the "clean" SNRs, are shown in table 8.1. The mismatch between the clean and the noisy conditions was about 30dB. To obtain alternative SNRs the noise was scaled accordingly and added.

The baseline speech recognition system was built using the Resource Management Toolkit distributed with HTK Version 1.5 [102] and used for the November 1992 ARPA

[1] Only three of the four possible RM test sets were considered. The Sep'92 test set was not used. A variety of "spot checks" were performed with the Sep'92 and no difference in the trends described were observed.

[2] Perplexity is the average number of words which have to be hypothesised at any decision point.

[3] NIST "wavmd" code was used to determine the SNR.

Test Set	Clean SNR (dB)	Noisy SNR (dB)
Feb'89	48.9	18.2
Oct'89	48.7	18.4
Feb'91	48.7	18.6

Table 8.1: Average SNR for the RM test sets adding Lynx Helicopter noise attenuated by 20dB

Evaluation [96]. Two modifications to the parameterisation used for the evaluation were made. In order to use PMC to compensate a set of models, it is necessary to have the zeroth Cepstra to represent the energy whereas the standard HTK system uses a normalised log-energy. In addition to modifying the energy term, the method for calculating the delta and delta-delta parameters was altered. As previously mentioned it is not possible to "correctly" compensate delta parameters when they are calculated using linear regression. HTK was therefore extended to calculate deltas and delta-deltas using simple difference expressions.

The data was preprocessed using a 25 msec Hamming window and a 10 msec frame period. Additionally, the data was pre-emphasised with a factor of 0.97 and liftered with a factor of 22 using equation 2.42. 24 Log-Spectral parameters were used to generate the basic acoustic analysis. From this vector, an observation vector consisting of 12 MFCCs, appended by the zeroth Cepstra, was generated. In addition to the static parameters, delta and delta-delta parameters were added. These were calculated using

$$\boldsymbol{\Delta}\mathbf{O}^c(\tau) = \frac{1}{2w}\left(\mathbf{O}^c(\tau+w) - \mathbf{O}^c(\tau-w)\right) \tag{8.1}$$

$$\boldsymbol{\Delta}^2\mathbf{O}^c(\tau) = \frac{1}{2w_a}\left(\boldsymbol{\Delta}\mathbf{O}^c(\tau+w_a) - \boldsymbol{\Delta}\mathbf{O}^c(\tau-w_a)\right) \tag{8.2}$$

where in these experiments, $w = 2$ and $w_a = 2$. Since all the Cepstral parameters were liftered and the delta and delta-delta parameters scaled, it was necessary to inverse lifter and rescale to achieve the distributions required for compensation. In addition, Cepstral smoothing was applied. The 24 Log-Spectral values were mapped to 13 Cepstral parameters. This meant that when performing the inverse DCT it was necessary to zero pad the feature vector, as described in section 5.10.3. Similarly when mapping back it was necessary to truncate the Cepstral vector to 13 again.

For the section where spectral tilt was added to the observations, the experiments with convolutional noise, the normal pre-emphasis was suppressed when parameterising the test data. This caused the low frequency components of the corrupted speech to be amplified and the high frequency components approximately unaffected compared to the clean speech data used for training.

For this work the models were trained from "scratch" with the new speech parameters. Thus, the standard clustering and "mixing up" techniques described in [99] were used. This resulted in a state-clustered triphone system consisting of 1805 tied states forming 1959 distinct triphones.

All results quoted were generated using the standard clean parameter settings for the recogniser. Diagonal corrupted-speech covariance matrices were used for all results. All results are quoted in terms of percentage word error rate.

8.2 Results on the Clean RM Database

Energy Term	Dynamic Terms	Num. Mix.	Word Err
Log	Regr.		7.2
Cepstra	Regr.	1	6.8
Cepstra	Diff.		8.0
Log	Regr.		5.9
Cepstra	Regr.	2	5.7
Cepstra	Diff.		6.1

Table 8.2: Word error rates (%) on clean RM Feb'89 test set for various parameterisations

In order to use PMC to compensate the models it was necessary to alter the speech parameter set from that used in the standard HTK RM recogniser. To test the effect of this, a series of systems were built using a variety of speech parameters and tested on the February 1989 test set. The results are shown in table 8.2. The first column shows the type of energy term used, *Log* is normalised log-energy, *Cepstra* is unnormalised C_0. The second column is the form of dynamic coefficients, *Regr.* is linear regression based, and *Diff.* is simple differences. These apply to both the delta and delta-delta parameters.

There was a slight difference in performance when changing the parameter set, particularly for the single mixture component case. However these differences were greatly reduced in the two mixture component case. From these results, it appears that there is little difference in clean performance between using log-energy with linear regression as used in the standard system and the zeroth Cepstra with simple differences used for the PMC experiments.

Energy Term	Dynamic Terms	Test Set			Average
		Feb'89	Oct'89	Feb'91	
Log	Regr.	4.5	5.1	4.0	4.5
Cepstra	Diff.	4.3	5.4	3.9	4.6

Table 8.3: Word error rates (%) of a six component model set on clean RM

A set of six component models, similar to those used for the ARPA RM system developed at CUED [96], were then generated. Table 8.3 shows the performance of this six component system on the clean RM test sets. Comparing these results with those published using the standard HTK parameterisation [96] shows little difference in performance, 4.6% compared to 4.5% for the standard HTK RM feature set. The six component model set with zeroth Cepstra energy and simple difference dynamic parameters was the standard model set used in this chapter.

8.3 Results on RM with Additive Noise

This section describes various experiments using the RM database with additive noise only. Initially, three forms of matched system were investigated. Single-pass training with and without additional iterations of Baum-Welch were examined, in addition to the use of multi-pass training. For the single-pass system, the importance of compensating various "groups" of parameters, such as the static means, the static and delta means etc., was examined. Various PMC compensation techniques were implemented[4] and tested, both iterative and non-iterative. The performance of these PMC compensated model sets was compared to those of the matched systems.

8.3.1 Training on RM Additive Noise-Corrupted Data

Comp Mean Param	Comp Var Param	Test Set			Average
		Feb'89	Oct'89	Feb'91	
—	—	38.7	32.0	33.4	34.7
O	—	19.9	17.7	16.9	18.2
O,ΔO	—	15.3	13.5	12.0	13.6
O,ΔO,$\Delta^2 O$	—	10.9	11.1	9.0	10.4
O	O	16.1	13.7	13.5	14.4
O,ΔO	O,ΔO	10.2	10.1	8.4	9.6
O,ΔO,$\Delta^2 O$	O,ΔO,$\Delta^2 O$	7.3	8.6	6.9	7.6

Table 8.4: Word error rates (%) compensating various parameter groups using single-pass training on additive Lynx noise-corrupted RM at 18dB

A six component system was trained in a single pass on 18dB Lynx Helicopter additive noise-corrupted data. Various parameters of this system were merged with the clean system, to examine the importance of compensating each "group", static, delta and delta-delta, of parameters in a model-based compensation scheme. The results on the various test sets are shown in table 8.4. The first column shows which of the means are "compensated", and the second column which variances. In both cases O, ΔO and $\Delta^2 O$ represent static, delta and delta-delta parameters respectively.

Simply compensating the static means, as is done with hypothesised Wiener filtering for example, reduced the error rate by 48%. A total reduction of 60% was achieved by also compensating the delta means, and finally, incorporating the delta-delta compensated means yielded a 70% reduction. For this noise it can clearly be seen that both the delta and the delta-delta means must be compensated to achieve good performance. If in addition to the means, the variances were compensated the following reductions in error rate were obtained: for the statics 58%, the static and deltas 72%, and all the means and variances of the system 78%. Compensating the static means gave the largest reduction in error rate, however it can be seen that to obtain the best performance both means and variances must be compensated.

[4] In this section Numerical Integration is not considered. The Numerical Integration scheme described and used in the previous chapter showed comparable results to DPMC, though at a far greater computational overhead.

The feature vector was then truncated, by removing the delta-delta parameters, to investigate whether these parameters were useful for this particular noise condition. The model set was not retrained with this new truncated feature set, so the results quoted additionally assume that the frame/state component alignment was not dramatically altered by removing the delta-delta parameters. The average word error rate over the three test sets was 8.6%. This indicates that, at least within the constraint that the alignment did not alter, the use of compensated delta-delta parameters was advantageous on this task. However, on this task it was better to remove the delta-deltas rather than leaving them uncompensated. For all subsequent experiments the complete feature vector was used, as in addition to yielding the best results on this noise-corrupted data, it gives a 2 to 3% (absolute) improvement over a wide range of system types [96] for the RM task in clean conditions.

Model Set	Test Set			Average
	Feb'89	Oct'89	Feb'91	
Single-Pass	7.3	8.6	6.9	7.6
Single-Pass (+2)	7.7	8.6	7.4	7.9
Multi-Pass	8.1	7.9	7.3	7.8

Table 8.5: Word error rates (%) using single-pass and multi-pass training on Lynx Helicopter additive noise-corrupted RM at 18dB

Having obtained a single-pass model set, it is possible to perform additional iterations of Baum-Welch estimation. This should partially overcome the poor modelling of the corrupted-speech distribution by a single multi-variate Gaussian, see Chapter 4. Table 8.5 shows the performance of the models generated by an additional two Baum-Welch estimations, *Single-Pass (+2)*. Despite the fact that the models were a better, in a likelihood sense, representation of the waveform, the recognition performance dropped slightly compared with the standard single pass training model set. This is felt to be due to the poor frame/state alignments obtained when training on noise-corrupted data.

In addition to the single-pass retraining, it is possible to cluster and train a noise-corrupted system from scratch on the noisy data, labelled *Multi-Pass*. Since the clustering was performed on the noisy data it differed from the clean clustering, though approximately the same number of distinct states were generated. The results for such a system are also shown in table 8.5. The performance of such models was comparable to the single-pass trained system, though again a slight drop in performance was observed. This agrees with the performance of multi-pass model sets on the NOISEX-92 database.

The SNR was decreased by about 8dB to 10dB SNR, an attenuation of the original noise signal by 12dB. The results for the single-pass model sets are shown in table 8.6. The performance of the single-pass model set was poor, showing a 256% increase in error rate over the clean conditions, though dramatically better than the clean model set on 10dB Lynx Helicopter noise. Two additional iterations of Baum-Welch were then performed on this model set. This further degraded the performance by 23%. This decrease is felt to be due to the poor alignments achieved when corrupted-speech data is used. This explanation of the decrease in performance is further backed up by the fact that the degradation due to the additional iterations of Baum-Welch was far worse in this 10dB system than the 18dB system previously investigated.

Model Set	Test Set			Average
	Feb'89	Oct'89	Feb'91	
Clean	86.6	84.7	85.1	85.5
Single-Pass	18.3	15.2	15.7	16.4
Single-Pass (+2)	21.4	17.7	21.2	20.1

Table 8.6: Word error rates (%) using single-pass training on Lynx Helicopter noise-corrupted additive noise-corrupted RM at 10dB

8.3.2 PMC on Lynx Additive Noise-Corrupted RM

This section details a range of PMC experiments carried out using the RM database with noise from the Lynx Helicopter NOISEX-92 source artificially added at a variety of SNRs. Performance was assessed in terms of both average KL number and word error rates. The initial set of experiments only considered non-iterative PMC. The first systems examined were based on the Diagonal covariance approximation at 18dB SNR. Both the number of points required in a non-iterative DPMC scheme and the use of Cepstral smoothing were examined. Various non-iterative PMC approximations were compared to the DPMC and single-pass trained systems. The importance of variance compensation was also explored at a variety of SNRs. Different covariance approximations were then considered and assessed at two different SNRs, 18dB and 10dB. The final set of experiments examined the use of iterative PMC implementations. For all experiments a single component single state noise model was used.

The first aspect of the non-iterative PMC compensation scheme assessed was how closely did the compensated model set match the single-pass, matched, model set. The previous chapter illustrated that static parameters may be well compensated using PMC. However, here where dynamic parameters were incorporated in the feature vector, it was necessary to compensate delta and delta-delta parameters. Figure 8.1 shows the average KL number for both a clean system and a non-iterative DPMC-compensated system, using 1000 data points and the Diagonal covariance approximation for the compensation. All model sets were compared to the single-pass trained system unless otherwise stated. In the figure, feature vector coefficients 1 to 13 are C_0 to C_{12}, 14 to 26 are the respective delta parameters, and 27 to 39 the delta-delta parameters.

For the clean model set, *Clean*, the static parameters were the most affected by additive noise, with the low-order static Cepstra being distorted to a greater extent than the higher-order Cepstra. The delta parameters were less affected than the static parameters, though the same general shape was observed. The delta-delta parameters were affected, approximately, to the same extent as the delta parameters. This agrees with the word error rate results in table 8.4, where best performance was achieved when all the parameters were compensated. The use of DPMC, labelled *Diagonal*, reduced the average KL number, in particular for the static and the delta parameters. However, the delta-delta parameters were not so well compensated, especially the lower-order delta-delta parameters. This may be attributed to the lack of correlation modelling between the static and delta-delta parameters.

In contrast to the NOISEX-92 experiments, Cepstral smoothing was used for the RM task. This meant that information, in the form of the higher-order static, delta and delta-

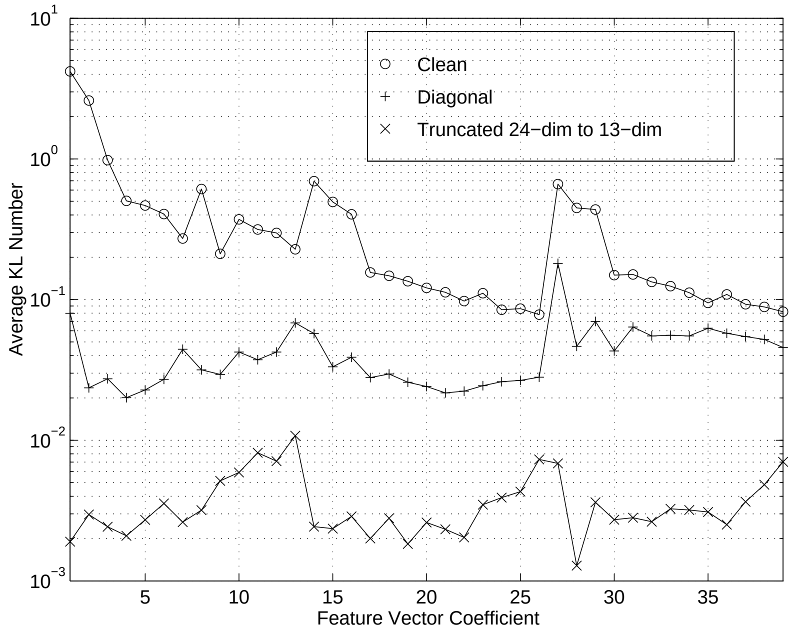


Figure 8.1: Average KL number for clean, DPMC-compensated and 24-dimensional truncated to 13-dimensions DPMC compared to 13-dimensional DPMC-compensated model sets on Lynx Helicopter additive noise-corrupted RM at 18dB

delta Cepstra, was lost. Thus for the compensation process, the models were zero padded as described in section 5.10.3 to allow them to be mapped from the Cepstral to the Log-Spectral domain. To investigate the effects of this, a system with no truncation, i.e. using all 24 dimensions, was built using the same complete dataset as the standard truncated models. This 24-dimensional model set was then compensated using non-iterative DPMC with the Diagonal covariance approximation, and finally truncated to the same dimensions as the standard RM model set. The average KL number between this non-truncated compensated model set and the standard compensated set is also shown in figure 8.1, labelled *Truncated 24-dim to 13-dim*. This plot has a low average KL number indicating that the two model sets were "close" to one another. As expected, the higher-order Cepstra were more affected by the truncation process. In terms of word error rate, however, this truncation was found not to affect performance compared to using all Cepstral parameters, compensating and then truncating. For all further experiments in this section, truncated feature vector models were used in the compensation scheme.

For the model sets compared above 1000 data points were used for each component in the DPMC process. In practical systems, it is necessary to minimise the computational overhead associated with noise compensation process. To this end it would be preferable to reduce the number of points used. Table 8.7 shows the average performance of ten randomly initialised runs with varying number of points used in the DPMC process. Using 1000 to 100 points showed little variation in performance. To put the standard deviation, *Std*, in context, there are 2561 words present in the Feb'89 test set, thus a standard deviation of 0.16 equates to 4 words. Below 100 points both the mean word error rate and the standard deviation increased. For all the remaining results quoted in this section where DPMC was used, 1000 observations, initialised with the same random seed, were

Number Points	Word Error			
	Mean	Std	Max	Min
1000	7.48	0.16	7.69	7.15
500	7.51	0.13	7.69	7.38
250	7.45	0.14	7.61	7.26
100	7.48	0.22	7.81	7.15
50	7.72	0.32	8.16	7.03
25	8.41	0.33	8.90	7.81

Table 8.7: Word error rates (%) of DPMC-compensated model sets with various numbers of "observations" on Feb'89 test set with Lynx Helicopter additive noise-corrupted RM at 18dB

generated for each component pairing.

It is not necessary to use DPMC to compensate both the means and the variances of the models. Figure 8.2 shows the average KL number for non-iterative DPMC mean-and-variance-compensated and mean-compensated, *Diagonal* and *Diagonal (Mean)* respectively, model sets both using the Diagonal covariance approximation. In terms of average KL number compensating the variance was important. Again this agrees with the single-pass results shown in table 8.4.

In addition to the DPMC-compensated models, figure 8.2 shows the average KL number of the Log-Add approximation described in section 5.6[5]. Two implementations of the Log-Add approximation were considered. The first used models of all the statistics of $\mathbf{V}^c(\tau)$, labelled *Log-Add*. This achieved comparable performance to the DPMC mean-compensated scheme in terms of the average KL number, indicating that the effect of the variance on the value of the mean was small. Using the Log-Add approximation, with the standard HMM parameters, that is no $\tau - w$ statistics, *Std. Log-Add*, achieved poor performance on the dynamic coefficients. This highlights the need to explicitly model all the parameters used in the mismatch functions for good compensation performance.

Model Set	Test Set			Average
	Feb'89	Oct'89	Feb'91	
Diagonal	7.5	8.1	6.4	7.4
Diagonal (Mean)	11.6	10.8	9.1	10.5
Log-Add	12.2	10.7	9.4	10.8
Standard Log-Add	14.9	13.4	12.2	13.5
Log-Normal	18.4	14.8	14.3	15.8

Table 8.8: Word error rates (%) of various PMC-compensated model sets on Lynx Helicopter additive noise-corrupted RM at 18dB

The word error rates associated with the model sets shown in figure 8.2 are shown in ta-

[5]The Log-Normal approximation is not shown, as this is only applicable to compensating the static parameters and this has already been illustrated in figure 7.1.

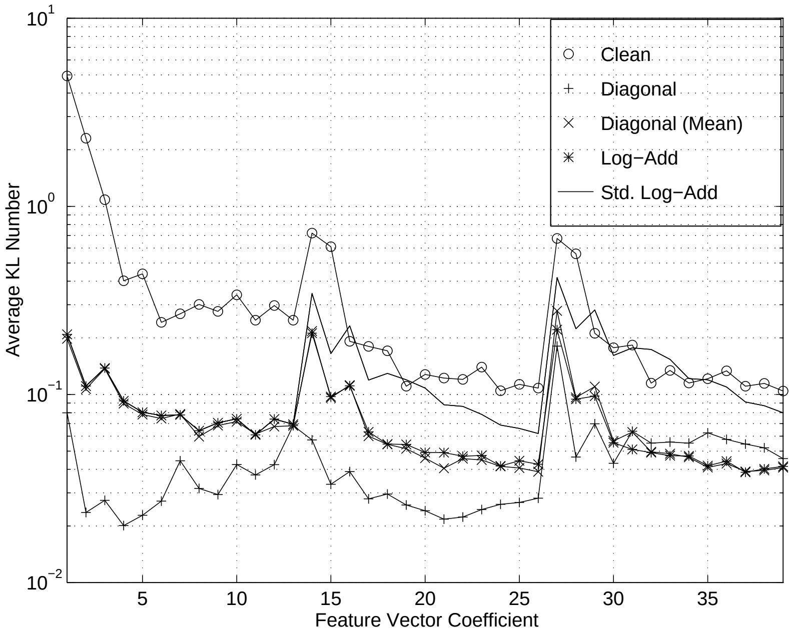


Figure 8.2: Average KL number for the clean, DPMC mean-and-variance-compensated, DPMC mean-compensated, the Log-Add approximation and Standard Log-Add approximation model sets on Lynx Helicopter additive noise-corrupted RM at 18dB

ble 8.8. For the DPMC mean-compensated system performance was comparable with that of the matched system where only the means are altered, 10.4% from table 8.4. However, the performance was worse than that achieved by compensating the means and variances, again in line with the results for the matched system. Table 8.8 also shows the performance of some alternative non-iterative PMC schemes. The performance of the Log-Add approximation was comparable to the DPMC mean-compensated system. This confirms, in terms of word error rate, that the variances of the clean models had little effect on the compensated means. The Standard Log-Add approximation degraded the performance compared to the Log-Add system. Even using the simpler approximate schemes it is important to model all the compensation parameters. The recognition performance of the Log-Normal approximation is also shown in table 8.8, here just the mean and variances of the static parameters were compensated. This shows a slight degradation in performance compared to the matched system, 14.4% from table 8.4.

The word error rates using non-iterative DPMC with the Diagonal covariance approximation to compensate either the means or the means and variances at various SNRs on the Feb'89 test set is shown in table 8.9. The interesting aspect of this table is the comparison between the mean-compensated and the mean-and-variance-compensated systems. Though the results were not optimised for the insertion penalty, in particular the 10dB and 4dB mean-compensated system, the importance of the compensating variances for good performance increased as the SNR decreased[6]. As the performance of the system at the 4dB noise condition was very poor, dropping the SNR further was felt not to yield useful results.

[6] The mean-compensated systems were examined at a variety of insertion penalties and grammar scale factors. Little increase in accuracy was observed even on the 4dB and 10dB systems.

Approx SNR	Model Set	Words Corr	Subs Err	Del Err	Ins Err	Word Err
Clean	—	96.1	2.3	1.5	0.4	4.3
24	Diagonal	95.1	3.3	1.6	0.6	5.5
	Diagonal (Mean)	94.2	3.6	2.3	0.5	6.3
18	Diagonal	93.0	4.8	2.1	0.5	7.5
	Diagonal (Mean)	89.3	7.4	3.4	0.9	11.6
10	Diagonal	81.6	13.9	4.5	1.6	20.0
	Diagonal (Mean)	63.2	21.7	15.0	1.3	38.1
4	Diagonal	61.0	26.8	12.2	2.7	41.7
	Diagonal (Mean)	30.8	29.7	39.5	1.1	70.3

Table 8.9: Word error rates (%) compensating means and variance, and just means, using DPMC on the Lynx Helicopter additive noise-corrupted RM Feb'89 test set

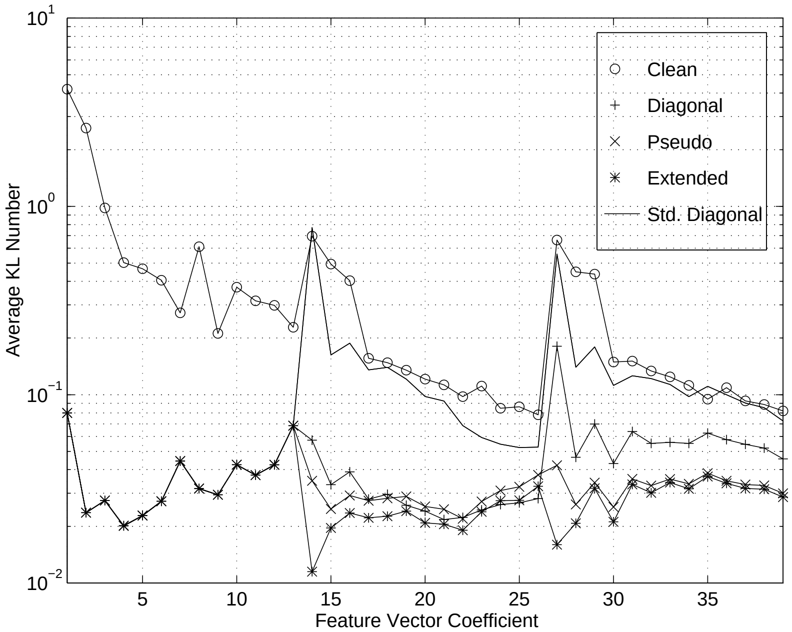


Figure 8.3: Average KL number for clean model set and DPMC-compensated model sets using Diagonal, Pseudo, Extended and Standard Diagonal covariance approximations on Lynx Helicopter additive noise-corrupted RM at 18dB

The DPMC experiments described so far have all used the Diagonal covariance approximation. Section 5.10 described other covariance approximation schemes, Extended, Pseudo and Standard Diagonal. The labels of the various model sets correspond to the descriptions given in section 5.10.1. The recognition system was compensated with each of these covariance approximations. The average KL numbers for each system is shown in figure 8.3. The static parameter compensation is independent of the covariance approximation used and this was reflected in the average KL number measure. As the covariance approximation became more accurate, the average KL number decreased for the dynamic parameters. The use of the Pseudo approximation, though better than the Diagonal case, was slightly worse than the Extended covariance approximation. The improvement from those covariance approximations where some correlation was modelled was particularly noticeable for the delta-delta parameters. This is not surprising, as the static and the delta-delta parameters are known to be highly correlated, whereas the delta parameters do not exhibit such a high correlation. When the Standard Diagonal system was used, labelled *Std Diagonal* in figure 8.3, the average KL number measures for the dynamic parameters was far larger than the other compensation schemes.

Model Set	Test Set			Average
	Feb'89	Oct'89	Feb'91	
Clean	38.7	32.0	33.4	34.7
Single-Pass	7.3	8.6	6.9	7.6
Diagonal	7.5	8.1	6.4	7.4
Pseudo	7.1	7.7	6.8	7.2
Extended	6.7	7.9	6.6	7.1
Standard Diagonal	9.6	9.0	8.2	8.9

Table 8.10: Word error rates (%) using various DPMC covariance approximations on Lynx Helicopter additive noise-corrupted RM at 18dB

The recognition performance of the various covariance approximations is shown in table 8.10. Comparing the Diagonal covariance approximation with the matched system showed approximately the same level of performance. The Pseudo and Extended approximations both slightly improved the performance, with only a little difference between these two cases. If the Standard Diagonal covariance approximation was used there was a marked drop in performance. Hence, for good recognition performance on this database, it was necessary to explicitly model the additional parameters required for the dynamic coefficient mismatch functions, agreeing with the observations using the average KL number.

The SNR was lowered by 8dB to about 10dB SNR, an attenuation of the original noise signal by 12dB. The word error rates for the various DPMC covariance approximations are shown in table 8.11. At this lower SNR the need to explicitly all the parameters of the mismatch functions and some correlations became more obvious. Both the Pseudo and Extended covariance approximations reduced the word error rate by about 10% over the Diagonal case. Though the performance of the Pseudo and Extended covariance approximations were about the same in terms of the word error rate, the Extended covariance approximation was again found to have a lower average KL number for the dynamic parameters. It is interesting to note that the Pseudo and Extended covariance approxima-

Model Set	Test Set			Average
	Feb'89	Oct'89	Feb'91	
Clean	86.6	84.7	85.1	85.5
Single-Pass	18.3	15.2	15.7	16.4
Diagonal	20.0	15.8	16.3	17.4
Pseudo	17.6	14.4	15.2	15.7
Extended	17.3	14.2	15.3	15.6
Standard Diagonal	24.8	19.3	20.4	21.5

Table 8.11: Word error rates (%) using various DPMC covariance approximations on Lynx Helicopter additive noise-corrupted RM at 10dB

tions performed slightly better than the matched system (this is also true for the results in table 8.10), though the difference was only slight. A possible explanation is that in the compensation schemes all clean speech models are effectively observed in all noise conditions. For components which are not observed many times in the training data, the matched system may not have observations in all noise conditions.

Approx SNR (dB)	Model Set	Test Set			Average
		Feb'89	Oct'89	Feb'91	
18	Extended	6.7	7.9	6.6	7.1
	Iter. Extended	6.7	7.0	6.6	6.8
	Mix Up Extd.	6.5	6.6	6.0	6.4
10	Extended	17.3	14.2	15.3	15.6
	Iter. Extended	14.8	12.7	13.5	13.7
	Mix Up Extd.	13.8	11.8	11.8	12.5
4	Extended	35.8	26.9	31.2	31.2
	Iter. Extended	30.1	23.2	27.5	26.9
	Mix Up Extd.	28.5	22.5	25.4	25.4

Table 8.12: Word error rates (%) using iterative PMC and the Extended covariance approximation on Lynx Helicopter additive noise-corrupted RM Feb'89 test set

As mentioned in Chapter 4, the assumption that the corrupted-speech distribution is Gaussian distributed is poor. There are two approaches that may be adopted to solve this problem, both based on iterative PMC using DPMC. The corrupted-speech distributions may be viewed on a per-state level. This is then a standard training problem of fitting a set of Gaussians to an observed PDF. In this case the frame/state component alignment of all the components within a state is altered to optimise the likelihood. This was implemented using the Extended covariance approximation. The initial estimates used in the iterative process were the models obtained using non-iterative DPMC. This is labelled *Iter Extended* in table 8.12. Alternatively, the frame/state component alignment may remain fixed and multiple components used to model the distribution for a particular speech noise component pairing. Again, this was implemented with the Extended covariance approximation, labelled *Mix Up Extd*, using two mixture components to model each component pairing. When the frame/state component alignment within a state is varied, or multiple components are used to model a particular pairing, the iterative PMC compensation scheme is required. This involves additional computational overhead at the compensation stage, and in the case of the "mixing up", an increase at the recognition stage due to the doubling of the number of corrupted-speech components[7]. Both iterative schemes used 10 iterations of Baum-Welch.

From the results, the importance of accurately modelling the distributions can be seen. The use of iterative DPMC showed better performance than the non-iterative scheme, for example at 10dB the word error rate was reduced by 12% using iterative DPMC without increasing the number of components. This indicates that the frame/state component alignment within a state is altered by the addition of noise. This shifting of the component alignment may result either from the non-Gaussian nature of the corrupted-speech distributions, or due to the components modelling correlation introduced by the additive noise. The word error rate was further reduced by modelling each component pairing by two mixture components, the mixing-up case, giving a 20% reduction over the non-iterative DPMC case at 10dB SNR. This indicates that in noise-corrupted environments it may be necessary to increase the number of mixture components per state to obtain best performance. The importance of accurately modelling the corrupted-speech distribution increased as the SNR decreased. At 18dB SNR the improvement by mixing-up was only 10% compared to 20% improvements obtained at 10dB and 4dB SNR.

8.3.3 PMC on Operations Room Additive Noise-Corrupted RM

Having investigated the performance of PMC on the Lynx Helicopter noise-corrupted data, an alternative noise source, the Operations Room noise, was considered. As discussed in Chapter 6, this source has more temporal variation than the Lynx Helicopter noise and a larger high frequency component. Only non-iterative PMC schemes were examined on this noise source.

Initially the performance of DPMC on this new noise condition was assessed in terms of the average KL number between the various compensation schemes and a single-pass trained matched system. Figure 8.4 shows the average KL number against feature vector component for a variety of PMC compensation schemes. The same overall shape as the Lynx Helicopter noise systems was observed. The lower-order Cepstral parameters were the most affected by the addition of noise. As the complexity of the covariance approximation increased, the average KL number decreased for the dynamic coefficients, particularly for the delta-delta parameters. Thus the Extended compensation scheme had the lowest KL number, followed by the Pseudo, then the Diagonal covariance approximations. The Standard Diagonal case was not considered for this noise source, given its poor performance on the Lynx Helicopter noise source.

The word error rates for uncompensated, single-pass trained, and various PMC compensated systems are shown in table 8.13. The same general trends as for the Lynx Helicopter noise-corrupted data were observed, despite the very different nature of the interfering noise source. However, for this case there was little improvement in terms of word error rate in using the more complex covariance approximations.

The importance of compensating the variances was also investigated. Table 8.14 compares the performance of a DPMC compensated system to that of a DPMC mean-

[7] In practice more accurate modelling of the PDFs leads to more effective pruning and the actual computational load may not increase by much.

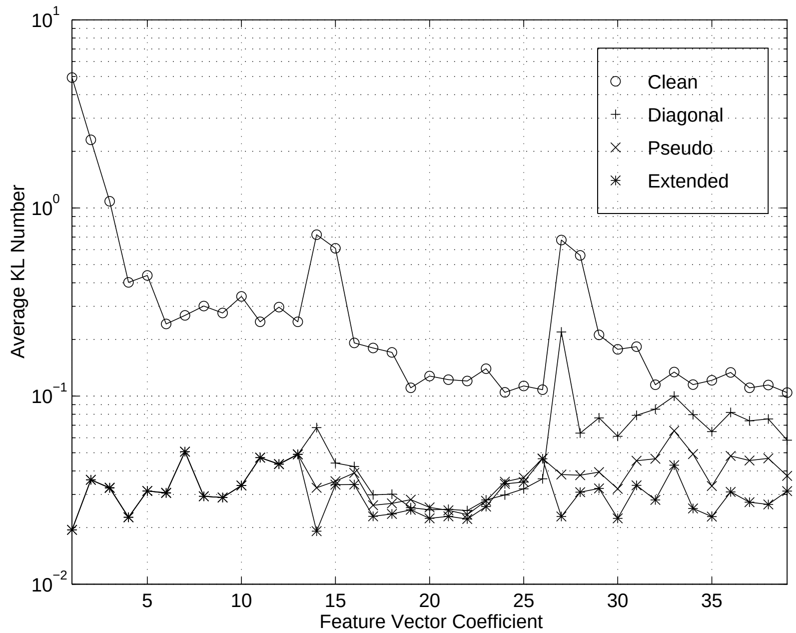


Figure 8.4: Average KL number for clean model set and DPMC-compensated model set using the Diagonal, Pseudo and Extended covariance approximations on Operations Room additive noise-corrupted RM at 18dB

Model Set	Test Set			Average
	Feb'89	Oct'89	Feb'91	
Clean	44.3*	39.6*	37.4*	40.5
Single-Pass	9.2	11.4	8.4	9.7
Diagonal	9.7	11.5	8.3	9.9
Pseudo	9.2	10.4	8.5	9.4
Extended	9.5	10.6	8.9	9.7

Table 8.13: Word error rates (%) using various DPMC covariance approximations on Operations Room additive noise-corrupted RM at 18dB, * indicates some sentences ran out of hypotheses

Model Set	Test Set			Average
	Feb'89	Oct'89	Feb'91	
Diagonal	9.7	11.5	8.3	9.9
Diagonal (Mean)	13.9	14.1	11.2	13.1

Table 8.14: Word error rates (%) of DPMC mean-compensated and mean-and-variance-compensated model sets on Operations Room additive noise-corrupted RM at 18dB

compensated system. Compensating the variance improved the noise robustness in this different noise condition, reducing the error rate by 24%. Again this agrees with the results obtained on Lynx Helicopter noise, showing that variance compensation is useful on the RM task.

8.3.4 PMC on Machine Gun Additive Noise-Corrupted RM

Having established that PMC can give performance comparable to single-pass training, or better when the frame/state component alignment within a state is allowed to vary, on relatively stationary noise conditions, a complex-noise condition was investigated. A distinctly non-stationary source, the Machine Gun noise, from the NOISEX-92 database was chosen. The noise was added at a scaling factor of 0.5 to give approximately a 4dB SNR. For this section only non-iterative DPMC with the Diagonal covariance approximation was considered.

Noise Components	Test Set Feb'89	Oct'89	Feb'91	Average
Clean	51.6*	51.1*	52.3*	51.7
1	13.6	13.9	10.8	12.8
2	11.4	10.8	9.0	10.4
2→1	13.0	13.0	10.0	12.0

Table 8.15: Word error rates (%) of clean and DPMC-compensated model sets on Machine Gun noise-corrupted RM at 4dB, *indicates some sentences ran out of hypotheses

The results are shown in table 8.15. For these results the frame/state component alignment was assumed unaltered by the noise, so if a 2 Gaussian component noise model is used then the resulting system will have twice the number of Gaussian components. The simple noise model performed remarkably well, given the nature of the noise. Increasing the complexity of the noise model reduced the error rate by 19%. If the number of Gaussian components was then reduced, so that both corrupted-speech components generated by the same clean speech component were merged, labelled *2→1*, a slight improvement over the simplest noise model was observed, 6%. Note that when the noise source is less distinctly multi-state, such as the Operations Room noise, little gain was observed when the complexity of the noise model was increased.

Init Models	Test Set Feb'89	Oct'89	Feb'91	Average
Speech	11.9	11.5	10.1	11.2
Heaviest	11.8	11.3	9.7	11.0
Mutual Info.	12.1	12.0	10.3	11.5

Table 8.16: Word error rates (%) of iterative DPMC- compensated model sets on Machine Gun noise-corrupted RM at 4dB.

Using iterative DPMC it is possible to allow the frame/state component alignment within a state to alter, and adjust the number of components in the corrupted-speech model. In contrast to the work on Lynx Helicopter noise, the aim here is to model a complex-noise source without increasing the number of components in the recognition system. For these iterative schemes it is necessary to specify the initial models to be used. Three different initial model sets were examined. The first, and the one used for the iterative experiments on the Lynx Helicopter noise, was to use the models determined by merging according to the speech component. This is labelled *Speech* in table 8.16. Alternatively, all the corrupted-speech components from each pairing were calculated and the components with the largest component weights were chosen, in this case the six heaviest components. This is labelled *Heaviest*. Finally, all the corrupted-speech components were calculated and then merged, according to a symmetric divergence distance measure based on the complete feature vector, until the required number of components were achieved. This is marked as *Mutual Info*. In all cases 10 iterations of Baum-Welch re-estimation were performed.

The word error rates for each initial model set are shown in table 8.16. All three schemes performed approximately the same, though the use of the heaviest components in this noise condition consistently gave slightly better performance yielding a 14% improvement compared with the simplest noise model case. In all cases the average performance was better than using both a single-component noise model and a two component noise model mixed down to a single component with no re-estimation. However, the performance was still worse than that achieved using a two component noise model without merging any mixture components, but now the possible recognition-time computational overhead of increasing the number of components is avoided.

8.4 Results on RM with Additive and Convolutional Noise

The previous section has only considered the additive noise case. Convolutional noise was incorporated into the system, by removing the pre-emphasis on the test data. This resulted in the corrupted speech being approximately unaffected at high frequencies and amplified at low frequencies compared to the clean speech. Various DPMC convolutional noise schemes were tested on RM with this additive and convolutional noise present. No *a-priori* assumptions, such as smoothness of the spectral difference, were made about the convolutional noise, only that it was constant with time. The noise model used was trained under the new spectral tilt condition, as this model is expected to be generated *on-line*. A 30-component global speech model was used in the convolutional noise estimate. Both the iterative and non-iterative DPMC schemes were examined. Note the SNR quoted in this section were not recalculated and were generated using the same noise attenuation as the previous section.

Table 8.17 shows the performance of the non-iterative DPMC additive noise compensation with the Diagonal covariance approximation in the presence of additive and convolutional noise. The addition of spectral tilt has degraded the system performance by about 80%. All the test data was then used to estimate the convolutional noise, with the non-iterative estimation scheme described in section 5.8. The performance was approximately the same as that of the simple additive noise case[8]. In this situation the iterative scheme was unlikely to improve the performance, as there was sufficient test data that it

[8] As the noise was added randomly to the two test sets, the two test sets were not directly comparable. However given the stationary nature of the Lynx Helicopter noise, this difference should have little effect.

Spec Tilt	Comp Tilt	Adapt. Sent.	Test Set Feb'89	Oct'89	Feb'91	Average
$\times$	$\times$	—	7.5	8.1	6.4	7.4
$\surd$	$\times$	—	15.5	13.0	11.5	13.3
$\surd$	$\surd$	all	7.3	8.2	7.0	7.5

Table 8.17: Word error rates (%) of DPMC-compensated model sets on Lynx Helicopter additive and convolutional noise-corrupted RM at 18dB.

should be representative of the training speech. This result also indicates that there was no systematic difference between the training and test conditions on the RM task.

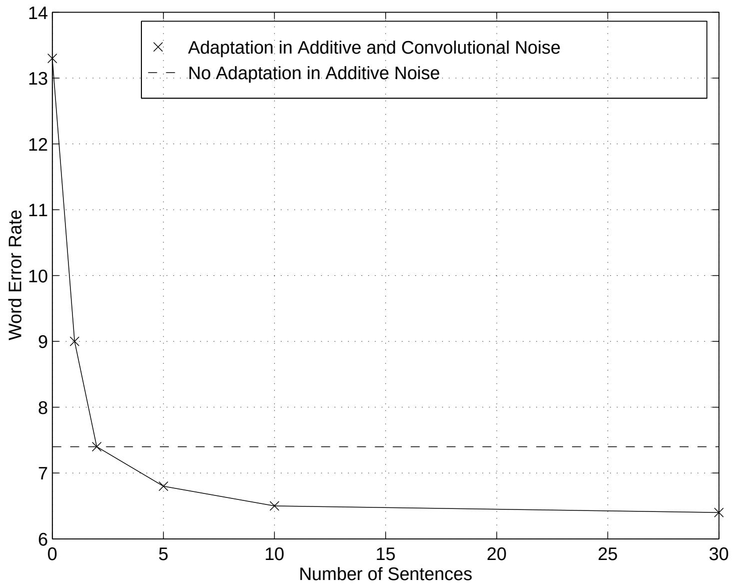


Figure 8.5: Average word error rate (%) on the Feb'89, Oct'89 and Feb'91 test sets against number of adaptation sentences using DPMC with the Diagonal covariance approximation on Lynx Helicopter additive and convolutional noise-corrupted RM at 18dB.

In practice it is seldom acceptable to wait until the end of a complete session before the recognition process is initialised. Figure 8.5 shows the average word error rate over the three test sets, Feb'89, Oct'89 and Feb'91, for the non-iterative scheme tilt estimation scheme against the number of adaptation sentences used to estimate the spectral tilt. The Diagonal covariance approximation was used in all cases and the convolutional noise adaptation was performed in a non-iterative fashion on a per-speaker level. Thus for the 1 adaptation sentence case, the first test utterance of each speaker was used to estimate the spectral tilt and the models compensated once for all the test sentences of that speaker. The average sentence length was just over 3 seconds for the test sets used. The 0 adaptation sentences point represents using DPMC to compensate for additive noise, but no convolutional noise. By using 2 adaptation sentences the performance was approximately the same as the additive noise DPMC system and the use of further sentences resulted in a 14% reduction in word error rate. This shows that the convolutional noise adaptation performed simple speaker adaptation, in addition to compensating for the convolutional noise component.

Spec Tilt	Comp Tilt	Adapt. Sent.	Test Set			Average
			Feb'89	Oct'89	Feb'91	
$\surd$	$\surd$	30^*	6.2	7.5	5.5	6.4
$\surd$	$\surd$	1^*	8.9	8.2	10.1	9.0
$\surd$	$\surd$	$1^\dagger$	8.0	7.8	7.1	7.6

Table 8.18: Word error rates (%) with per-speaker convolutional noise estimation on Lynx Helicopter additive and convolutional noise-corrupted RM at 18dB, * indicates that tilt compensation was performed on a speaker level and † indicates the use of the iterative scheme.

Table 8.18 shows a breakdown of two of the results shown in figure 8.5, the 30 sentence and the 1 sentence cases. The performance of the single adaptation sentence case was far poorer than the 30 sentence case. In particular, the performance on the Feb'91 test set, and to a lesser extent the Feb'89 test set, was seriously degraded. For the Feb'91 test set the performance of one speaker, *JWG*, dropped from 4.1% word error to 40.1% word error. Examining the adaptation sentence used, showed a speech to silence duration of 47.1% compared to an average of 96.5% over the whole of the Feb'91 test set. Looking at the average Cepstral values from all thirty *JWG* sentences and the single utterance case showed the C_0 value drops from 64.2 to 59.7. Since this is the dominant component when compensating for noise it was not surprising that the performance was seriously degraded.

To overcome the problem of mismatch between the training and adaptation data, the iterative scheme of section 5.8 was used, whereby the global speech model's component weights are altered using a recognition pass of the adaptation data to reflect the component occupancy of the adaptation data. This showed a consistent reduction in word error rate over all the test sentences for one sentence of adaptation data. As expected, the greatest improvement in performance was on the Feb'91 test set. The performance using this single sentence was not as good as using all 30 sentences, however, it was comparable to the no convolutional noise case.

An important question is whether such a simple convolutional noise compensation technique will work at lower SNRs. The SNR was, therefore, dropped to 10dB and the tests repeated. The results on this task are shown in table 8.19. Again, the introduction of convolutional noise badly degraded the performance of the system if it was left uncompensated. Using all the test data to estimate the spectral tilt brought the performance back to the level of no convolutional noise mismatch. Compensating for the convolutional noise on a per-speaker level improved the performance. Using all the test sentences on a per-speaker level reduced the average error rate by 21% compared to the matched condition. With a single adaptation utterance and the iterative estimation scheme the performance, though poorer than using all the utterances, was still better than using all the data from all the speakers.

All the convolutional noise experiments above used the Diagonal covariance approximation DPMC compensation scheme. From the RM additive noise section, this was not the best compensation scheme. Table 8.20 shows the performance of the best com-

Spec Tilt	Comp Tilt	Adapt. Sent.	Test Set Feb'89	Oct'89	Feb'91	Average
$\times$	$\times$	—	20.0	15.8	16.3	17.4
$\surd$	$\times$	—	34.9	25.0	28.2	29.3
$\surd$	$\surd$	all	20.0	17.3	15.8	17.7
$\surd$	$\surd$	30^*	14.6	14.8	11.7	13.7
$\surd$	$\surd$	$1^\dagger$	17.4	16.5	14.4	16.1

Table 8.19: Word error rates (%) of DPMC-compensated model sets on Lynx Helicopter additive and convolutional noise-corrupted RM at 10dB, * indicates that tilt compensation was performed on a speaker level and † indicates the use of the iterative scheme.

Spec Tilt	Comp Tilt	Adapt. Sent.	Test Set Feb'89	Oct'89	Feb'91	Average
$\surd$	$\surd$	30^*	11.7	11.1	9.8	10.9
$\surd$	$\surd$	$1^\dagger$	13.9	13.2	12.9	13.3

Table 8.20: Word error rates (%) of iterative DPMC-compensated model sets with the Extended covariance approximation on Lynx Helicopter additive and convolutional noise-corrupted RM at 10dB, * indicates that tilt compensation was performed on a speaker level and † indicates the use of the iterative scheme.

pensation schemes. The best schemes used the Extended covariance approximation and iterative DPMC with two components per speech-and-noise component pairing, doubling the number of components in the recognition system. Using this more complex and computationally intensive compensation scheme decreased the word error rate by 20% over the average of the three test sets for the 30 adaptation sentences case. Comparing this result with the performance on the additive noise test sets[9], table 8.12, shows an increase in performance of 13%. Even with a single sentence to estimate the convolutional noise the performance was good. Comparing with the simple additive noise case showed only a slight drop in performance of 6%.

8.5 Discussion

This section has investigated the use of PMC and DPMC on a speaker-independent medium-size vocabulary task. For such a task, it is necessary to incorporate dynamic coefficients into the feature vector to achieve good recognition performance in the clean environment. Having incorporated dynamic coefficients into the feature vector, the importance of compensating them in a noise-corrupted environment was investigated using a matched system. The various dynamic coefficient compensation schemes were examined and compared in terms of both average KL number and word error rate. Both stationary and non-stationary noise conditions were used. Furthermore, the use of iterative PMC

[9] Again, the two test sets were not identical, as the noise was added randomly to the speech signal. However the average performance should be comparable.

schemes was investigated. Finally the use of DPMC for situations where there is both additive and convolutional noise was examined.

The first question investigated was which parameters affect the performance of the speech recognition system in noise when dynamic coefficients are incorporated into the feature vector. This was examined by building a matched single-pass system where only specific elements of the feature vector were compensated. For the 18dB Lynx Helicopter additive noise case it was found that all the elements of the feature vector were affected by the addition of noise and contributed to the degradation in performance.

The corrupted-speech distribution is known to be non-Gaussian. Improved performance should therefore be obtained by allowing the component alignments to vary, thus better modelling the corrupted-speech distribution. However, the use of additional iterations of Baum-Welch re-estimation in the single-pass system was not found to improve performance. Furthermore, training a system from "scratch" on the corrupted data did not give better performance than the single-pass trained system. This failure, despite the poor corrupted-speech modelling by a single multivariate Gaussian, can be attributed to the poor frame/state alignment achieved when noise-corrupted data is used in training.

Having established that for good recognition performance it was necessary to compensate all the parameters of the model set, the various dynamic coefficient compensation schemes within the PMC framework were examined. This concentrated on which statistics must be explicitly stored in the clean speech and noise models. Improving the correlation modelling between elements of the feature vector in the compensation process reduced both the average KL number and the word error rate for the Lynx Helicopter noise. The best results on the 18dB Lynx Helicopter noise data were achieved by the Extended covariance approximation, where a quin-diagonal covariance matrix was used. However such an approximation required additional storage space for the model sets. Similar results were obtained at lower SNRs.

Still using the Lynx Helicopter noise, the effect of varying the frame/state component alignment within a state, iterative PMC, was examined. The Extended covariance approximation was used for these experiments. Firstly, the same number of components as the original clean models was used and the alignments within the state allowed to alter. This improved the performance by 4% on the 18dB system and 12% on the 10dB system. By using multiple components to model the corrupted-speech distribution for each speech and noise component pairing, a 10% improvement in performance on the 18dB data, and 20% at 10dB were achieved. This indicates that for best performance in noise-corrupted environments it is necessary to allow the frames/state component alignment within a state to vary. Furthermore, it is preferable to increase the number of mixture components per state.

Similar results were obtained on a noise source with slightly more temporal variability, the Operations Room noise.

The performance in a complex-noise environment, the Machine Gun noise was then examined. Good performance on this task was achieved using a single-component noise model, despite its poor modelling of the noise source. A two-component noise model reduced the error rate by 19% compared to the single component case. However, this case may result in a recognition-time computational overhead due to doubling the number of components in the system. To avoid increasing the number of components, corrupted-speech model components were merged according to the speech component that generated them. This resulted in a slight improvement, 6%, compared to the single component noise model case. As an alternative to the simple merging of components to reduce the number, the iterative PMC schemes may be used. By allowing the component alignment within a state to vary error rate reductions up to 14% compared to the single noise component case were observed without any recognition time computational overhead. This shows that iterative PMC may be used to improve performance in complex-noise environments.

The final section in this chapter addressed the general problem of additive and convolutional noise. Convolutional noise was artificially introduced by removing the pre-emphasis on the test data. Initially, the convolutional noise compensation was investigated to find out whether, given sufficient training data the spectral tilt may be estimated and incorporated into the PMC process. Using all the test sentences to estimate the convolutional noise, performance comparable to the case where only additive noise was present was achieved. This also shows that there is no systematic mismatch between the training and the test conditions for RM.

For practical applications it is not normally possible to wait to the end of a session to produce results, so the effect of the number of sentences on the convolutional noise estimate was also investigated. Using two sentences on a per-speaker basis, about 6 seconds of speech, comparable performance to the convolutional noise bias estimated using all the test data (300 sentences) was achieved. The two-sentence performance was obtained using a single sentence with weight adaptation of the general speech model. Using all the test data on a per-speaker level to calculate a convolutional noise bias for that speaker, a 15% gain was achieved over calculating a convolutional noise bias over all the speakers. This indicates that the tilt compensation can perform a simple version of speaker adaptation, in addition to compensating for the convolutional noise.

In summary, this chapter has shown that PMC, and in particular DPMC, can achieve good performance on a standard medium-size vocabulary task. Convolutional and additive noise can be handled within this framework. The level of performance attainable is dependent on the computational cost available at the compensation stage. The best performances were achieved by using iterative DPMC schemes that allow the frame/state component alignment within a state to alter.

Chapter 9

Evaluation on the ARPA 1994 CSRNAB Spoke 10

The ARPA 1994 CSRNAB evaluation involved, as a spoke task (S10), an evaluation of speech recognition systems in the presence of additive interfering noise. This is a large vocabulary task, with the added advantage that other sites around the world evaluated their systems using the same training and test data.

9.1 The Wall Street Journal System

The ARPA 1994 CSRNAB Spoke 10 consists of data taken from a closed 5000 word vocabulary task with car noise added at various SNRs. The baseline system used for the recognition task was a gender-independent cross-word-triphone mixture-Gaussian tied-state HMM system and was similar to that used in the CUED HTK 1994 evaluation system [95]. However, there were some important modifications made to allow the use of PMC to compensate the model parameters. The set of speech parameters consisted of 12 MFCCs, C_1 to C_{12}, appended by C_0, not normalised log-energy as in the standard system, and the first and second differentials were based on simple differences rather than linear regression. This yielded a 39-dimensional feature vector. Note that, in contrast to the standard HTK system, no Cepstral Mean Normalisation (CMN) was used.

The acoustic training data consisted of 36493 sentences from the SI-284 WSJ0 and WSJ1 sets, and the LIMSI WSJ lexicon and phone set were used. Since CMN was not incorporated, it was necessary to map the global signal levels of the WSJ0 and WSJ1 databases to be the same. This was achieved by offsetting the C_0 feature vector coefficient of the WSJ0 data to have the same average value as the WSJ1 database. To generate the models with the new parameter set, a standard HTK system was trained initially. This used decision-tree-based state clustering [99] to define 6399 speech states. A 12 component mixture Gaussian distribution was then trained for each tied state, a total of about 6 million parameters. The parameter set was then changed to that described above in a single pass and an additional smoothing pass performed. For the S10 spoke, the standard MIT Lincoln Labs 5k trigram language model was specified. Decoding used the single-pass dynamic-network decoder [72].

As the set of speech parameters was altered from the standard HTK WSJ system, in particular the removal of CMN from the static parameters, it was necessary to use different grammar scale factors and insertion penalties to those normally used. These were estimated on the 1994 S0 development spoke data. The S0 dataset was used as it is a comparable task to the S10 task with more test sentences (the S0 dataset consists of 424 sentences spoken by 20 speakers, the S10 dataset is 108 of these sentences spoken by 10 speakers). Note that this dataset is a superset of the S10 dataset, thus the development S10 dataset results may be slightly biased. On the S0 dataset, the optimised performance figure was 5.84%, compared with 6.19% for the standard HTK WSJ system and parameter settings[1]. These clean speech settings were used for all the noise conditions.

For this chapter, either DPMC in the form of the non-iterative Diagonal covariance approximation or the Log-Add approximation was used to compensate the models unless otherwise stated.

9.2 Results on the Development Data

The ARPA 1994 CSRNAB Spoke 10 development data consists of 108 sentences spoken by 10 different speakers, recorded in "clean" conditions. Noise from three sources labelled Car1, Car2 and Car3 was then added at 5 SNRs, 6dB, 12dB, 18dB, 24dB and 30dB. The SNR was calculated by NIST, using an A-weighted scale, comparing the maximum speaker value over all the sentences and the noise mean value. The 6dB noise condition was found to be too severe to obtain meaningful results so was not considered in any of the development results. For the development results only one of the car noises, Car3, was used.

Model Set	SNR (dB)				
	Clean	30	24	18	12
Diagonal	7.2	11.0	14.2	19.9	34.1
Diagonal (Mean)	—	10.4	13.9	20.7	36.6
Log-Add*	—	10.3	14.0	21.0	37.0

Table 9.1: Word error rates (%) of PMC-compensated model sets on Car3 ARPA 1994 CSRNAB Spoke 10 development data, * indicates that the results were generated by re-scoring the Diagonal mean-compensated lattices.

Table 9.1 shows the results on the four development noise levels. At the higher SNRs, 30dB and 24dB, the mean-compensated system outperformed the mean-and-variance-compensated system. This was slightly surprising given the previously quoted results on the RM database, where compensating the variances was found to improve the performance up to 24dB[2]. As the SNR dropped, the variance compensation was found to slightly improve the performance, though not by the amount observed on the RM task. The results using the Log-Add approximation are also shown in table 9.1. Use of this

[1] The standard HTK WSJ system when optimised yielded a figure of 5.62%. This indicates that at least on this data there was little degradation in performance using C_0, no CMN, and simple differences, compared to normalised log-energy, CMN, and linear regression delta and delta-delta parameters

[2] There is a difference in the way that the SNRs were calculated in the two systems. For the RM data the SNR was calculated on a per-sentence basis, using the mean of the speech and noise (utilising the NIST "wavmd" software). However on the WSJ database the max of the speech was calculated over the whole of the test set. This resulted in an approximate 6dB difference in SNR, thus the 18dB noise level in table 9.1 was, on a mean speech to noise level, approximately 12dB.

simpler approximation showed comparable performance to the DPMC mean-compensated system.

From the development results, there was little to gain, indeed there may be a slight degradation in performance, using variance compensation at high SNRs. However, at lower SNRs slight gains in performance were observed performing variance compensation.

Model Set	SNR (dB)				
	Clean	30	24	18	12
Continuous-Time	8.3	12.4	15.5	23.5	42.7

Table 9.2: Word error rates (%) of the IBM system on Car3 ARPA 1994 CSRNAB Spoke 10 development data.

As a comparison with other systems, table 9.2 shows the performance of the IBM system, which used the Continuous-Time approximation [31]. Even taking into account the slight bias introduced by optimising on a superset of the S10 development data, the PMC performance showed a lower error rate than the IBM system, particularly at lower SNRs.

9.3 Results on the Evaluation Data

The ARPA 1994 CSRNAB Spoke 10 evaluation data consists of 113 sentences spoken by 10 different speakers, again recorded in clean conditions. Noise from a single noise source was added at three SNRs, labelled Level 1, Level 2 and Level 3, the actual SNRs were unknown a-priori[3]. The results presented here are not the ones submitted for the evaluation itself, owing to a software problem with the evaluation system [76].

Model Set	Noise Condition			
	Clean	Level 1	Level 2	Level 3
Clean	6.7	35.0	20.4	59.3
Diagonal	—	10.3	9.4	15.3
Diagonal (Mean)	—	8.5	7.8	11.2
Log-Add*	—	8.4	7.8	11.2

Table 9.3: Word error rates (%) of PMC-compensated model sets on ARPA 1994 Spoke 10 evaluation data, * indicates that the results were generated by re-scoring the Diagonal mean-compensated lattices.

The results of the various systems are shown in table 9.3. These showed a far slower degradation in performance as the SNR dropped than the development data. For the worst noise condition, the mean-compensated system's performance had only dropped by 64% compared to the development data where at 18dB the performance had dropped by

[3]The official SNRs quoted for the noise conditions are Level 1 16dB, Level 2 22dB and Level 3 10dB. These SNR were calculated using mean speech and noise values by NIST so cannot be directly compared to the development data SNRs.

190%. For all the evaluation noise conditions the mean-and-variance-compensated system performed far worse than the mean-compensated system. The level of degradation in performance when both means and variances are compensated was far larger than that observed in the development data results. The performance of the Log-Add approximation was approximately the same as the DPMC mean-compensated system over all the noise conditions.

To investigate the failing of the variance compensation on both the evaluation and, to a lesser extent, the development data, a matched Level 3 evaluation data system was built by adding the adaptation noise data to all the WSJ training data. A set of models was then built on this corrupted-speech data, using the frame/state component alignment of the clean speech. The Level 3 noise condition was chosen as it shows the largest disparity in performance. Since this matched system should be the same as the compensated system, assuming that the models were well compensated, it was possible to compare the performance of the compensation scheme with the matched system in detail.

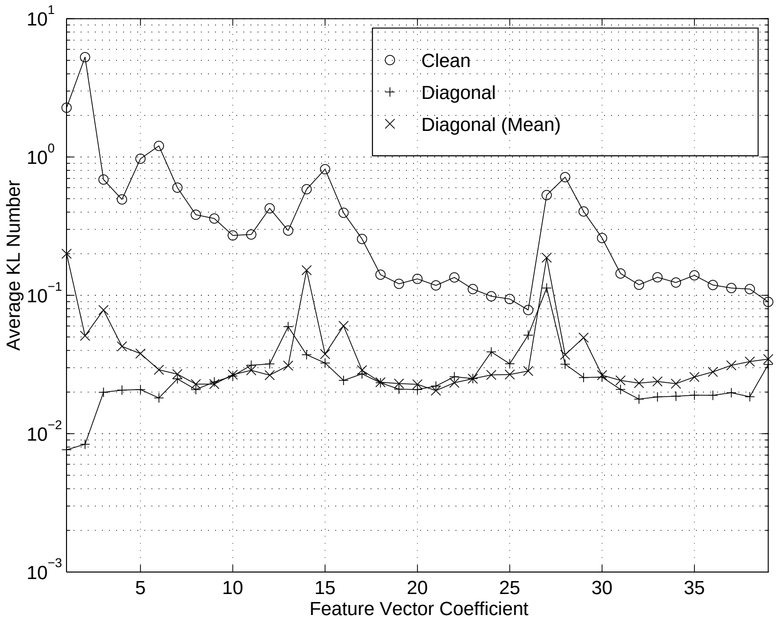


Figure 9.1: Average KL number for clean, DPMC mean-and-variance-compensated and mean-compensated model sets on Level 3 ARPA Spoke 10 evaluation noise

Figure 9.1 shows the average KL number between the single-pass matched system and each of the mean-compensated and the mean-and-variance-compensated systems. Feature vector coefficients 1 to 13 are C_0 to C_{12}, 14 to 26 the respective delta parameters and 27 to 39 the delta-delta parameters. The same general trends were observed as in similar RM systems. The static parameters were more affected by the addition of noise, with the lower-order Cepstra being shifted more than the higher-order Cepstra for the clean system. Comparing the performance of the mean-compensated system and the mean-and-variance-compensated system showed that, at least in terms of the average KL number, it was slightly better to compensate the means and the variances.

The word error rate for the single-pass system is shown in the first column of table 9.4. This shows the surprising result that even when training a system under matched con-

Model Set	Level 3 SNR	
	Evaluation	Adapt. Noise
Diagonal	15.3	14.3
Diagonal (Mean)	11.2	11.4
Single-Pass	12.1	12.3
Single-Pass (Mean)	10.7	11.4

Table 9.4: Word error rates (%) of PMC-compensated and matched model sets on the ARPA 1994 CSRNAB Spoke 10 evaluation data and adaptation-noise-corrupted evaluation data.

ditions, using the clean speech variances gave better performance. There are a number of possible explanations for this. As observed in previous work, the addition of noise to speech tends to reduce the variances in the Log-Spectral domain. This shrinking of the variances may make the system less robust to noise mismatches, such as differences between the noise data used to adapt the models and the actual noise corrupting the speech. Alternatively, there may be large differences between the speakers in the training data and the evaluation data. This may be in the form of a simple gain difference (no gain compensation was employed in this implementation of DPMC), or as a more general difference in speaker styles.

To investigate whether the poor performance was due to mismatches in the additive noise, a new set of test sentences were generated by adding the noise used to obtain the compensation noise models. The results for this new test set are also shown in table 9.4, labelled *Adapt. Noise*. Again, no advantage was gained by compensating the variances for the data generated on the noise adaptation data. This indicates that the problem was not due to large variations in the background noise conditions.

The second possible cause of mismatch was speaker variability. The standard HTK WSJ system used CMN to generate the speech parameters. This may be viewed as a simple speaker-adaptation scheme. The standard deviation of the word error rate per-speaker on the clean speech using this standard HTK system was 2.4 with a word error rate of 5.8%. With the system described in this work this figure rose to 3.9 with a word error rate of 6.7%, indicating that part of the performance drop may be due to outlier speakers. Furthermore, on the Level 3 evaluation data the standard deviation of the word error rate per speaker was 5.1 without variance compensation, this rose to 9.6 with variance compensation. Again, this indicates that the degradation in performance may be dominated by a few speakers with high error rates.

From the previous experiments on the RM database, adding an offset to each of the clean static means acted as a very simple form of speaker adaptation. The offset was estimated in the same way as the convolutional noise component. Table 9.5 shows the performance of such a scheme on the ARPA 1994 CSRNAB Spoke 10 evaluation data. Two offset, speaker adaptation, estimation schemes were investigated. The first used ten test sentences from each speaker to estimate the offset and was only examined on the worst, Level 3, noise condition. This was not acceptable for an evaluation result, but does indicate the performance possible with the simple offset adaptation scheme. Both the mean-compensated and mean-and-variance-compensated model sets showed improvement over the no speaker-adaptation schemes. For the mean-and-variance-compensated model

Model Set	Adapt Sent.	Noise Condition		
		Level 1	Level 2	Level 3
Diagonal	$10^\dagger$	—	—	10.0
Diagonal (Mean)	$10^\dagger$	—	—	10.3
Diagonal	1^*	7.8	7.2	10.3
Diagonal (Mean)	1^*	7.7	6.9	10.7
Extended	1^*	7.5	7.2	10.1
Log-Add	1^*	—	—	10.7

Table 9.5: Word error rates (%) of PMC-compensated model sets with simple speaker adaptation, using either 10 or 1 adaptation sentences, on ARPA 1994 CSRNAB Spoke 10 evaluation data, * indicates that tilt compensation was performed on a speaker level and † indicates the use of the iterative scheme.

set a reduction of 35% reduction in error rate was achieved. Furthermore, the use of variance compensation showed a slight improvement over the mean-compensated system.

The second offset estimation scheme used only the first test sentence and the iterative convolutional noise estimation scheme, it was therefore an acceptable evaluation system. The performance here was slightly worse than using the first ten sentences, but showed improvement over the no-adaptation case. Again, on the Level 3 noise compensating the variance slightly improved the performance. In terms of the standard deviation of the word error rate per speaker, the mean-and-variance-compensated system was at 4.9, the mean-compensated system at 4.9. These were both lower than using no adaptation. This second scheme was examined at all three noise conditions and showed improvement over the no adaptation model set for all cases. Little difference in performance was observed between the Extended covariance approximation and the Diagonal covariance approximation systems, though a slight improvement was observed at the two lower SNRs. On this more complex task, there appears to be little gain from improving the correlation modelling. Finally, the performance of the Log-Add approximation on the Level 3 noise condition was examined. As expected from the RM and development results, the performance was approximately the same as the mean-compensated system.

9.4 Discussion

This chapter has described the application of a number of PMC-based schemes to a large vocabulary speech recognition task. This larger task highlighted some interesting points. In contrast to previous experiments on a medium-size vocabulary task, compensating the variances of the models did not improve the performance. Moreover, at higher SNR it degraded the performance if the standard PMC additive noise compensation scheme was used.

When the variances of the corrupted models were not compensated the system was found to show good noise robustness. At the worst noise condition, the uncompensated performance was 59.3% compared with a compensated performance of 11.2%. Use of the Log-Add approximation was found to give comparable performance to DPMC when the variances were not compensated.

The performance of a simple speaker-adaptation scheme was examined, whereby the speaker differences were modelled as a simple bias in the Log-Spectral domain. This reduced the error rate for both the mean-compensated and mean-and-variance-compensated systems. This improvement in performance using a simple speaker-adaptation scheme agreed with results obtained on the RM task. The error rates for both the systems after adaptation were approximately the same. This implied that at least some of the degradation in the performance of the mean-and-variance-compensated system was due to speaker variability, however there was still little gain in compensating the variances.

Without further experiments it is hard to state whether the results obtained on the evaluation data are indicative of the general performance of model-based schemes such as PMC. The evaluation experiments were performed on a test set of ten speakers, uttering approximately ten sentences each. Previous work on RM, where compensating the variances was found to improve the performance, used up to 40 speakers each uttering 30 sentences. The DPMC scheme implemented here did not alter the frame/state component alignment within a state, experiments on RM indicate that this improves performance. Furthermore, using additional components to model the non-Gaussian nature of the corrupted speech distribution also improved performance on the RM task and could improve performance on this more difficult task. In these cases, variance compensation may show improvements.

System	Compensation Enabled	Noise Condition			
		Clean	Level 1	Level 2	Level 3
IBM	$\times$	7.2	42.2	15.4	77.4
	$\surd$	—	10.0	8.4	12.8
SRI	$\times$	6.7	18.4	11.1	35.4
	$\surd$	—	9.8	8.4	12.2
Std. HTK	$\times$	5.8	30.0	14.7	54.3*
HTK C_0	$\times$	6.7	35.0	20.4	59.3
	$\surd$	—	7.5	7.2	10.1

Table 9.6: Word error rates (%) of other sites on Car1 WSJ Spoke 10 1994 evaluation data, * indicates that one sentence failed to give a result and resulted in all deletions for that sentence.

It is hard to make a fair comparison with the other systems submitted for the S10 spoke, as results presented here were produced after the official evaluation date. However, the results presented here were not optimised on the test data, and used the grammar scale factor and insertion penalty determined for the clean system on the S0 development data. The "official" results [76] from SRI [68] and IBM [31] are shown in table 9.6. The SRI system was based on probabilistic optimal filtering and linear regression model adaptation. The IBM system used the PMC Continuous-Time approximation to compensate all the means. There were some implementation differences between the IBM system and the one described here. Since the Continuous-Time approximation was used, IBM used linear regression delta and delta-delta parameters. Additionally, a per-sentence gain to match the speech levels was estimated and the feature vector used was 72 dimensional. Note that the HTK C_0 results in table 9.6 are the simple one-sentence speaker adaptation results, compensating both the means and the variances using the Extended covariance approximation.

Initially looking at the clean performance on the evaluation, the standard HTK system had a far lower error rate than all the other systems, 15% lower than the SRI clean performance and the HTK C_0 system implemented here. The difference in performance between the standard HTK system and the one implemented here, HTK C_0, is surprising given the very similar levels of performance on the development data. The main difference between the two systems was the use of CMN for the standard system. It is assumed that CMN is performing a very crude form of speaker adaptation, as there was no convolutional noise on this test set. It is interesting to note that the SRI system, which used CMN, achieved only the same level of performance as the HTK C_0 system without CMN. As the SNR decreased, the performance of all the clean systems degraded rapidly. However the performance of the SRI system was consistently better than either the IBM system, the standard HTK system, or HTK C_0. This is surprising as the standard HTK system used the same front-end and achieved better performance on the clean data. In terms of noise robustness with the compensation enabled, all the systems, SRI, IBM and DPMC, perform approximately the same, though the DPMC compensated system consistently performed slightly better than either of the other two schemes.

Overall this chapter has shown that PMC and DPMC techniques may be applied successfully to a large, 5000 word, vocabulary task. In contrast to the RM task, variance compensation was found not to dramatically improve performance, indeed without simple speaker-adaptation it degraded performance. Comparing the performance of DPMC to published evaluation results showed that DPMC could achieve state-of-the-art performance.

Chapter 10

Conclusions and Future Work

This thesis has detailed the development of a model-based noise compensation technique, Parallel Model Combination (PMC). The basic aim is to alter the parameters of a set of HMMs trained in a clean environment to be representative of a system trained in some noise-corrupted environment. The new noise environment may differ from the clean environment due to additive and/or convolutional noise. For PMC, it is necessary to have a clean model set, a model that is representative of the ambient background noise and, in the case of additive and convolutional noise, a small amount of adaptation data, to estimate the new corrupted-speech models. Theory has been derived to allow all the parameters, other than the transition probabilities, of a continuous density HMM-based speech recognition system, to be compensated.

There is no simple exact closed form for the full compensation process, so various approximations have been described. Numerical integration based techniques using Gaussian integration may be used to obtain estimates of the corrupted-speech means and variances. Alternatively, faster, simpler, estimation schemes may be used. For static parameters, closed-form solutions for the compensation are available if a Log-Normal assumption is used. An even simpler approximation, the Log-Add approximation, whereby only the means of the system are compensated, may be used to compensate all the parameters. All these schemes are non-iterative and make the assumption that the frame/state component alignment is not altered by the addition of noise. This is known to be false, however, good noise robustness was nevertheless achieved using these schemes.

An iterative scheme that allows the frame/state component alignment within a state to vary was also described, Data-driven Parallel Model Combination (DPMC). Allowing the frame/state component alignment to vary alters the components weights and allows the corrupted-speech distribution to be better modelled. In addition the number of components in the compensated model set may be chosen independently of the number of components in the clean speech and noise models.

The performance of the compensation systems were assessed in two different ways. Firstly it was necessary to discover how close the compensated model sets were to models trained in the new acoustic environment. To this end, single-pass training was used to generate noise-corrupted model sets and an average KL number distance measure was performed on each element of the feature vector. In addition, the standard word error rate was used to assess the recognition performance of the compensated systems.

10.1 Summary of Results

Three different databases were chosen to evaluate the technique. The first, the NOISEX-92 database is a small vocabulary task, the digits, in low SNR conditions. For this small vocabulary task it was only necessary to use static parameters to achieve no errors in the clean environment. When additive noise was introduced, the performance rapidly degraded, until at 0dB the word error rate was 83%. Single-pass training improved this figure to 4%. Using PMC-based techniques the clean model set was transformed to be "close", in terms of the average KL number, to the single pass noise-corrupted models. Recognition performance was also comparable with the matched systems, achieving 2% word error rate at 0dB SNR. In addition to the additive noise task, an additive and convolutional noise task was supplied, allowing a more general noise environment to be investigated. The convolutional noise was found to degrade performance to above 50% word error rate at 0dB SNR. Using limited amounts of data, down to 2 digits, under 10% word error rate was achieved at 0dB when both additive and convolutional noise were present.

Having established that PMC may be successively applied to compensating static parameters in extreme noise conditions, the problem of compensating the dynamic coefficients was addressed. Since NOISEX-92 was too simple to require the use of dynamic coefficients, the DARPA Resource Management (RM) database was selected. This is a medium-size vocabulary task, 1000 words, and is widely distributed. On this harder task it was necessary to incorporate dynamic coefficients into the feature vector to achieve good performance. For the additive noise case it was shown that the matched single-pass trained system performed best when both the means and variances of all the model parameters (i.e. static, delta and delta-delta) were compensated. The need to compensate the dynamic coefficients introduces a new problem into the PMC process, what parameters need to be stored for the clean speech and noise models? Increasing the covariance correlation modelling, whilst still constraining the corrupted-speech models' covariance matrices to be diagonal, was found to improve performance. For this task, using either a "Pseudo" matrix, or a quin-diagonal covariance matrix to model the correlation between feature element sets, such as between the static and the delta parameters, was found to yield the best performance. The full covariance case was not examined. All the experiments described so far involved non-iterative PMC, where the frame/state component alignment is assumed to be unaltered by the addition of noise. Further improvements in performance were found by allowing this alignment to vary within the state. When additive and convolutional noise were both present, the compensation scheme was found to outperform the standard additive noise system if the spectral mismatch was estimated on a per-speaker level. This shows the gain of using speaker adaptation, albeit simple, in noise compensation techniques.

The final database examined was the ARPA 1994 CSRNAB S10 noise spoke. This is a large vocabulary task, 5000 words. The results on this task were mixed. Mean compensation, either using DPMC or the Log-Add approximation, gave good noise robustness. Unlike both the RM and NOISEX-92 systems, however, it was found that compensating the variances degraded performance. The use of simple speaker adaptation, in the form of a bias on the static means, was found to improve performance, particularly when the means and variances were compensated. However even in this case little gain was observed when the variances were compensated. Due to computational limitations, it was only possible to examine non-iterative PMC techniques on this task.

In summary, PMC and its various implementations, are capable of achieving state-of-the-art performance on a wide variety of additive and convolutional noise tasks. Though the techniques have currently only been examined on artificial tasks, it is felt that good performance can be achieved in real world conditions.

10.2 Future Work

To date, the majority of the work on PMC and related techniques has concentrated on the additive noise task. Simple convolutional noise estimation techniques have been developed, but these are felt to be far from optimal. In particular, techniques for rapidly estimating the spectral tilt differences on very little data are required. Also, as highlighted by the RM and ARPA CSRNAB results, improvements can be gained using simple speaker adaptation. Incorporating more complex adaptation techniques, such as MLLR [49] is highly desirable. In contrast to the use of linear regression adaptation to model the complete acoustic environment, here it would be used for "clean' speech adaptation. Thus, the same speaker adaptation parameters may be used in a wide variety of noise conditions.

Iterative PMC has been found to improve performance, particularly at the lower SNRs. Unfortunately, there is a computational overhead associated with its implementation. The various play-offs, such as the number of points in the DPMC process and number of iterations of Baum-Welch used, has not been investigated for these iterative PMC cases. Overall, there is a need for faster PMC approximations, both iterative and non-iterative. Whether this will result from new implementations, or from speeding up present implementations needs to be investigated. Of particular interest is whether variance compensation, which is computationally expensive, yields sufficient improvements in performance to justify its use.

All the experiments presented here use diagonal covariance matrices for the corrupted-speech distributions. It is likely that the DCT in additive noise conditions is less appropriate with diagonal covariance matrices than in clean conditions. The use of full covariance matrices for corrupted speech needs to be examined to see whether the gains in performance outweigh the additional run time computational cost and the memory requirements.

The performance of PMC in complex-noise environments has been touched upon, but not extensively investigated. Initial experiments in altering the number of Gaussian components within a state have been performed [28], but further work is required. The question as to whether simple single-state models are sufficient for real applications also needs to be addressed.

Presently, PMC has only been applied to databases with noise artificially added, thus eliminating the Lombard effect. In real systems, particularly at low SNRs, the Lombard effect may seriously degrade performance. Compensating for this effect within PMC is possible. However, it requires statistics about the effect of noise on speakers. This effect may be possible to model as a simple offset to the Cepstral parameters. In addition, simplifying assumptions have been made about the form that the convolutional noise can take. For all the tasks investigated, the convolutional noise was stationary with time. Whether this simple approximation is appropriate for real microphones needs to be investigated.

Moving away from the direct applications of PMC for speech recognition in noise, the framework developed could be applicable to training models in situations where the data has been collected in a variety of noise conditions. Previous work in the area [45, 81] used the *max()* approximation in the Log-Spectral domain and did not consider re-estimation of the dynamic coefficients. Working in the Cepstral domain and estimating the dynamic parameters would allow a state-of-the-art recogniser to be trained. This gives a greater degree of flexibility to the training data that can be used.

Appendix A

Derivation of the Delta and Delta-Delta Mismatch Function

For the mismatch functions derived in this work, the delta parameters, $\mathbf{\Delta O}^c(\tau)$, are assumed to be calculated using simple differences. Thus

$$\begin{aligned}\mathbf{\Delta O}^c(\tau) &= \mathbf{O}^c(\tau+w) - \mathbf{O}^c(\tau-w) \\ &= \mathbf{C}(\mathbf{O}^l(\tau+w) - \mathbf{O}^l(\tau-w))\end{aligned} \tag{A.1}$$

where $\mathbf{O}^c(\tau)$ is the "static" observation vector at time τ. Applying the assumption that the speech and noise are additive in the Linear Spectral yields

$$\begin{aligned}\mathbf{\Delta O}^c(\tau) &= \mathbf{C}\log\left[(\mathbf{O}(\tau+w))./(\mathbf{O}(\tau-w))\right] \\ &= \mathbf{C}\log\left[(g\mathbf{S}(\tau+w)+\mathbf{N}(\tau+w))./(g\mathbf{S}(\tau-w)+\mathbf{N}(\tau-w))\right]\end{aligned} \tag{A.2}$$

This may then be re-expressed to give

$$\begin{aligned}\Delta O_i^l(\tau) &= \log\left(\frac{\exp(\Delta S_i^l(\tau)+S_i^l(\tau-w)+g^l-N_i^l(\tau-w))+\exp(\Delta N_i^l(\tau))}{\exp(S_i^l(\tau-w)+g^l-N_i^l(\tau-w))+1}\right) \\ &= \log\Big(\exp(\Delta S_i^l(\tau)+S_i^l(\tau-w)+g^l)+\exp(\Delta N_i^l(\tau)+N_i^l(\tau-w))\Big) \\ &\quad -\log\Big(\exp(S_i^l(\tau-w)+g^l)+\exp(N_i^l(\tau-w))\Big)\end{aligned} \tag{A.3}$$

where $g^l = \log(g)$.

If the delta-delta parameters, $\mathbf{\Delta^2 O}^c(\tau)$, are calculated using simple differences then

$$\mathbf{\Delta^2 O}^c(\tau) = \mathbf{\Delta O}^c(\tau+w_a) - \mathbf{\Delta O}^c(\tau-w_a) \tag{A.4}$$

In the Linear-Spectral domain[1]

$$\Delta^2 O_i(\tau) = \frac{\Delta O_i(\tau+w_a)}{\Delta O_i(\tau-w_a)} = \frac{O_i(\tau+w_a+w)O_i(\tau-w_a-w)}{O_i(\tau-w_a+w)O_i(\tau+w_a-w)} \tag{A.5}$$

Expanding this expression in terms of the static parameters, where the noise is assumed to be additive, and additionally letting the window for the delta parameters be the same as the delta-delta parameters, $w_a = w$, yields

$$\Delta^2 O_i(\tau) = \frac{(S_i(\tau+2w)+N_i(\tau+2w))(S_i(\tau-2w)+N_i(\tau-2w))}{(S_i(\tau)+N_i(\tau))^2} \tag{A.6}$$

[1] g has been dropped for clarity.

In the same fashion as the mismatch function for the delta parameters was derived, this expression needs to be rewritten in terms of known statistics.

$$\Delta^2 O_i(\tau) = \left(\frac{S_i(\tau)N_i(\tau)}{S_i(\tau)+N_i(\tau)}\right)^2 \left[\frac{\Delta^2 S_i(\tau)}{N_i(\tau)^2} + \frac{\Delta^2 N_i(\tau)}{S_i(\tau)^2} + \right. \\ \left. \frac{\Delta^2 N_i(\tau)\Delta N_i(\tau-w)}{\Delta S_i(\tau-w)S_i(\tau)N_i(\tau)} + \frac{\Delta^2 S_i(\tau)\Delta S_i(\tau-w)}{\Delta N_i(\tau-w)S_i(\tau)N_i(\tau)}\right] \quad \text{(A.7)}$$

Finally expressing this equation in the Log-Spectral domain

$$\Delta^2 O_i^l(\tau) = \log(\exp(\Delta^2 S_i^l(\tau) + 2(S_i^l(\tau) - O_i^l(\tau))) + \exp(\Delta^2 N_i^l(\tau) + 2(N_i^l(\tau) - O_i^l(\tau))) + \\ \exp(\Delta^2 N_i^l(\tau) + \Delta N_i^l(\tau-w) - \Delta S_i^l(\tau-w) + S_i^l(\tau) + N_i^l(\tau) - 2O_i^l(\tau)) + \\ \exp(\Delta^2 S_i^l(\tau) + \Delta S_i^l(\tau-w) - \Delta N_i^l(\tau-w) + S_i^l(\tau) + N_i^l(\tau) - 2O_i^l(\tau))) \quad \text{(A.8)}$$

Though complex, this expression is the sum of a set of exponentials, similar to the static and the delta parameters.

Appendix B

Numerical Integration

Parameter estimation may be performed using numerical integration. If implemented directly this would involve a two-dimensional integration to estimate the mean and a four-dimensional integration to estimate the covariance matrix. This would be computationally very expensive, so a more efficient form for the integration is required. Firstly it is necessary to select an appropriate numerical integration approximation. Given the form of the integration over Gaussian probability distributions, the numerical integration approximation used is

$$\int_{-\infty}^{\infty} f(x)\exp(-x^2)\,dx = \sum_{i=1}^{I} w_i f(x_i) \tag{B.1}$$

where x_i and w_i are given by the Gauss-Hermite abscissa and weights respectively. For simplicity of notation single Gaussian mixture distributions will be considered. The extension to multiple mixture Gaussian distributions is trivial.

Initially examining the estimation of the corrupted mean, rewriting

$$\mathcal{E}\{\log(\exp(S_i^l(\tau)) + \exp(N_i^l(\tau)))\} = \mathcal{E}\{N_i^l(\tau) + \log(\exp(S_i^l(\tau) - N_i^l(\tau)) + 1)\} \tag{B.2}$$

and noting that the sum or difference of two Gaussian distributed variables is itself Gaussian distributed then

$$\hat{\mu}_i^l = \tilde{\mu}_i^l + \int_{-\infty}^{\infty} \mathcal{N}(x; \mu_i^l - \tilde{\mu}_i^l, \Sigma_{ii}^l + \tilde{\Sigma}_{ii}^l)\log(\exp(x) + 1)\,dx \tag{B.3}$$

where x denotes $S_i^l - N_i^l$. It is a simple transformation to convert the above equation to the correct form.

Having dealt with the mean, it is necessary to obtain the covariance estimate.

$$\begin{aligned}
&\mathcal{E}\{\log(\exp(S_i^l(\tau)) + \exp(N_i^l(\tau)))\log(\exp(S_j^l(\tau)) + \exp(N_j^l(\tau)))\} = \\
&\quad \mathcal{E}\{N_i^l(\tau)N_j^l(\tau) + N_i^l(\tau)\log(\exp(S_j^l(\tau) - N_j^l(\tau)) + 1) + N_j^l(\tau)\log(\exp(S_i^l(\tau) - N_i^l(\tau)) + 1) \\
&\qquad + \log(\exp(S_i^l(\tau) - N_i^l(\tau)) + 1)\log(\exp(S_j^l(\tau) - N_j^l(\tau)) + 1)\}
\end{aligned} \tag{B.4}$$

The expected value of the above expression is required. When dealing with the leading diagonal terms the above expression may be simplified to

$$\mathcal{E}\{(O_i^l)^2\} = \mathcal{E}\{(N_i^l)^2 + 2N_i^l\log(\exp(S_i^l - N_i^l) + 1) + \Big(\log(\exp(S_i^l - N_i^l) + 1)\Big)^2\} \tag{B.5}$$

The first term is known. The second term is a directly implementable two dimensional integration with variables N_i^l and S_i^l. The final term is a single dimension integration having variable $S_i^l - N_i^l$. For the off-diagonal terms examining the terms individually.

$$\mathcal{E}\{N_i^l(\tau)N_j^l(\tau)\} = \tilde{\Sigma}_{ij}^l + \tilde{\mu}_i^l\tilde{\mu}_j^l \tag{B.6}$$

by definition. Examining the second term

$$\mathcal{E}\{N_i^l(\tau)\log(\exp(S_j^l(\tau) - N_j^l(\tau)) + 1)\} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} N_i^l \log(\exp(x) + 1)\mathcal{L}(x, N_i^l)\, dx\, dN_i^l \tag{B.7}$$

where x denotes $S_j^l - N_j^l$. The speech and the noise are known to be uncorrelated. However, the various noise channels may be correlated depending on the form of the noise modelling. Thus $\mathcal{L}(x, N_i^l)$ may have a full covariance matrix associated with it. This covariance matrix will be

$$\mathbf{\Sigma} = \begin{bmatrix} \Sigma_{ii}^l + \tilde{\Sigma}_{ii}^l & -\tilde{\Sigma}_{ij}^l \\ -\tilde{\Sigma}_{ij}^l & \tilde{\Sigma}_{jj}^l \end{bmatrix} \tag{B.8}$$

and the mean

$$\mu = \begin{bmatrix} \mu_i^l - \tilde{\mu}_i^l & \tilde{\mu}_j^l \end{bmatrix}^T \tag{B.9}$$

This new feature vector space may then be rotated to diagonalise $\mathbf{\Sigma}$ and then standard numerical integration performed. The third term is of the same form as the second term. Examining the final term will yield an expression to be integrated over the variables $S_i^l - N_i^l$ and $S_j^l - N_j^l$. The covariance matrix for such an integration is given by

$$\mathbf{\Sigma} = \begin{bmatrix} \Sigma_{ii}^l + \tilde{\Sigma}_{ii}^l & \Sigma_{ij}^l + \tilde{\Sigma}_{ij}^l \\ \Sigma_{ij}^l + \tilde{\Sigma}_{ij}^l & \Sigma_{jj}^l + \tilde{\Sigma}_{jj}^l \end{bmatrix} \tag{B.10}$$

and the mean by

$$\mu = \begin{bmatrix} \mu_i^l - \tilde{\mu}_i^l & \mu_j^l - \tilde{\mu}_j^l \end{bmatrix}^T \tag{B.11}$$

Again the feature vector space may be rotated to allow integration over two independent variables. By breaking the integration up into four separate components it is possible to perform a series of two dimensional integrations, as opposed to performing a direct four dimensional integration.

The dynamic coefficient parameter estimation may be performed using a similar approach. The expression for the corrupted delta mean

$$\Delta\hat{\mu}_i^l = \Delta\tilde{\mu}_i^l - \mathcal{E}\{\log(\exp(S_i^l - N_i^l) + 1)\} + \mathcal{E}\{\log(\exp(S_i^l + \Delta S_i^l - N_i^l - \Delta N_i^l) + 1)\} \tag{B.12}$$

Initially assume that the delta and the static parameters are independent. The first term is known, the delta mean of the noise. The second term is a function of the variable $S_i^l - N_i^l$, which is known to be Gaussian distributed. The final term is solely a function of $S_i^l + \Delta S_i^l - N_i^l - \Delta N_i^l$, which is, again, known to be Gaussian distributed. The estimation of the mean is therefore the sum of two one dimensional integrations. Like the static parameter compensation, the integration of the function of a Gaussian distributed variable is performed using Gaussian integration with Gauss-Hermite abscissa and weights.

The estimation of the covariance matrix may be obtained using a similar identity.

$$\begin{aligned}
\Delta\hat{\Sigma}^l_{ij} \quad = \quad & \Delta\tilde{\Sigma}^l_{ij} + \Delta\tilde{\mu}^l_i \Delta\tilde{\mu}^l_j + \mathcal{E}\{\log(\exp(S^l_i - N^l_i) + 1)\log(\exp(S^l_j - N^l_j) + 1)\} \\
& + \mathcal{E}\{\log(\exp(S^l_i + \Delta S^l_i - N^l_i - \Delta N^l_i) + 1)\log(\exp(S^l_j + \Delta S^l_j - N^l_j - \Delta N^l_j) + 1)\} \\
& - \mathcal{E}\{\Delta N^l_i \log(\exp(S^l_j - N^l_j) + 1)\} \\
& + \mathcal{E}\{\Delta N^l_i \log(\exp(S^l_j + \Delta S^l_j - N^l_j - \Delta N^l_j) + 1)\} \\
& - \mathcal{E}\{\Delta N^l_j \log(\exp(S^l_i - N^l_i) + 1)\} \\
& + \mathcal{E}\{\Delta N^l_j \log(\exp(S^l_i + \Delta S^l_i - N^l_i - \Delta N^l_i) + 1)\} \\
& - \mathcal{E}\{\log(\exp(S^l_i + \Delta S^l_i - N^l_i - \Delta N^l_i) + 1)\log(\exp(S^l_j - N^l_j) + 1)\} \\
& - \mathcal{E}\{\log(\exp(S^l_i - N^l_i) + 1)\log(\exp(S^l_j + \Delta S^l_j - N^l_j - \Delta N^l_j) + 1)\} - \Delta\hat{\mu}^l_i \Delta\hat{\mu}^l_j
\end{aligned} \tag{B.13}$$

Though this expression appears to be highly complex, it is simply the sum of a set of two dimensional numerical integration functions.

If the static and delta parameters are not assumed to be independent then it is necessary to include additional cross terms in the covariance matrices.

The same approach may be adapted to estimating the delta-delta parameters.

Appendix C

Derivation of Multi-Variate Gaussian to Log-Normal Mapping

Mathematically the Log-Normal approximation gives a distribution that has the same first two moments as the combined speech and noise distributions in the Linear-Spectral domain. Given the distributions in the Log-Spectral domain are Gaussian and known, the mean, the first moment, may be calculated from

$$\begin{aligned}
\mu_i &= \mathcal{E}\{S_i\} = \mathcal{E}\left\{\exp(S_i^l)\right\} \\
&= \int_{\mathcal{R}^n} \frac{1}{(\sqrt{2\pi})^n |\Sigma^l|^{\frac{1}{2}}} \exp(S_i^l - \frac{1}{2}(\mathbf{S}^l - \mu^l)^T (\Sigma^l)^{-1} (\mathbf{S}^l - \mu^l))\, d\mathbf{S}^l \\
&= k \int_{\mathcal{R}^n} \exp(-\frac{1}{2}\mathcal{F}(\mathbf{S}^l, \Sigma^l, \mu^l))\, d\mathbf{S}^l \qquad \text{(C.1)}
\end{aligned}$$

where $\mathcal{R}^n$ is the region of all possible values of $\mathbf{S}^l$, the observed vector and

$$k = \frac{1}{(\sqrt{2\pi})^n |\Sigma^l|^{\frac{1}{2}}} \qquad \text{(C.2)}$$

Letting $\mathbf{e}^i$ be the i^{th} row of the n by n identity matrix and $\mu_n = \mu^l + \mathbf{e}^i \Sigma^l$. We can complete the square in $\mathcal{F}(\mathbf{S}^l, \Sigma^l, \mu^l)$ by considering $(\mathbf{S}^l - \mu_n)$.

$$\begin{aligned}
&(\mathbf{S}^l - \mu_n)^T (\Sigma^l)^{-1} (\mathbf{S}^l - \mu_n) \\
&= (\mathbf{S}^l - \mu^l)^T (\Sigma^l)^{-1} (\mathbf{S}^l - \mu^l) - 2(\mathbf{S}^l)^T (\Sigma^l)^{-1} (\mathbf{e}^i \Sigma^l) + 2(\mathbf{e}^i)^T \Sigma^l (\Sigma^l)^{-1} \mu^l \\
&\quad + (\mathbf{e}^i \Sigma^l)^T (\Sigma^l)^{-1} \mathbf{e}^i \Sigma^l \\
&= \mathcal{F}(\mathbf{S}^l, \Sigma^l, \mu^l) + 2(\mathbf{e}^i)^T \mu^l + (\mathbf{e}^i)^T \Sigma^l \mathbf{e}^i \\
&= \mathcal{F}(\mathbf{S}^l, \Sigma^l, \mu^l) + 2\mu_i^l + \Sigma_{ii}^l \qquad \text{(C.3)}
\end{aligned}$$

Noting that

$$\int_{\mathcal{R}^n} \exp(-\frac{1}{2}(\mathbf{S}^l - \mu_n)^T (\Sigma^l)^{-1} (\mathbf{S}^l - \mu_n))\, d\mathbf{S}^l = (\sqrt{2\pi})^n |\Sigma^l|^{\frac{1}{2}} = \frac{1}{k} \qquad \text{(C.4)}$$

Equation C.1 may be simplified to

$$\mu_i = \exp(\mu_i^l + \Sigma_{ii}^l/2) \qquad \text{(C.5)}$$

The variance may similarly be calculated.

$$\mathcal{E}\left\{\exp(S_i^l)\exp(S_j^l)\right\} = \int_{\mathcal{R}^n} \frac{1}{(\sqrt{2\pi})^n|\mathbf{\Sigma}^l|^{\frac{1}{2}}} \exp(S_i^l + S_j^l - \frac{1}{2}(\mathbf{S}^l - \mu^l)^T(\mathbf{\Sigma}^l)^{-1}(\mathbf{S}^l - \mu^l))d\mathbf{S}^l \tag{C.6}$$

By considering $\mu_n = \mu^l + (\mathbf{e}^i + \mathbf{e}^j)\mathbf{\Sigma}^l$ and completing the square as before the above equation may be simplified to

$$\mathcal{E}\left\{\exp(S_i^l)\exp(S_j^l)\right\} = \mu_i\mu_j\exp(\Sigma_{ij}^l) \tag{C.7}$$

The variance may then be shown to be

$$\begin{aligned} \Sigma_{ij} &= \mathcal{E}\left\{\exp(S_i^l)\exp(S_j^l)\right\} - \mathcal{E}\left\{\exp(S_i^l)\right\}\mathcal{E}\left\{\exp(S_j^l)\right\} \\ &= \mu_i\mu_j\left[\exp(\Sigma_{ij}^l) - 1\right] \end{aligned} \tag{C.8}$$

The same transformation will map $\{\tilde{\mu}^l, \tilde{\mathbf{\Sigma}}^l\}$ to $\{\tilde{\mu}, \tilde{\mathbf{\Sigma}}\}$. Having obtained the two distributions in the Linear-Spectral domain, using the assumption that they are additive and independent

$$\hat{\mu} = g\mu + \tilde{\mu} \tag{C.9}$$

$$\hat{\mathbf{\Sigma}} = g^2\mathbf{\Sigma} + \tilde{\mathbf{\Sigma}} \tag{C.10}$$

will give a distribution in the Linear-Spectral domain, whose first two moments are the same as the true, theoretical, noise-corrupted speech distribution.

Given the combined mean and variance, and assuming that the combined distribution is approximately Log-normally distributed, the combined distribution in the quefrency domain may be obtained by inverting the process previously described. Hence

$$\mu_i^l \approx \log(\mu_i) - \frac{1}{2}\log\left[\frac{\Sigma_{ii}}{\mu_i^2} + 1\right] \tag{C.11}$$

$$\Sigma_{ij}^l \approx \log\left[\frac{\Sigma_{ij}}{\mu_i\mu_j} + 1\right] \tag{C.12}$$

Bibliography

[1] M Abramowitz and I A Stegen, editors. *Handbook of Mathematical Functions with Formulas, Graphs and Mathematical Tables.* Dover Publications, 1965.

[2] A Acero and R M Stern. Environmental robustness in automatic speech recognition. In *Proceedings ICASSP*, pages 849–852, 1990.

[3] A Acero and R M Stern. Robust speech recognition by normalization of the acoustic space. In *Proceedings ICASSP*, pages 893–896, 1991.

[4] P Alexandre, J Boudy, and P Lockwood. Root homomorphic deconvolution schemes for speech processing in car environments. In *Proceedings ICASSP*, volume 2, pages 99–102, 1993.

[5] Y Ariki, S Mizuta, M Nagata, and T Sakai. Spoken-word recognition using dynamic features analysed by two-dimensional Cepstrum. *IEE Proceedings Communications, Speech and Vision*, pages 133–140, 1989.

[6] X Aubert, H Bourland, Y Kamp, and C J Wellekens. Improved hidden Markov models for speech recognition. *Phillips Journal of Research*, 43:254–245, 1988.

[7] L R Bahl, P V de Souza, P S Gopalkrishnan, D Nahamoo, and M A Picheny. Context dependent modelling of phones in continuous speech using decision trees. In *Proceedings DARPA Speech and Natural Language Processing Workshop*, pages 264–270, 1991.

[8] L E Baum, T Petrie, G Soules, and N Weiss. A maximisation technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Annals of Mathematical Statistics*, 41:164–171, 1970.

[9] V L Beattie and S J Young. Hidden Markov model state-based Cepstral noise compensation. In *Proceedings ICSLP*, pages 519–522, 1992.

[10] A D Berstein and I D Shallom. An hypothesized Wiener filtering approach to noisy speech recognition. In *Proceedings ICASSP*, pages 913–916, 1991.

[11] H Bourlard and N Morgan. *Continuous Speech Recognition: A Hybrid Approach.* Kluwer Academic Publishers, 1993.

[12] J S Bridle, K M Ponting, M D Brown, and A W Borrett. A noise compensation spectrum distance measure applied to automatic speech recognition. In *Proceedings IOA*, pages 307–314, 1984.

[13] P Brown. *The Acoustic-Modelling Problem in Automatic Speech Recognition.* PhD thesis, IBM T.J. Watson Research Center, 1987.

[14] S Das, R Bakis, A Nadas, D Nahamoo, and M Picheny. Influence of background noise and microphone on the performance of the IBM Tangora speech recognition system. In *Proceedings ICASSP*, volume 2, pages 71–74, 1993.

[15] S Das, A Nadas, D Nahamoo, and M Picheny. Adaptation techniques for ambience and microphone compensation in the IBM Tangora speech recognition system. In *Proceedings ICASSP*, volume 1, pages 21–24, 1994.

[16] S B Davis and P Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions ASSP*, 28:357–366, 1980.

[17] A P Dempster, N M Laird, and D B Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39:1–38, 1977.

[18] V Digalakis and L Neumeyer. Speaker adaptation using combined transformation and Bayesian methods. In *Proceedings ICASSP*, pages 680–683, 1995.

[19] Y Ephraim. A Bayesian estimation approach for speech enhancement using hidden Markov models. *IEEE Transactions SP*, 40:725–735, 1992.

[20] Y Ephraim, D Malah, and B H Juang. On the application of hidden Markov models for enhancing noisy speech. *IEEE Transactions ASSP*, 37:1846–1856, 1989.

[21] S Furui. Speaker independent isolated word recognition using dynamic features of speech spectrum. *IEEE Transactions ASSP*, 34:52–59, 1986.

[22] S Furui. Toward robust speech recognition under adverse conditions. In *Proceedings ESCA Workshop in Speech Processing in Adverse Conditions*, pages 31–42, 1992.

[23] L Gagnon. A noise reduction approach for non-stationary additive interference. In *Proceedings ESCA Workshop in Speech Processing in Adverse Conditions*, pages 139–142, 1992.

[24] M J F Gales and S J Young. An improved approach to the hidden Markov model decomposition of speech and noise. In *Proceedings ICASSP*, pages 233–236, 1992.

[25] M J F Gales and S J Young. Cepstral parameter compensation for HMM recognition in noise. *Speech Communication*, 12:231–240, 1993.

[26] M J F Gales and S J Young. HMM recognition in noise using parallel model combination. In *Proceedings Eurospeech*, pages 837–840, 1993.

[27] M J F Gales and S J Young. PMC for speech recognition in additive and convolutional noise. Technical Report CUED/F-INFENG/TR154, Cambridge University, 1993. Available via anonymous ftp from: svr-ftp.eng.cam.ac.uk.

[28] M J F Gales and S J Young. A fast and flexible implementation of parallel model combination. In *Proceedings ICASSP*, pages 133–136, 1995.

[29] J L Gauvain and C H Lee. Improved acoustic modelling with Bayesian learning. In *Proceedings ICASSP*, pages 481–484, 1992.

[30] Y Gong. Speech recognition in noisy environments: A survey. *Speech Communication*, 16:261–291, 1995.

[31] R A Gopinath, M J F Gales, P S Gopalakrishnan, S Balakrishnan-Aiyer, and M A Picheny. Robust speech recognition in noise — performance of the IBM continuous speech recognizer on the ARPA noise spoke task. In *Proceedings ARPA Workshop on Spoken Language Systems Technology*, pages 127–130, 1995.

[32] J H L Hansen and L M Arslan. Robust feature-estimation and objective quality assessment for noisy speech recognition using the Credit Card corpus. *IEEE Transactions SAP*, 3:169–184, 1995.

[33] J H L Hansen, B D Womack, and L M Arslan. A source generator based production model for environmental robustness in speech recognition. In *Proceedings ICSLP*, pages 1003–1006, 1994.

[34] H Hermansky. Perceptual linear predictive (PLP) analysis of speech. *Journal of the Acoustical Society of America*, 87:1738–1752, 1990.

[35] H Hermansky and N Morgan. RASTA processing of speech. *IEEE Transactions SAP*, 2:578–579, 1994.

[36] J Hernando and C Nadeu. A comparative study of parameters and distances for noisy speech recognition. In *Proceedings Eurospeech*, pages 91–94, 1991.

[37] J Hernando and C Nadeu. Speech recognition in noisy car environment based on OSALPC representation and robust similarity measuring techniques. In *Proceedings ICASSP*, pages 69–72, 1994.

[38] J N Holmes and N C Sedgwick. Noise compensation for speech recognition using probabilistic models. In *Proceedings ICASSP*, pages 741–744, 1986.

[39] M J Hunt and C Lefebre. A comparison of several acoustic representations for speech recognition with degraded and undegraded speech. In *Proceedings ICASSP*, pages 262–265, 1989.

[40] B H Juang. Maximum likelihood estimation for mixture multivariate stochastic observations of Markov chains. *AT&T Technical Journal*, 64:1235–1249, 1985.

[41] B H Juang and L R Rabiner. Mixture autoregressive hidden Markov models for speech signals. *IEEE Transactions ASSP*, 33:1404–1413, 1985.

[42] B H Juang and L R Rabiner. A probabilistic distance measure for hidden Markov models. *AT&T Technical Journal*, 64:391–408, 1985.

[43] B H Juang, L R Rabiner, and J G Wilpon. On the use of bandpass liftering in speech recognition. *IEEE Transactions ASSP*, 35:947–954, 1987.

[44] J Junqua and Y Anglade. Acoustic and perceptual studies of Lombard speech: Application to isolated-word automatic speech recognition. In *Proceedings ICASSP*, pages 841–844, 1990.

[45] M Kadirkamanathan and A P Varga. Simultaneous model re-estimation from contaminated data by 'Composed hidden Markov modelling'. In *Proceedings ICASSP*, pages 897–900, 1991.

[46] D H Klatt. A digital filterbank for spectral matching. In *Proceedings ICASSP*, pages 573–576, 1979.

[47] T Kobayashi and S Imai. Spectral analysis using generalised Cepstrum. *IEEE Transactions ASSP*, 32:1087–1089, 1984.

[48] J Koehler, N Morgan, H Hermansky, H Gunter-Hirsh, and G Tong. Integrating RASTA-PLP into speech recognition. In *Proceedings ICASSP*, volume 1, pages 421–424, 1994.

[49] C J Legetter and P C Woodland. Flexible speaker adaptation using maximum likelihood linear regression. In *Proceedings ARPA Workshop on Spoken Language Systems Technology*, pages 110–115, 1995.

[50] J S Lim. Spectral root homomorphic deconvolution system. *IEEE Transactions ASSP*, 27:223–233, 1979.

[51] J S Lim and A V Oppenheim. Enhancement and bandwidth compression of noisy speech. *Proceedings IEEE*, 67:1586–1604, 1979.

[52] L A Liporace. Maximum likelihood estimation for multivariate observations of Markov sources. *IEEE Transactions Information Theory*, 28:729–734, 1982.

[53] F Liu, P J Moreno, R M Stern, and A Acero. Signal processing for robust speech recognition. In *Proceedings ARPA Workshop on Human Language Technology*, pages 309–314, 1994.

[54] P Lockwood and P Alexandre. Root adaptive homomorphic deconvolution schemes for speech recognition in noise. In *Proceedings ICASSP*, volume 1, pages 441–444, 1994.

[55] P Lockwood, J Boudy, and M Blanchet. Non-linear spectral subtraction (NSS) and hidden Markov models for robust speech recognition in car noise environments. In *Proceedings ICASSP*, pages 265–268, 1992.

[56] D Mansour and B H Juang. The short-time modified coherence representation and noisy speech recognition. *IEEE Transactions ASSP*, 37:795–804, 1989.

[57] J D Markel and A H Gray Jr. *Linear Prediction of Speech*. Springer-Verlag, 1976.

[58] F Martin, K Shikano, and Y Minami. Recognition of noisy speech by composition of hidden Markov models. In *Proceedings Eurospeech*, pages 1031–1034, 1993.

[59] B A Mellor and A P Varga. Noise masking in the MFCC domain for the recognition of speech in background noise. In *Proceedings IOA*, volume 14, pages 503–510, 1992.

[60] B P Milner and S V Vaseghi. Comparison of some noise-compensation methods for speech recognition in adverse environments. *IEE Proceedings Vision, Image and Signal Processing*, pages 280–288, 1994.

[61] Y Minami and S Furui. A maximum likelihood procedure for a universal adaptation method based on HMM composition. In *Proceedings ICASSP*, pages 129–132, 1995.

[62] C Mokbel, L Barbier, and G Chollet. Adapting a HMM speech recogniser to noisy environments. In *Proceedings ESCA Workshop on Speech Processing in Adverse Conditions*, pages 211–214, 1992.

[63] P J Moreno, B Raj, E Gouvea, and R M Stern. Multivariate-Gaussian-based Cepstral normalization for robust speech recognition. In *Proceedings ICASSP*, pages 137–140, 1995.

[64] N Morgan and H Hermansky. RASTA extensions: Robustness to additive and convolutional noise. In *Proceedings ESCA Workshop on Speech Processing in Adverse Conditions*, pages 115–118, 1992.

[65] A Nadas, D Nahamoo, and M Picheny. Speech recognition using noise-adaptive prototypes. In *Proceedings ICASSP*, pages 517–520, 1988.

[66] R M Neal. Probabilistic inference using Markov chain Monte Carlo methods. Technical Report CRG-TR-93-1, Department of Computer Science, University of Toronto, 1993.

[67] L Neumeyer and M Weintraub. Probabilistic optimum filtering for robust speech recognition. In *Proceedings ICASSP*, volume 1, pages 417–420, 1994.

[68] L Neumeyer and M Weintraub. Robust speech recognition in noise using adaptation and mapping techniques. In *Proceedings ICASSP*, pages 141–144, 1995.

[69] H Ney, U Essen, and R Kneser. On structuring probabilistic dependencies in stochastic language models. *Computer Speech and Language*, 8:1–38, 1994.

[70] J A Nolazco-Flores and S J Young. Adapting a HMM based recogniser for noisy speech enhanced by spectral subtraction. In *Proceedings Eurospeech*, pages 829–832, 1993.

[71] J A Nolazco-Flores and S J Young. Continuous speech recognition in noise using spectral subtraction and HMM adaptation. In *Proceedings ICASSP*, volume 1, pages 409–412, 1994.

[72] J J Odell, V Valtchev, P C Woodland, and S J Young. A one pass decoder design for large vocabulary recognition. In *Proceedings ARPA Workshop on Human Language Technology*, pages 405–410, 1994.

[73] J P Openshaw and J S Mason. On the limitations of Cepstral features in noise. In *Proceedings ICASSP*, volume 2, pages 49–52, 1994.

[74] J P Openshaw, Z P Sun, and J S Mason. A comparison of feature performance under degraded speech in speaker recognition. In *Proceedings ESCA Workshop in Speech Processing in Adverse Conditions*, pages 119–122, 1992.

[75] M Ostendorf and S Roukos. A stochastic segment model for phoneme-based continuous speech recognition. *IEEE Transactions ASSP*, 37:1857–1869, 1989.

[76] D S Pallett, J G Fiscus, W M Fisher, J S Garofolo, B A Lund, A Martin, and M A Przybocki. 1994 benchmark tests for the ARPA spoken language program. In *Proceedings ARPA Workshop on Spoken Language Systems Technology*, pages 5–36, 1995.

[77] P Price, W M Fisher, J Bernstein, and D S Pallett. The DARPA 1000-word Resource Management database for continuous speech recognition. In *Proceedings ICASSP*, pages 651–654, 1988.

[78] L R Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77, February 1989.

[79] L R Rabiner and S E Levinson. Isolated and connected word recognition — theory and selected applications. *IEEE Transactions on Communications*, 29:621–659, 1981.

[80] L R Rabiner and R W Schafer. *Digital Processing of Speech Signals*. Prentice Hall, 1978.

[81] R C Rose, E M Hofstetter, and D A Reynolds. Integrated models of signal and background with application to speaker identification in noise. *IEEE Transactions SAP*, 2:245–257, 1994.

[82] A Sankar and C H Lee. Robust speech recognition based on stochastic matching. In *Proceedings ICASSP*, pages 121–124, 1995.

[83] C W Seymour and M Niranjan. An HMM based Cepstral-domain speech enhancement scheme. In *Proceedings ICSLP*, pages 1595–1598, 1994.

[84] S D Silvey. *Statistical Inference*. Chapman and Hall, 1991.

[85] O Siohan. On the robustness of linear discriminant analysis as a preprocessing step for noisy speech recognition. In *Proceedings ICASSP*, pages 125–128, 1995.

[86] H B D Sorenson and U Hartmann. Robust speaker-independent speech recognition using non-linear spectral subtraction based IMELDA. In *Proceedings Eurospeech*, 1993.

[87] K Tokuda, T Kobayashi, T Masuko, and S Imai. Mel-generalised Cepstral analysis — A unified approach to speech spectral estimation. In *Proceedings ICSLP*, pages 1043–1046, 1994.

[88] D van Compernolle. Noise adaptation in the hidden Markov model speech recognition system. *Computer Speech and Language*, 3:151–168, 1987.

[89] A P Varga, R Moore, J Bridle, K Ponting, and M Russell. Noise compensation algorithms for use with hidden Markov model based speech recognition. In *Proceedings ICASSP*, pages 481–484, 1988.

[90] A P Varga and R K Moore. Hidden Markov model decomposition of speech and noise. In *Proceedings ICASSP*, pages 845–848, 1990.

[91] A P Varga, H J M Steeneken, M Tomlinson, and D Jones. The NOISEX-92 study on the effect of additive noise on automatic speech recognition. Technical report, DRA Speech Research Unit, 1992.

[92] S V Vaseghi and B P Milner. Speech modelling using Cepstral-time feature matrices in hidden Markov models. *IEE Proceedings Communications, Speech and Vision*, pages 317–320, 1993.

[93] A J Viterbi. Error bounds for convolutional codes and asymptotically optimum decoding algorithm. *IEEE Transactions Information Theory*, 13:260–269, 1982.

[94] A Waibel and K F Lee, editors. *Readings in Speech Recognition*. Morgan Kaufmann, 1990.

[95] P C Woodland, J J Odell, V Valtchev, and S J Young. The development of the 1994 HTK large vocabulary speech recognition system. In *Proceedings ARPA Workshop on Spoken Language Systems Technology*, pages 104–109, 1995.

[96] P C Woodland and S J Young. The HTK tied-state continuous speech recogniser. In *Proceedings Eurospeech*, pages 2207–2210, 1993.

[97] F Xie, D van Compernolle, and K U Leuven. Speech enhancement by nonlinear spectral estimation. In *Proceedings Eurospeech*, pages 617–620, 1993.

[98] S J Young. *HTK Version 1.4: Reference Manual and User Manual.* Cambridge University Engineering Department Speech Group, 1992.

[99] S J Young, J J Odell, and P C Woodland. Tree-based state tying for high accuracy acoustic modelling. In *Proceedings ARPA Workshop on Human Language Technology*, pages 307–312, 1994.

[100] S J Young, N H Russell, and J H S Thornton. Token passing; A conceptual model for connected speech recognition systems. Technical Report CUED/F-INFENG/TR38, Cambridge University, 1989. Available via anonymous ftp from: svr-ftp.eng.cam.ac.uk.

[101] S J Young and P C Woodland. The use of state tying in continuous speech recognition. In *Proceedings Eurospeech*, pages 2207–2210, 1993.

[102] S J Young, P C Woodland, and W J Byrne. *HTK: Hidden Markov Model Toolkit V1.5.* Entropic Research Laboratories Inc., 1993.

[103] V W Zue. The use of speech knowledge in automatic speech recognition. *Proceedings IEEE*, 73:1602–1615, 1985.