All rights reserved

INFORMATION TO ALL USERS
The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.
Robust Text-Independent Speaker Recognition
Over Telecommunication Systems

Halil Uzuner

Submitted for the Degree of
Doctor of Philosophy
from the
University of Surrey

Unis

Centre for Communication Systems Research
School of Electronics and Physical Sciences
University of Surrey
Guildford, Surrey GU2 7XH, U.K.

March 2006

© Halil Uzuner 2006
"To my parents Yäcel and Ünsel, and my brothers Ülker and Derviş"
Summary

Biometric recognition methods, using human features such as voice, face or fingerprints, are increasingly popular for user authentication. Voice is unique in that it is a non-intrusive biometric which can be transmitted over the existing telecommunication networks, thereby allowing remote authentication. Current speaker recognition systems can provide high recognition rates on clean speech signals. However, their performance has been shown to degrade in real-life applications such as telephone banking, where speech compression and background noise can affect the speech signal.

In this work, three important advancements have been introduced to improve the speaker recognition performance, where it is affected by the coder mismatch, the aliasing distortion caused by the Line Spectral Frequency (LSF) parameter extraction, and the background noise. The first advancement focuses on investigating the speaker recognition system performance in a multi-coder environment using a Speech Coder Detection (SCD) System, which minimises training and testing data mismatch and improves the speaker recognition performance.

Having reduced the speaker recognition error rates for multi-coder environment, further investigation on GSM-EFR speech coder is performed to deal with a particular problem related to LSF parameter extraction method. It has been previously shown that the classic technique for extraction of LSF parameters in speech coders is prone to aliasing distortion. Low-pass filtering on up-sampled LSF vectors has been shown to alleviate this problem, therefore improving speech quality. In this thesis, as a second advancement, the Non-Aliased LSF (NA-LSF) extraction method is introduced in order to reduce the unwanted effects of GSM-EFR coder on speaker recognition performance.

Another important factor that affects the performance of speaker recognition systems is the presence of the background noise. Background noise might severely reduce the performance of the targeted application such as quality of the coded speech, or the performance of the speaker recognition systems. The third advancement was achieved by using a noise-canceller to improve the speaker recognition performance in mismatched environments with varying background noise conditions. Speaker recognition system with a Minimum Mean Square Error - Log Spectral Amplitudes (MMSE-LSA) noise-canceller used as a pre-processor is proposed and investigated to determine the efficiency of noise cancellation on the speaker recognition performance using speech corrupted by different background noise conditions. Also the effects of noise cancellation on speaker recognition performance using coded noisy speech have been investigated.

Key words: Identification, Verification, Recognition, Gaussian Mixture Models, Speech Coding, Noise Cancellation.
Acknowledgements

I would like to express my sincere gratitude to my PhD supervisors, Professor Ahmet Kondoz and Dr. Stéphane Villette for their full support, encouragement and guidance during my PhD. It would have been unbearable without their presence. I also would like to thank all my colleagues in the Multimedia group who have contributed to this work with their friendship, advice and encouragement, especially to our ex-members Khaldoon Taha Al-Naimi and Nilantha Katugampala for sharing their valuable experience and advisory. I would also like to thank my colleagues and friends Safak Dogan, Akin Celatoglu, Cem Ergun, Halcan Altincay, Richard Edwards, Banu Gunel, Huseyin Hacihabiboglu, and Mingyou Hu for their help and support at various stages of my research work. I would like to thank my dearest friends Eleni Papatheasoupolou and Laura Martinez, for their friendship and support, which helped me to overcome many difficult problems. I also would like to thank Yabanok family for their support and love, especially to my cousin Bilgin.

I finally want to take this opportunity to say thanks to my parents and brothers for their love, continuous encouragement, and support.
Contents

Contents .............................................................................................................. v
List of Figures ........................................................................................................ x
List of Tables ........................................................................................................ xiii
Glossary of Terms .................................................................................................... xiv

1 Introduction 1
  1.1 Background ................................................................................................... 1
  1.2 Thesis Outline .............................................................................................. 3
  1.3 Original Contributions ................................................................................. 5

2 Speaker Recognition and Speech Signal Processing Techniques 7
  2.1 Introduction ................................................................................................... 7
  2.2 Biometrics ..................................................................................................... 8
    2.2.1 Application Areas ............................................................................... 9
  2.3 Speaker Recognition .................................................................................... 10
    2.3.1 Text-Dependent versus Text-Independent Systems ................... 11
    2.3.2 Speaker Identification Systems ...................................................... 12
    2.3.3 Speaker Verification Systems ......................................................... 13
    2.3.4 Problems with Speaker Recognition Systems ............................ 14
    2.3.5 Application Areas of Speaker Recognition ................................. 14
    2.3.6 Performance Evaluations ............................................................... 15
    2.3.7 Speech Databases .......................................................................... 15
  2.4 Speech Production ....................................................................................... 17
    2.4.1 Human Speech Production Mechanism ........................................... 17
    2.4.2 Synthetic Speech Production Mechanism ....................................... 19
2.5 Pre-Processing of Speech Signal ................................................................. 20
  2.5.1 Signal Pre-Emphasising ................................................................. 20
  2.5.2 Silence Detection ........................................................................ 20
  2.5.3 Frame Analysis and Windowing ................................................... 21
2.6 Feature Extraction .................................................................................... 22
  2.6.1 Linear Prediction Coefficients (LPC) ........................................... 23
  2.6.2 Mel-Frequency Cepstral Coefficients (MFCCs) ......................... 29
2.7 Speech Signal Enhancement ...................................................................... 36
  2.7.1 Pre-processing ........................................................................... 37
  2.7.2 Post-Processing ....................................................................... 38
2.8 Distance Measures ................................................................................... 39
2.9 Conclusion ............................................................................................... 40

3 Speaker Modeling and Recognition ............................................................. 42
  3.1 Introduction ....................................................................................... 42
  3.2 Parametric versus Non-Parametric Modeling ..................................... 43
  3.3 Non-parametric Modeling ................................................................... 44
    3.3.1 Long-Term Averaging ............................................................... 44
    3.3.2 Vector Quantization (VQ) ......................................................... 45
    3.3.3 Neural Networks (NNs) ........................................................... 47
    3.3.4 Support Vector Machines (SVMs) ........................................... 48
  3.4 Parametric Modeling .......................................................................... 50
    3.4.1 Gaussian Mixture Models (GMMs) ........................................... 50
    3.4.2 Hidden Markov Models (HMMs) ............................................. 66
  3.5 Experimental Setup ............................................................................. 68
    3.5.1 Speaker Databases ................................................................... 68
    3.5.2 Front-End Feature Extraction ................................................ 70
    3.5.3 Speaker Identification Experiments ........................................ 71
    3.5.4 Speaker Verification Experiments ........................................... 73
  3.6 Conclusion ........................................................................................... 77
4 Coded Speech Speaker Recognition 79
4.1 Introduction ................................................................. 79
4.2 Data Mismatch ............................................................. 80
4.3 Speech Coders .............................................................. 81
  4.3.1 Mixed-Excitation Linear Predictive (MELP) Speech Coder ...... 82
  4.3.2 G723.1 Speech Coder ............................................... 82
  4.3.3 G729 Speech Coder .................................................. 83
  4.3.4 GSM-AMR Speech Coder ......................................... 84
4.4 Speaker Recognition Performance Using Coded Speech .......... 84
  4.4.1 Matched and Mismatched case Definitions for Speech Coding 85
4.5 Data Mismatch of Multi-Coder Situation ............................... 89
4.6 Speech Coder Detection System ........................................ 90
4.7 Speech Coder Detection System Performance ....................... 92
4.8 Speaker Verification Experiments with Speech Coder Detection System 93
4.9 Analysis and Observations ............................................... 95
  4.9.1 Observation One: Effect of Two Coding Algorithms ......... 96
  4.9.2 Observation Two: Effect of Coder Components ............. 97
  4.9.3 Observation Three: Effects of Filtering on the Coded Speech 98
4.10 Conclusion ................................................................... 105

5 Application of Improved LSF Extraction Through Anti-Aliasing Filter 107
5.1 Introduction ................................................................. 107
5.2 LSF Smoothing Through Anti-Aliasing Filtering ................... 108
5.3 Anti-Aliasing Filtering Process ......................................... 110
  5.3.1 Sampling Theory ..................................................... 110
  5.3.2 Sampling Rate Conversion ........................................ 110
  5.3.3 LSF Extraction From a Decimation Perspective ............. 111
  5.3.4 Low-Pass Filtering .................................................. 113
5.4 Speaker Verification Using NA-LSFs in GSM-EFR ................. 119
  5.4.1 Speaker Verification Experiments ................................ 119
5.5 Conclusion ................................................................... 122
6 Noise Cancellation for Speaker Verification

6.1 Introduction ......................................................................................................... 123
6.2 Background Noise and MMSE-LSA Noise Suppression ............................... 124
  6.2.1 Soft-Decision Gain ............................................................................. 126
  6.2.2 Log-Spectral Amplitude Gain .............................................................. 127
  6.2.3 A Priori Probability Estimation ........................................................... 128
  6.2.4 Noise Spectrum Adaptation ................................................................. 129
  6.2.5 Voice Activity Detector (VAD) ........................................................... 130
6.3 Speaker Verification using Noise-Canceller ..................................................... 130
  6.3.1 Speaker Training and Test Data ............................................................ 130
  6.3.2 Experimental Setup ............................................................................. 131
  6.3.3 Verification Experiments ...................................................................... 131
  6.3.4 Noise Cancellation and Speech Coding ............................................... 133
  6.3.5 Noise Cancellation for Different Background Noise Characteristics 135
6.4 Conclusion ........................................................................................................ 140

7 Conclusions .......................................................................................................... 143
  7.1 Concluding Overview ..................................................................................... 143
  7.2 Future Work .................................................................................................... 146

A Male and Female Speaker Verification Experiments for Chapter 8 148
List of Figures

2.1 Block diagram of a speaker recognition system ........................................ 11
2.2 Speaker identification system ................................................................. 12
2.3 Speaker verification system ..................................................................... 13
2.4 Anatomy of human speech production ................................................... 17
2.5 Source-filter model ................................................................................ 19
2.6 LPC filter frequency response and LSPs of a voiced speech ...................... 29
2.7 Block diagram of a MFCC front-end processor ......................................... 30
2.8 Mel-scale filter-bank frequency responses .............................................. 33

3.1 Comparison of feature vectors of the unknown speaker with models of the known speakers ................................................................. 43
3.2 Optimal separating hyperplane in two-dimensional space ......................... 49
3.3 M component Gaussian mixture density .................................................. 52
3.4 Speaker identification system .................................................................. 57
3.5 Speaker verification system .................................................................... 59
3.6 Speaker model adaptation from UBM ..................................................... 62
3.7 Detection Error Trade-off curve ............................................................... 66
3.8 Example of a three state left-to-right HMM ............................................ 67
3.9 DET curve of baseline verification system for male speech using TIMIT8k and NTIMIT8k databases ................................................................. 76
3.10 DET curve of baseline verification system for female speech using TIMIT8k and NTIMIT8k databases ................................................................. 77

4.1 Speaker recognition via various networks .............................................. 86
4.2 EER values for male speaker verification experiments using four different coders with matched and mismatched conditions .............................................. 88
4.3 EER values for female speaker verification experiments using four different coders with matched and mismatched conditions ................................. 88
4.4 Combined speech coder detection and speaker verification system ........ 91
4.5 Male and female speaker speech coder detection rate in four-coder environment using TIMIT8k database ............................................................... 93
4.6 Male speaker DET curves of matched case verification, mismatched case verification, and combined speech coder detection/verification system using TIMIT8k database ............................................................. 94
4.7 Female speaker DET curves of matched case verification, mismatched case verification, and combined speech coder detection/verification system using TIMIT8k database ............................................................. 94
4.8 Effect of speech coder component removal on female speaker speech coder detection rate in two coder environment using TIMIT8k database ................. 98

5.1 Over-sampled LSF parameter extraction .................................................. 108
5.2 Spectra of LSF tracks extracted at every sample ......................................... 109
5.3 Concentrating on region of interest (i.e. vector transmission rate at 50 Hz) of the spectra of LSF tracks extracted at every sample ........................... 109
5.4 Time plots of the GSM-EFR coder LP analysis windows ......................... 112
5.5 Frequency responses of the GSM-EFR coder LP analysis windows .......... 113
5.6 Variations in the 1st LSF track for original $f_1$ and low-pass filtered $g_1$ LSFs ................................................................. 115
5.7 Variations in the 4th LSF track for original $f_4$ and low-pass filtered $g_4$ LSFs ................................................................. 116
5.8 Variations in the 7th LSF track for original $f_7$ and low-pass filtered $g_7$ LSFs ................................................................. 117
5.9 Variations in the 10th LSF track for original $f_{10}$ and low-pass filtered $g_{10}$ LSFs ................................................................. 118
List of Tables

<table>
<thead>
<tr>
<th></th>
<th>Table Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Mel-scale filter-bank</td>
<td>32</td>
</tr>
<tr>
<td>2.2</td>
<td>Mel-scale like filter-bank</td>
<td>34</td>
</tr>
<tr>
<td>3.1</td>
<td>Results of baseline identification system</td>
<td>72</td>
</tr>
<tr>
<td>3.2</td>
<td>EER results of baseline verification system for male speech using TIMIT8k and NTIMIT8k databases</td>
<td>75</td>
</tr>
<tr>
<td>3.3</td>
<td>EER results of baseline verification system for female speech using TIMIT8k and NTIMIT8k databases</td>
<td>76</td>
</tr>
<tr>
<td>4.1</td>
<td>Possible versions of the mismatched conditions for speaker modeling and testing</td>
<td>87</td>
</tr>
<tr>
<td>4.2</td>
<td>Clean female speech transduced using the G729 coder and then transduced using G723.1 coder</td>
<td>96</td>
</tr>
<tr>
<td>4.3</td>
<td>Clean female speech transduced using the G723.1 coder and then transduced using G729 coder</td>
<td>97</td>
</tr>
<tr>
<td>4.4</td>
<td>G723.1 coded male test speech SCD performance (in %) in two coder environment using G723.1, and G729 coders with different filtering applied to coded speech</td>
<td>99</td>
</tr>
<tr>
<td>4.5</td>
<td>G723.1 coded female test speech SCD performance (in %) in two coder environment using G723.1, and G729 coders with different filtering applied to coded speech</td>
<td>100</td>
</tr>
<tr>
<td>4.6</td>
<td>AMR coded male test speech SCD performance (in %) in three coder environment using G723.1, G729, and AMR coders with different filtering applied to coded test speech</td>
<td>101</td>
</tr>
<tr>
<td>4.7</td>
<td>AMR coded female test speech SCD performance (in %) in three coder environment using G723.1, G729, and AMR coders with different filtering applied to coded test speech</td>
<td>102</td>
</tr>
<tr>
<td>4.8</td>
<td>HPF (60 Hz cut-off freq.) AMR coded male test speech SCD performance (in %) in three coder environment using G723.1, G729, and AMR coders</td>
<td>103</td>
</tr>
</tbody>
</table>
List of Tables

4.9 HPF (60 Hz cut-off freq.) AMR coded female test speech SCD performance (in %) in three coder environment using G723.1, G729, and AMR coders ................................................................. 103

4.10 MELP coded male test speech SCD performance (in %) in four coder environment using MELP, G723.1, G729, and AMR coders with different filtering applied to coded test speech ........................................... 104

4.11 MELP coded female test speech SCD performance (in %) in four coder environment using MELP, G723.1, G729, and AMR coders with different filtering applied to coded test speech ........................................... 105

5.1 EER values of verification system for male speech using TIMIT8k, Original-LSF TGSM, and NA-LSF TGSM databases. ......................................................... 120

5.2 EER values of verification system for female speech using TIMIT8k, Original-LSF TGSM, and NA-LSF TGSM databases. ......................................................... 121

6.1 Averaged male and female EER values of verification system using TIMIT8k, Noisy TIMIT8k corrupted with vehicle noise (15 dB SNR), and noise-cancelled speech using MMSE-LSA noise canceller. ...................... 132

6.2 Averaged male and female speaker verification EER values for noisy training speech (15 dB SNR, vehicular noise) and noisy test speech with different noise types and SNRs .................. 136

6.3 Averaged male and female speaker verification EER values for noise-cancelled training speech (15 dB SNR, vehicular noise) and noise-cancelled test speech with different noise types and SNRs .................. 137

6.4 Averaged male and female speaker verification EER values for noise-cancelled training speech (15 dB SNR, vehicular noise) and noisy test speech with different noise types and SNRs .................. 138

6.5 Averaged male and female speaker verification EER values for clean training speech and noise-cancelled test speech with different noise types and SNRs .................. 139

6.6 Averaged male and female speaker verification EER values for clean training speech and noisy test speech with different noise types and SNRs .................. 139

A.1 EER values of verification system for male speech using TIMIT8k, Noisy TIMIT8k corrupted with vehicle noise (15 dB SNR), and noise-cancelled speech using MMSE-LSA noise canceller. ...................... 149

A.2 EER values of verification system for female speech using TIMIT8k, Noisy TIMIT8k corrupted with vehicle noise (15 dB SNR), and noise-cancelled speech using MMSE-LSA noise canceller. ...................... 150

A.3 Male speaker verification EER values for noisy training speech (15 dB SNR, vehicular noise) and noisy test speech with different noise types and SNRs .................. 151
A.4 Female speaker verification EER values for noisy training speech (15 dB SNR, vehicular noise) and noisy test speech with different noise types and SNRs .............................................. 151
A.5 Male speaker verification EER values for noise-cancelled training speech (15 dB SNR, vehicular noise) and noise-cancelled test speech with different noise types and SNRs ............................................. 152
A.6 Male speaker verification EER values for noise-cancelled training speech (15 dB SNR, vehicular noise) and noisy test speech with different noise types and SNRs ............................................. 152
A.7 Female speaker verification EER values for noise-cancelled training speech (15 dB SNR, vehicular noise) and noise-cancelled test speech with different noise types and SNRs ............................................. 153
A.8 Female speaker verification EER values for noise-cancelled training speech (15 dB SNR, vehicular noise) and noisy test speech with different noise types and SNRs ............................................. 153
A.9 Male speaker verification EER values for clean training speech and noise-cancelled test speech with different noise types and SNRs ............................................. 154
A.10 Male speaker verification EER values for clean training speech and noisy test speech with different noise types and SNRs ............................................. 154
A.11 Female speaker verification EER values for clean training speech and noise-cancelled test speech with different noise types and SNRs ............................................. 155
A.12 Female speaker verification EER values for clean training speech and noisy test speech with different noise types and SNRs ............................................. 155
# Glossary of Terms

<p>| <strong>ACELP</strong> | Algebraic Code Excited Linear Prediction |
| <strong>AMR</strong> | Adaptive Multi-Rate |
| <strong>ANN</strong> | Artificial Neural Networks |
| <strong>ATM</strong> | Automated Teller Machine |
| <strong>BPF</strong> | Band-Pass Filter |
| <strong>CCSR</strong> | the Centre for Communication Systems Research |
| <strong>CMS</strong> | Cepstral Mean Subtraction |
| <strong>CS-ACELP</strong> | Conjugate-Structure Algebraic Code Excited Linear Prediction |
| <strong>DCT</strong> | Discrete Cosine Transform |
| <strong>DET</strong> | Detection Error Trade-off |
| <strong>DFT</strong> | Discrete Fourier Transform |
| <strong>EER</strong> | Equal Error Rate |
| <strong>EFR</strong> | Enhanced Full-Rate |
| <strong>EM</strong> | Expectation-Maximisation |
| <strong>ETSI</strong> | European Telecommunications Standards Institute |
| <strong>FA</strong> | False Acceptance |
| <strong>FAR</strong> | False Acceptance Rate |
| <strong>FFT</strong> | Fast Fourier Transform |
| <strong>FR</strong> | False Rejection |
| <strong>FRR</strong> | False Rejection Rate |
| <strong>GMM</strong> | Gaussian Mixture Model |
| <strong>GSM-AMR</strong> | Global System for Mobile Communications - Adaptive Multi-Rate |
| <strong>HMM</strong> | Hidden Markov Model |
| <strong>HPF</strong> | High-Pass Filter |
| <strong>ITU</strong> | International Telecommunication Union |
| <strong>LAR</strong> | Log Area Ratios |
| <strong>LDC</strong> | Linguistic Data Consortium |
| <strong>LFCC</strong> | Linear-Frequency spaced filter-bank Cepstral Coefficients |
| <strong>LP</strong> | Linear Prediction |
| <strong>LPCC</strong> | Linear Prediction Cepstral Coefficients |
| <strong>LPF</strong> | Low-Pass Filter |
| <strong>LSA</strong> | Log-Spectral Amplitude |
| <strong>LSF</strong> | Line Spectral Frequency |
| <strong>LSP</strong> | Line Spectral Pair |
| <strong>MAP</strong> | Maximum a posteriori probability |
| <strong>MELP</strong> | Mixed-Excitation Linear Predictive |</p>
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MFCC</td>
<td>Mel-Frequency Cepstral Coefficient</td>
</tr>
<tr>
<td>ML</td>
<td>Maximum Likelihood</td>
</tr>
<tr>
<td>MMSE</td>
<td>Minimum Mean Square Error</td>
</tr>
<tr>
<td>MMSE-LSA</td>
<td>Minimum Mean Square Error - Log Spectral Amplitudes</td>
</tr>
<tr>
<td>msc</td>
<td>maximally spread close</td>
</tr>
<tr>
<td>msf</td>
<td>maximally spread far</td>
</tr>
<tr>
<td>MPLPC</td>
<td>Multi-Pulse Linear Prediction Coding</td>
</tr>
<tr>
<td>NA-LSF</td>
<td>Non-Aliased Line Spectral Frequency</td>
</tr>
<tr>
<td>NN</td>
<td>Neural Networks</td>
</tr>
<tr>
<td>UBM</td>
<td>Universal Background Model</td>
</tr>
<tr>
<td>NIST</td>
<td>National Institute of Standards and Technology</td>
</tr>
<tr>
<td>PC</td>
<td>Personal Computer</td>
</tr>
<tr>
<td>pdf</td>
<td>probability density function</td>
</tr>
<tr>
<td>PIN</td>
<td>Personal Identification Number</td>
</tr>
<tr>
<td>PLMN</td>
<td>Public Land Mobile Networks</td>
</tr>
<tr>
<td>PLPC</td>
<td>Perceptual Linear Prediction Coefficients</td>
</tr>
<tr>
<td>PSTN</td>
<td>Public Switched Telephone Network</td>
</tr>
<tr>
<td>RASTA</td>
<td>RelAtive SpecTrAl</td>
</tr>
<tr>
<td>RBF</td>
<td>Radial Basis Function</td>
</tr>
<tr>
<td>RMS</td>
<td>Root Mean Square</td>
</tr>
<tr>
<td>SCID</td>
<td>Speech Coder Detection</td>
</tr>
<tr>
<td>SNR</td>
<td>Signal-to-Noise Ratio</td>
</tr>
<tr>
<td>STSA</td>
<td>Short-Time Spectral Amplitudes</td>
</tr>
<tr>
<td>SVM</td>
<td>Support Vector Machine</td>
</tr>
<tr>
<td>VAD</td>
<td>Voice Activity Detection</td>
</tr>
<tr>
<td>VQ</td>
<td>Vector quantization</td>
</tr>
<tr>
<td>WMSE</td>
<td>Weighted Mean Square Error</td>
</tr>
<tr>
<td>3GPP</td>
<td>3rd Generation Partnership Project</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Background

In modern daily life, people are interacting with machines more than ever. Most of these human-machine interactions require different kinds of user recognition methods to ensure that only the correct user is allowed to gain access to a service that is provided. Passwords, Personal Identification Numbers (PIN) and swipe cards have been the most common methods of identification between humans and machines for applications such as computer access, transaction security, and data protection. In order to increase the level of security and eliminate the dependency to PINs and cards, biometric recognition methods have been used to authenticate people. Biometric recognition methods use people's physiological and/or behavioral characteristics to perform recognition. Some of the biometric features used for measurement are iris, retinal, fingerprint, signature, face and voice, which are more convenient than the conventional methods since there is no possibility of not remembering your password or leaving your swipe card at home. Voice as a biometric feature is a very popular method. It is very easy to obtain, store and transmit over a telephone network, and more importantly it is non-intrusive to the users, which makes it very popular compared to other biometric features [1]. Speaker recognition is a process of identifying an individual from his/her voice. The speaker is recognised by his/her voice characteristics, which are modeled using pattern recognition methods such as Gaussian Mixture Models (GMM) [2] and Hidden Markov Models
(HMM) [3].

Speaker recognition systems can be classified as identification and verification systems. The identification systems detect the identity of a speaker from a group of known speakers, whereas the verification system detects whether the claimed speaker is an enrolled true speaker or an impostor. The verification task is a binary class problem, and the outcome of the speaker recognition test is either acceptance or rejection of the claimant speaker. The speaker verification systems are popular for commercial applications, since they only compute the claimed speaker’s identity.

The recognition process has two major parts, which are speaker training and speaker testing processes. The speaker training part uses the speaker’s voice, and creates a representative model of his/her voice characteristics. The speaker testing part involves the recognition decision process, where the test speaker’s voice and the enrolled speaker models are compared to see if the test speaker’s voice matches with one of the enrolled speaker models.

State-of-the-art speaker recognition systems produce high recognition rates, such as the system shown in [4] with 99.5% correct identification performance, when clean speech is used during the recognition process. However, the recognition performance may degrade due to the variability in the environment such as existence of background noise, channel noise, and speech compression methods (speech coding). The distortions introduced by telephone handset microphones, background noises such as office and vehicular noise, and coding schemes, which are used to compress speech for transmission, affect the speaker voice characteristics and make the speaker recognition task more difficult. In all these problems mentioned above, the level of speaker performance degradation increases further when the training and the testing data are collected from the environments, where the characteristics of the training data do not match with the characteristics of the testing data. An example to the mismatched training and testing conditions can be shown as the training speech is corrupted by an office noise and the test speech is corrupted by a vehicular (car engine) noise.

Handset dependent speaker models and score normalisation techniques such as HNORM have been used to reduce the effects of non-linear distortions produced by different handset microphone types (i.e. electret and carbon-button type) [5].
1.2 Thesis Outline

Background noise also effects the speaker recognition systems. The different level and the type of noise may cause different amounts of degradation on speaker recognition performance. The effects of background noise can be compensated by using methods such as speaker models with integrated noise models, and speech enhancement methods such as stand-alone noise-cancellers.

The speech coding algorithms, such as GSM-AMR and other low bit rate types such as MELP, are used to compress speech for transmission purposes. The speech coding algorithms also cause recognition performance degradation. Methods such as score normalisation techniques, and Speech Coder Detection (SCD) systems have been introduced to reduce the effects of speech coding on speaker recognition performance.

The work presented in this thesis focuses on the effects of coding schemes and background noise on speaker recognition performance in matched and mismatched training and testing conditions, and the development of methods to improve the speaker recognition performance under such circumstances.

1.2 Thesis Outline

The research work is mainly focused on improving speaker recognition performance influenced by noise and coding techniques. The thesis is organised as follows:

- **Chapter 2: Speaker Recognition and Speech Signal Processing Techniques**

  In Chapter 2, the biometric recognition methods are briefly introduced. The speaker recognition process, with identification and verification tasks, is described. After introducing the concept of speaker recognition, the human speech production mechanism and signal processing techniques for speaker recognition are presented. Parts of a typical front-end processor, such as pre-processing and feature extraction, of a speaker recognition system are explained. Finally the distance measures used in speaker recognition applications are introduced.

- **Chapter 3: Speaker Modeling and Recognition**

  Chapter 3 presents some of the popular speaker modeling techniques, such as GMMs, that are used for speaker recognition systems. Practical issues such as
initialisation, training and testing processes are described in detail. This is followed by the introduction of the baseline speaker identification and verification systems. The experimental setup of speaker identification and verification tasks are described, explaining the details of the training and testing processed for both tasks. The identification and verification system performances are presented using clean and noisy speech samples.

- **Chapter 4: Coded Speech Speaker Recognition**

Chapter 4 focuses on the speaker recognition performance degradation introduced by using coded speech in a speaker recognition system. The relationship between the speech coder bit rate and speaker performance degradation is investigated. In order to analyse and improve the speaker recognition performance under mismatched training and testing conditions (coding scheme mismatch), a description of the effects of the Speech Coder Detection (SCD) system is presented. The results obtained from the SCD system are further analysed to determine factors allowing the SCD to distinguish between different coders.

- **Chapter 5: Application of Improved LSF Extraction Through Anti-Aliasing Filtering**

This chapter investigates the use of the Non-Aliased-LSF (NA-LSF) parameter extraction method, which was originally developed to eliminate the weaknesses observed in the classical LSF extraction method. The performance of a speaker recognition system using coded speech is then analysed to demonstrate the benefits of using NA-LSF's during speech coding.

- **Chapter 6: Noise Cancellation for Speaker Verification**

In Chapter 6, the speaker verification system performance under noisy conditions is investigated. The use of the Minimum Mean Square Error - Log Spectral Amplitudes (MMSE-LSA) noise-cancellation technique for the speaker verification task under different noise conditions is described. The effect of background noise on speaker verification performance using clean and coded speech is investigated. The noise-canceller is used as a pre- and post-processing technique on speaker
recognition systems and the recognition performance is analysed. Later in Chapter 6, the effects of different noise types and Signal-to-Noise Ratio (SNR) levels on speaker recognition performance are demonstrated, hence analysing the efficiency of a noise-canceller for matched and mismatched training and testing conditions.

- **Chapter 7: Conclusions**

Concluding remarks are given in Chapter 7, which summarises the results obtained from the previous chapters and contains possible future work to be carried out.

### 1.3 Original Contributions

The original contributions included in this thesis are summarised as follows:

- Speaker identification and speaker verification systems have been investigated and the baseline systems have been implemented and tested using GMMs for speaker classification. The speaker recognition performances of the baseline systems using clean and noisy speech have been shown to be consistent with [4].

- A speech coder detection system has been investigated and developed to be used for speaker recognition in a multi-coder environment to prevent the performance degradation introduced by mismatched training and testing environments. It was shown that the speaker recognition performance degradation improves, providing recognition performance nearly as good as the matched case, when such a system is combined with a speaker verification system. The results obtained from speech coder detection experiments have been further investigated to analyse the detection system in detail. It has been shown that the frequency bandwidth of the speech signals plays an important role in determining the speech coder type, rather than the speech coding algorithms. This information can be utilised to improve the speaker recognition performance further.

- Previous work has shown that the LSF vectors obtained with classical extraction methods contain undesired frequency components, which results in some aliasing
noise in the LSF parameters. A novel NA-LSF parameter extraction approach has been introduced in order to remove the undesired frequency components on the LSF tracks of the GSM-EFR coder. The results obtained from these experiments show that the use of NA-LSF parameter extraction increases speaker verification performance.

- Speaker verification system performance has been investigated using a noisecanceller to demonstrate the effectiveness of the MMSE-LSA noise cancellation algorithm under various combinations of matched and mismatched training and testing conditions with various background noise types and SNR levels. It was shown that when the MMSE-LSA noise-canceller is used as a signal enhancement technique prior to verification task, the speaker verification performance degradation introduced by the background noise and training and testing data mismatch reduces.
Chapter 2

Speaker Recognition and Speech
Signal Processing Techniques

2.1 Introduction

Automated personal recognition (here recognition is used as a term for detecting a person's identity and also known as authentication) using biometric technologies has been the center of attention for the past several decades. The traditional recognition methods mainly rely on possession of a unique item such as a swipe card, and knowledge of unique secret information such as a password. However, most of these systems cannot determine the difference between the actual user and the impostors, when the required input is presented. A solution to this problem is the use of biometric features for automated personal recognition. Each person has unique physiological and behavioral characteristics, which can be used to distinguish that individual from other people. Our biological features such as face, eyes, voice, hands, and fingerprints can be used to distinguish one person from another. It is desired to have features that are very distinct for an individual compared to the other people. On the other hand, it is desired to have ideally very small or no changes in these features over time and under certain conditions such as aging. Such features, when used for authentication, provide high levels of security preventing impostors from accessing personal information and/or services.
Chapter 2 is organised as follows:
First, biometric recognition methods and their application areas are presented. Then the concept of speaker recognition is introduced, providing the details of recognition systems such as different recognition tasks, problems associated with speaker recognition, speaker databases, and performance evaluations. This is followed by the descriptions of a human speech production mechanism, and the necessary building blocks of a typical speaker recognition system such as the signal pre-processing and feature extraction parts. The chapter is finalised by definitions of speech signal enhancement methods and distance measures.

2.2 Biometrics

Biometrics can be described as automated methods, which use measurement of physiological and behavioral characteristics to recognise a person. Firstly, biometric features are collected using a capturing device such as cameras. The collected biometric feature is then transformed into a model, which represents a unique identity template for future comparison purposes. This process is called an enrollment process, where user related information for each individual is modeled and stored in a database. Later, at the test stage, the stored reference model is compared with the measured test features of an individual to perform recognition.

There are many different biometric methods described in the literature [6, 7]. The most popular biometric methods are as follows:

- Fingerprint recognition
- Voice (speaker) recognition
- Facial recognition
- Iris recognition
- Signature recognition
- Hand geometry recognition
2.2. Biometrics

There are some other less popular biometric recognition methods such as retinal scanning, keystroke dynamics, and vein pattern recognition. Each biometric method has its own advantages and disadvantages. The choice of biometric method depends on the application, and its practicality/performance in that particular application. Therefore one should carefully consider the targeted application and choose the type of biometric feature accordingly. As an example, depending on the environment, where the biometric recognition is taking place, some biometric recognition methods might be less advantageous. One might prefer not to use a facial recognition method if there is a constant change in the lighting of the room; or if there is an excessive amount of background noise present, one should consider not to use voice recognition methods [7]. Hence, each method provides unique information about the biometric feature of an individual, and can be utilised for different application areas, where the best recognition performance can be achieved.

2.2.1 Application Areas

The main application areas, where biometric features can be employed for user recognition, can be summarised as below:

- **Personal Banking**: Use of features such as voice over the telephone networks or Automated Teller Machines (ATMs), and use of fingerprint and iris recognition over ATMs for an extra layer of security service

- **Personal Computer (PC)/network access control**: Providing access to networks and use of computers employing recognition methods such as fingerprint and voice recognition

- **Prisoner control**: Physical access control for officers, inmates and prisoner visitors

- **Public services**: Biometric recognition systems can be used for services such as national identity cards, and electoral registration systems

- **Border control**: Control of country borders with assistance of biometric recognition methods usually provided in a chip placed in passports or identity cards.
The number of application areas can be further extended to many other daily examples. Briefly, biometric features can be used to identify people in many different application areas with increasingly improved level of security. It is expected that the widespread use of biometric recognition methods will be commonplace in the near future, especially for the national security purposes.

2.3 Speaker Recognition

Amongst all the biometric recognition methods, voice as a biometric feature has a unique place. Voice is a natural signal to produce, therefore is a non-intrusive biometric feature. It can be easily transmitted over the existing telecommunication networks, requiring no additional transmission medium and allowing remote authentication. It does not require any special sophisticated equipment apart from a microphone and a PC to be used for authentication purposes. Speaker recognition is a process of recognising who is speaking by using characteristics of a speaker's voice [8, 9]. A typical speaker recognition system has three main parts, namely feature extraction, training and testing. Both training and testing parts require features obtained from the feature extraction part. A block diagram of a typical speaker recognition system is shown in Figure 2.1, where training and testing blocks represent the training and testing processes following the feature extraction stage.

The feature extraction process creates a set of vectors called feature vectors, which carry information about speaker-specific characteristics of the input speech. The training part uses training set speaker voices to create an individual speaker model to represent each speaker in a set of speakers. This set of speakers is called variously, enrolled speakers, registered speakers, set of known speakers, or true speakers. The testing part compares the voice of a test speaker with the enrolled speaker models to check if the characteristics of the test speech is matching with the speaker model characteristics. The variations in a person's voice due to aging, or having a cold, are called intra-speaker variability. The variations between different people's voices are called inter-speaker variability. Speaker recognition systems can be text-dependent or text-independent
2.3. Speaker Recognition

2.3.1 Text-Dependent versus Text-Independent Systems

Text-dependent speaker recognition systems require users to read pre-arranged texts. The same pre-arranged texts must be used for both training and testing processes. Text-independent speaker recognition systems allow users to speak freely i.e. read any material for training and testing processes. However, the speaker's complete range of vocal sounds must be obtained during training process. Therefore, these systems require more speech samples for speaker training compared to text-dependent systems. Text-dependent speaker recognition systems generally provide better recognition performances than the text-independent systems, since the training and testing data are limited to prescribed texts. On the other hand, text-independent speaker recognition systems can be used in many application areas where the amount of speech to be used is very limited, or speakers are not cooperative, such as law-enforcement areas.

A typical speaker recognition system can be further classified as an identification system or a verification system depending on the aimed application.
Chapter 2. Speaker Recognition and Speech Signal Processing Techniques

2.3.2 Speaker Identification Systems

Speaker identification is a process of determining the unknown speaker's identity from a set of known speakers. Figure 2.2 shows a basic speaker identification process. The speaker identification task can be further classified as closed-set and open-set identification.

Closed-set

In a closed-set identification task, it is known that the unknown speaker is one of the speakers from the known set of speakers (i.e. enrolled speakers). The identification system only needs to decide which one of these enrolled speakers is the best match for the unknown speaker.
Speaker Recognition

2.3. Speaker Recognition

In an open-set identification task, the system has to decide if the unknown speaker is from the set of known speakers or an impostor. If the unknown speaker is an impostor, the recognition system rejects the user. Otherwise, if the recognition system decides that the unknown speaker is one of the registered speakers, the next stage is to determine the true identity of the speaker.

2.3.3 Speaker Verification Systems

Speaker verification is a process of determining if the claimed speaker is the person that he or she claims to be. Figure 2.3 shows a basic speaker verification process.

In the case of the speaker verification task, the decision criteria must be carefully chosen otherwise False Acceptance (FA) or False Rejection (FR) problems may occur. In the case of FA, an impostor is accepted as a known speaker. FR is the case of true speaker being rejected as an impostor.
2.3.4 Problems with Speaker Recognition Systems

In many real-life applications, the speaker recognition systems are designed to perform real-time voice recognition since results are needed immediately. Generally, it is not possible to provide an ideal recording environment, where there is not any background noise, echo etc. Also it may not be possible to use high-quality microphones to capture the speaker's voice. Speaker recognition systems may suffer from impostors using recorded speech to deceive the system. This problem can be solved by using text-prompted speaker recognition methods [10, 11] that ask users to utter different texts each time.

Speaker recognition systems also have to deal with problems introduced by using speech material that is corrupted by background noise, transmission channel variations, and different handset microphone types. Different background noises, telephone channels and microphone types have different effects on a speaker's speech. Speaker models and test feature vectors using speech samples from telephone conversations include speaker's characteristics plus distortions created by the background noise, transmission channel and handset type that is used for data collection. All these factors have to be considered and must be compensated to prevent recognition performance degradations [5].

2.3.5 Application Areas of Speaker Recognition

There are many areas, where user's speech can be used to gain access to services such as confidential information databases, telephone banking, computer access, and telephone shopping. Also speaker recognition systems can be used for law enforcement purposes, where a speaker's voice may be used as evidence for judgment.

As mentioned earlier, speaker identification is a process of comparing the unknown voice with known speaker models and finding the enrolled speaker, whose model is the best match for the unknown sample. The main application area for speaker identification is the law enforcement area. Typical examples are determining the identity of a suspect for bomb threat call or identifying the possible murderer's voice.

In the speaker verification task, the idea is to check if the test speech belongs to the
2.3. Speaker Recognition

claimant speaker. Application areas for speaker verification are mainly security applications to gain access to certain facilities or databases such as money transactions, and PC/network access. Another application area is a prisoner monitoring task, that is controlling prisoner’s availability in the specified location.

Text-dependent speaker recognition requires speakers to read given texts. Therefore, with a limited amount of speech material, it is crucial to have cooperative speakers in order to achieve high recognition performance. Hence, text-dependent systems can be used for applications, where speakers are willing to provide speech samples. On the other hand, text-independent systems do not require cooperative speakers, and can be used in applications where speakers are not cooperative such as forensic applications. There are many other application areas such as using voice for identity control at the national borders, accessing games and other types of entertainment sources, where speaker recognition techniques can be utilised to make our lives more secure and practical [7, 12].

2.3.6 Performance Evaluations

The speaker recognition system performance is based on the error rate of identification/verification process. Factors such as amount of training and testing data for each speaker, number of speakers used for training and testing processes, and quality of the speech data have an overall effect on the speaker recognition performance. In speaker identification systems, usually the number of correctly identified speakers are reported as percentages. However, in verification systems the error rates are usually demonstrated as Detection Error Trade-off (DET) [13] curves or reported as Equal Error Rate (EER) values [14] (see Section 3.4.1).

2.3.7 Speech Databases

It is very important to use standard databases (also known as speech corpus), which are specifically designed for speaker recognition applications. Such databases provide reliable and calibrated speech material, and allow researchers to compare the results of their research work with other people’s work. Some of the widely used databases for
speaker recognition tasks are TIMIT [15], NTIMIT [16], KING [17, 18], and YOHO [19]. In this work, TIMIT and NTIMIT databases are used, which provide clean and noisy speech samples, respectively. Further details regarding to speech databases can be found in Section 3.5.1.

After introducing the concept of speaker recognition, and providing a description of a typical recognition system, the following sections aim to introduce the human speech production mechanism and signal processing techniques for speaker recognition applications.

A spoken language is a naturally produced language, where the speech sounds are uttered by humans [20]. It's a natural way of communication between people, which carries information that allows us to interact with each other and as well as machines. Spoken language processing aims to extract and use the information carried by uttered speech signal for tasks such as speech coding, speech recognition, speaker recognition, and language recognition [20]. The required type of information, such as speaker characteristics, can be extracted using speech signal-processing methods. The feature extraction process represents the digital speech signal as a set of numerical descriptors called feature vectors. The feature extraction provides compact representation of the raw speech material (i.e. data reduction process). Extracted feature vectors (i.e. parameters) then can be used in applications such as speech coding and speaker recognition. In speaker recognition task, the feature vectors of a speaker are analysed to determine the speaker's identity.

The next section presents a review of the human speech production mechanism, which describes the production of sounds that are used for speech signal analysis. Later, the signal pre-processing and feature extraction parts are described, which are necessary building blocks of a speaker recognition system. The chapter concludes with the definitions of speech signal enhancement methods and distance measures. The speech signal enhancement methods aim to improve the speech quality as well as speaker recognition quality by removing the additive and convolutional disturbances, such as background noise and microphone transducer effects respectively [21]. In general, the distance measures are used to compare two feature vectors to determine their similarity for pattern classification purposes [3].
2.4 Speech Production

2.4.1 Human Speech Production Mechanism

Each person has their own unique speech production system in terms of physiological and behavioral factors, and this unique system produces the speaker-specific characters. The anatomy of the human speech production mechanism is given in Figure 2.4.

![Anatomy of human speech production](image)

Figure 2.4: Anatomy of human speech production

The speech production mechanism can be described by dividing the speech production organs into three main groups as the lungs, larynx, and the vocal tract [21]. The vocal tract shape is the most important physiological aspect of the human speech production system. The vocal tract consists of laryngeal pharynx, oral pharynx, nasal pharynx, oral cavity and nasal cavity. Physically the vocal tract is located between the opening of the vocal cords and the lips. The cross-sectional area may vary between zero and 20 cm$^2$ depending on the manipulation of the articulators such as jaw, velum, tongue, and lips, a process known as articulation [3]. The air is pushed from the lungs with the help of the pressure applied by the muscle system (control of the loudness). Then trachea
carries the airflow through the vocal cords from the lungs. The gap between the vocal cords is called the glottis. The glottis tends to change its shape i.e. becomes very narrow when the sound is in production, while it is normally open. Air coming from the trachea passes through the glottis and moves towards the larynx. The glottis is periodically manipulated by changing the gap size during airflow, which results in irregular airflow so called glottal source or, in other words, the source of speech [22]. The manipulation of the glottis causes vibration of the tensed vocal cords. The fundamental period is the period that the glottis changes its position from its open position to closed position (also known as the vocal cord vibration period). The reciprocal of the fundamental period is known as the fundamental frequency or the pitch. The pitch of the sound source is said to be high when the air pressure coming from the lungs are high, and the pitch of the sound is said to be low when the air pressure coming from the lungs are low. The formants, also known as the formant frequencies, describe the resonance frequencies of the vocal tract.

The sounds produced by the human speech production system can be classified into three general types as voiced, unvoiced and mixed excitation sounds [23]. The speech sound is voiced if there is a vocal cord vibration during the production of that sound, which breaks the air into quasi-periodic pulses. The spectral shaping of the sound is performed at the vocal tract by articulation. Voiced speech is characterised by its periodicity and high energy. The speech sound is unvoiced if there is no vocal cord vibration during speech production. These sounds are produced by a turbulent air flow passing a constriction in the vocal tract. Whispering is produced when the constriction is in the larynx and the vocal cords are open. Unlike voiced speech, the unvoiced speech waveform is a random-like signal in time domain, and contains less energy compared to the voiced speech segments. The mixed excitation sounds are generated by an airflow passing through a constricted vocal tract accompanied by voiced excitation generated by the vocal cords. Plosive sounds are generated by blocking the airflow in the vocal track and then releasing the air. Nasal sounds are generated by the airflow passing through the nasal cavity. A detailed description of the human speech production mechanism can be found in [3, 24].
2.4.2 Synthetic Speech Production Mechanism

The source-filter model is a widely used method of representing human speech production mechanism, which is then used in speech coding algorithms to mathematically model the speech production mechanism. The source-filter model is shown in Figure 2.5.

In this model, the vocal tract is modelled as a time-varying linear filter, and the excitation is assumed to be either voiced or unvoiced. A sequence of periodic pulses separated by the pitch period are used to generate the voiced excitation. A random noise source is used to generate the unvoiced excitation.

The source-filter model assumes that the excitation is independent of modulation. Another assumption is that the vocal tract filter is linear. Even though these assumptions are not entirely true, they are made to simplify the model, providing computational savings.

After describing the speech production mechanism, the remaining sections of the chapter explain signal processing techniques that are used for speaker recognition tasks. The front-end signal processing includes the signal pre-processing and feature extraction stages, which provides speech representation (i.e. speech coefficients) to be used in speaker modeling and recognition. Parts of a typical front-end signal processing unit are explained in the following sections.
2.5 Pre-Processing of Speech Signal

The pre-processing stage prepares the speech signal for the feature extraction part by performing some initial tasks, which aim to enhance the speech, ultimately improving the performance of the recognition system. The pre-processing stage of a typical recognition system is summarised as follows:

2.5.1 Signal Pre-Emphasising

The purpose of pre-emphasising of the speech signal is to minimise the undesired characteristics of the signal by flattening the spectrum. The speech production system generally attenuates the speech signal at high frequencies. Pre-emphasis filter spectrally improves the high frequency components of the signal. The signal pre-emphasising is performed by applying the filtering shown in Equation 2.1:

$$x_p(n) = x(n) - a \cdot x(n-1), \quad 0.95 < a < 0.98$$  \hspace{1cm} (2.1)

where $x_p(n)$ represents the pre-emphasised signal, $x(n)$ represents the original signal, and $n$ represents the time index.

The pre-emphasis filtering prior to signal processing is not always applied. The user should empirically determine whether or not to apply the pre-emphasis filtering [5, 25].

2.5.2 Silence Detection

Voice Activity Detection (VAD) (also known as silence detection) is a process of identifying voice and silence parts of the speech. The feature extraction part, which is explained in Section 2.6 extracts speaker specific information from the speech signal. This information is then used during the speaker recognition process to distinguish between different speakers. When periods of silence are included to model a speaker, they reduce the performance of the recognition system by representing the characteristics of the environment rather than the actual person in that model. Therefore, it is crucial to remove silences before speaker modeling. Generally, the VAD process is performed using
an energy-based VAD detector to remove frames that are silence [2, 26]. In this work, the silence detection process was performed differently. The speech databases used in our experiments are explained in Section 3.5.1. These databases provide transcription of the uttered sentences, with the exact location of silence and speech parts. This information is used and silences are removed without any additional VAD algorithm. However, in real-life speaker recognition systems, where there is no prior information given regarding to speech/silence intervals, a VAD should be used to remove silence intervals. Another advantage of the VAD process is to reduce the amount of data to be processed during the front-end and modeling/recognition stages.

2.5.3 Frame Analysis and Windowing

As we speak, the speech signal changes in time. However, speech is assumed to be a quasi-stationary signal i.e. slowly-varying signal [3]. Therefore the speech signal is analysed using short-term signal analysis methods. In speaker recognition systems, speech is analysed on a frame-by-frame basis. The window length of a speech frame is usually chosen to be 20-30 ms. Short time segments are used to validate the speech stationarity assumption. Speech frames are then overlapped in order to capture the changes in the speech from one frame to the next. A typical frame update rate is 10 ms. Each speech frame is multiplied with a windowing function. This process is called windowing, which minimises the signal discontinuities at the end points of a speech frame. The choice of a window shape is important since the shape of each filter has a different effect on the speech samples. It is desired to have a window with a frequency response that has a narrow main lobe and small side lobes. The narrow main lobe increases the frequency resolution and provides better spectral representation, and small side lobes reduce the frequency leakages. A rectangular window is the simplest window function and defined as follows:

$$w(n) = \begin{cases} 1 & , 0 \leq n \leq N_w - 1 \\ 0 & , \text{otherwise} \end{cases}$$

(2.2)

where $N_w$ is a window length.

In the time domain, a rectangular window has a sharp discontinuity at each end, which
causes large side lobes in the frequency domain. Different windows are defined in the literature to achieve preferable side lobes, such as the Hamming window, which has smaller side lobes. The Hamming window is a popular choice for speech applications, and is defined as follows:

\[
w[n] = \begin{cases} 
0.54 - 0.46 \cos\left(\frac{2\pi n}{N_w - 1}\right) & , 0 \leq n \leq N_w - 1 \\
0 & , \text{otherwise}
\end{cases}
\]

where \(N_w\) represents the window length.

There are some other window functions with different main and side lobe characteristics, such as Hanning, Bartlett, Blackman and Kaiser [23]. Each windowed speech frame is then further processed in the feature extraction part, which produces sets of feature vectors used in modeling and recognition of speakers.

\section*{2.6 Feature Extraction}

Speech signal feature extraction or parameterisation, is a process of extracting task-specific information from the speech waveform. The main purpose of the feature extraction process is to reduce the amount of data produced during speech production for effective speaker modeling by transforming the speech signal into feature vectors that can efficiently represent the characteristics of the signal. The choice of the parameter type varies according to the aimed task. In speaker recognition tasks, it is essential that the feature extraction part produces parameters that are robust to noise, transmission effects, and mimicry, while producing low intra-speaker variability (e.g. not affected by the health of the speaker, stable through time) and high inter-speaker variability (e.g. distinguishable amongst different speakers). The short-time spectral analysis provides spectral information that can characterise the speech signal given in an analysis window (e.g. 20 ms window length) [3, 27]. Cepstral coefficients compactly represent the information conveyed in the short-time speech spectra and have been widely used in speaker and speech recognition applications [3, 21]. This chapter introduces the Mel-Frequency Cepstral Coefficient (MFCC) extraction method, which is very popular for speaker recognition applications. In this thesis, the MFCCs are used as feature vectors
for speaker modeling and recognition tasks. Another method that is introduced in this chapter is the Linear Predictive Coding (LPC) parameter extraction method. The LPC method is widely used for speech coding applications.

### 2.6.1 Linear Prediction Coefficients (LPC)

**LP Method**

The LP method models the vocal tract. The speech production system can be modelled as a linear filter with a transfer function $H(z)$ as follows:

$$H(z) = \frac{S(z)}{U(z)} = G \frac{1 - \sum_{j=1}^{q} b_j z^{-j}}{1 - \sum_{j=1}^{p} a_j z^{-j}} \quad \text{ (2.4)}$$

where $H(z)$ is a pole-zero model, with the gain factor $G$, the filter coefficients $a_j$ and $b_j$, the $z$-transform of the speech signal and the excitation signal $S(z)$ and $U(z)$ respectively.

Unlike the pole-zero model that requires computation of a solution of non-linear equations [28], the all-pole model, which requires a solution of linear equations, is preferred for its computational simplicity. When the number of poles is high enough in Equation 2.4, the transfer function $H(z)$ can be represented by an all-pole filter as follows:

$$H(z) = \frac{G}{1 - \sum_{j=1}^{p} a_j z^{-j}} \quad \text{ (2.5)}$$

Generally there is no more than 4 or 5 formants in a speech signal with 4 kHz bandwidth, and each formant is represented by two poles. It is common to use 10th order LP filter in order to effectively represent the speech production mechanism.

The time domain representation of Equation 2.5 is known as the LPC difference equation and is written as follows:
\[ s(n) = Gx(n) + \sum_{j=1}^{p} a_j s(n-j) \]  
(2.6)

Equation (2.6) says that the output \( s(n) \) can be obtained by a weighted sum of the present input \( x(n) \), and the past outputs \( s(n-j) \). The following lines describe the calculation of the filter coefficients \( a_j \) (also known as the LP coefficients or the predictor coefficients).

**Autocorrelation Method**

There are different methods described in the literature such as Autocorrelation method, Covariance method and Lattice method that are used to determine the optimal LP coefficients [23]. The autocorrelation method is the most common method for LP filter coefficient computation. The autocorrelation method calculates the LP coefficients of a windowed frame of speech with time interval of \( N \) samples. The signal is assumed to be stationary during this time interval. The optimal LP coefficients \( a_j \) can be estimated by minimising \( E \), the energy of the prediction error:

\[
E = \sum_{n=1}^{N} \hat{e}^2(n) = \sum_{n=1}^{N} \left[ s(n) - \sum_{j=1}^{p} a_j s(n-j) \right]^2
\]  
(2.7)

where \( e(n) \) is the error signal (also known as the residual signal) that is obtained by filtering the input signal \( s(n) \) with the inverse of the predictor filter. The values of \( a_j \) that minimise \( E \) can be computed by setting \( \frac{\partial E}{\partial a_j} = 0 \) for \( j = 1, 2, \ldots, p \). From this, we obtain \( p \) linear equations:

\[
\sum_{j=1}^{p} a_j \sum_{n=1}^{N} s(n-i)s(n-j) = \sum_{n=1}^{N} s(n-i)s(n) \quad \text{for } i = 1, 2, \ldots, p.
\]  
(2.8)

with \( p \) unknowns \( a_j \).

If a function \( \phi(i,j) \) as is defined as follows:
2.6. Feature Extraction

\[
\phi(i, j) = \sum_{n=1}^{N+p} s(n-i)s(n-j) \quad (2.9)
\]

Then the Equation (2.8) can be rewritten using \(\phi(i, j)\) as follows:

\[
\sum_{j=1}^{p} a_j \phi(i, j) = \phi(i, 0) \quad \text{for } i = 1, 2, \ldots, p \quad (2.10)
\]

The Equation 2.10 can be written in the matrix form as follows [21]:

\[
\phi \mathbf{a} = \mathbf{b} \quad (2.11)
\]

where \(\phi\) is a \(p \times p\) matrix with its \((i,j)^{th}\) element given by \(\phi(i, j)\), \(\mathbf{a}\) is a \(1 \times p\) vector with its \(i^{th}\) element given by \(a_i\), and \(\mathbf{b}\) is a \(1 \times p\) vector with its \(i^{th}\) element given by \(\phi(i, 0)\). The Equation 2.11 can be solved by computing each \(\phi(i, j)\), forming the matrix equation and performing a matrix inversion.

If the speech signal is windowed and the signal values are zero outside the interval of 1 to \(N\), and the speech signal is stationary, then \(\phi(i, j)\) is only a function of \(|i - j|\). Hence, \(\phi(i, j)\) can be rewritten as the autocorrelation function \(R(|i - j|)\).

Linear equations given in Equation (2.10) can be written in matrix form as follows:

\[
\begin{bmatrix}
R(0) & R(1) & R(2) & \cdots & R(p-1) \\
R(1) & R(0) & R(1) & \cdots & R(p-2) \\
R(2) & R(1) & R(0) & \cdots & R(p-3) \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
R(p-1) & R(p-2) & R(p-3) & \cdots & R(0)
\end{bmatrix}
\begin{bmatrix}
a_1 \\
a_2 \\
a_3 \\
\vdots \\
a_p
\end{bmatrix}
=
\begin{bmatrix}
R(1) \\
R(2) \\
R(3) \\
\vdots \\
R(p)
\end{bmatrix}
\quad (2.12)
\]

The matrix given in Equation (2.12) is symmetric and all the elements on each diagonal are equal. This type of matrix is called as a Toeplitz matrix and can be inverted using a well known recursive algorithm known as the Levinson-Durbin algorithm [29]. As mentioned before, there are other methods used in the literature for computation of
the optimal coefficients of the LP filter. However, the autocorrelation method used with the Levinson-Durbin algorithm is the most commonly used method that provides a practical solution to the optimal coefficient computation.

Line Spectral Frequency (LSF) Representation of LPCs

The LPC parameters, which are defined above, provide an accurate representation of the speech spectral envelope. These parameters however, generally require quantisation and interpolation during speech coding. The spectral envelope is very sensitive to the small changes in the values of the LPC parameters, such as changes introduced by the quantisation process. Another concern during quantisation is the stability of the LPC filter, and simple way of stability check of the filter using LPCs is not available. Therefore, the LP parameters are rarely quantised or interpolated directly. To overcome these problems, several alternative LP parameter representations have been used such as LSFs [30], Reflection Coefficients [31] and Log Area Ratios (LAR) [32]. The LSFs are one of the most popular alternative representations of the LP parameters and are used during this work for speaker recognition experiments using coded speech (see Chapter 5). A unique set of $p$ Line Spectral Pair (LSP) parameters can be used to describe the $p^{th}$ order stable LP filter. The following equation shows the LSFs derived from the LSP parameters:

$$LSF(i) = \frac{\cos^{-1}(LSP(i))}{2\pi T}$$

(2.13)

where $T$ is the sampling period.

The LSP parameters are defined by a $p^{th}$ order all-pole filter $H(z)$ given as follows:

$$H(z) = \frac{1}{A(z)}$$

(2.14)

where:
2.6. Feature Extraction

\[ A(z) = 1 - \sum_{j=1}^{p} a_j z^{-j} \]  \hspace{1cm} (2.15)

For an even value of \( p \), \( A(z) \) can be written as:

\[ A(z) = \frac{1}{2} (P(z) + Q(z)) \]  \hspace{1cm} (2.16)

where:

\[ P(z) = A(z) + z^{-(p+1)} A(z^{-1}) \]  \hspace{1cm} (2.17)

and

\[ Q(z) = A(z) - z^{-(p+1)} A(z^{-1}) \]  \hspace{1cm} (2.18)

By using Equation (2.15), \( P(z) \) and \( Q(z) \) can be rewritten as follows:

\[ P(z) = 1 + (a_p - a_1)z^{-1} + \ldots + (a_1 - a_p)z^{-p} + z^{-(p+1)} \]  \hspace{1cm} (2.19)

\[ P(z) = z^{-(p+1)} \prod_{j=0}^{p+1} (z - \alpha_j) \]  \hspace{1cm} (2.20)

and

\[ Q(z) = z^{-(p+1)} \prod_{j=0}^{p+1} (z - \beta_j) \]  \hspace{1cm} (2.21)

The roots of \( P(z) \) and \( Q(z) \) are \( \alpha_j \) and \( \beta_j \) respectively. The roots of \( P(z) \) and \( Q(z) \) lie on the unit circle and occur in complex conjugate pairs, with the exception of the roots at \( z^{-1} = -1 \) for \( P(z) \) and \( z^{-1} = 1 \) for \( Q(z) \). There are \( p \) unknowns to be computed, which are the roots of \( P(z) \) and \( Q(z) \). The LSP parameters are equal to the cosine arguments of these roots. The angular information is sufficient to fully describe the filter, since the roots lie on the unit circle. The LSP parameters are computed as follows:
\[ LSP(2i) = \cos(\omega_{Q_i}) \] (2.22)

and

\[ LSP(2i + 1) = \cos(\omega_{P_i}) \] (2.23)

for \( i = 0, 1, \ldots, \frac{p}{2} - 1 \).

These parameters are called Line Spectral Pairs as the angles \( \omega \) are related in pairs \( (\omega_{Q_i}, \omega_{P_i}) \).

There are different methods such as complex root method, Chebyshev series method, and ratio filter method to solve the polynomials \( P(z) \) and \( Q(z) \) [23]. The transformation of LSF parameters to LPC parameters is a much simpler task and can be performed by substituting the results obtained from Equations (2.20) and (2.21), which are the roots of the polynomials, into Equation (2.16) to create the corresponding LPC filter.

Properties of the LSFs

- Fixed range between 0 and 4000 Hz for speech sampled at 8000 Hz
- LPC filter stability is guaranteed provided that the LSFs are in increasing order
- Closely grouped LSF parameters indicate the presence of a formant as shown in Figure 2.6
- Compression algorithms such as quantisation can benefit from the inter-frame and intra-frame correlations of the LSF coefficients [33]
2.6. Feature Extraction

![LPC filter frequency response and LSFs of a voiced speech](image.png)

Figure 2.6: LPC filter frequency response and LSFs of a voiced speech

### 2.6.2 Mel-Frequency Cepstral Coefficients (MFCCs)

The MFCC extraction is a very popular parameterisation method for speech and speaker recognition tasks. Mel cepstrum filter-bank design is based on the human ear perception of the frequency of sounds, which is non-linear [34]. The filter-bank exploits the fact that the human ear perceives the information contained in the low frequency speech signal components phonetically more important than ones contained in the high frequency components [35]. The frequency resolution of the Mel-scale filter-bank reduces as the frequency increases, which places less emphasis on higher frequencies. MFCCs can be calculated using the front-end processor given in Figure 2.7.

The stages of MFCC front-end processor is described below:

**Pre-emphasis**

First the speech signal is pre-emphasised using a pre-emphasis filter to boost high frequency components as described in Section 2.5.1.
Frame-Blocking

The quasi-stationary characteristic of the speech signal allows analysis of the signal using short speech segments (i.e. short analysis window lengths). After the pre-emphasising process, the signal is blocked into short overlapping segments (see Section 2.5.3). In this study, speech analysis window length is chosen to be 20 ms and the frame update rate to be 10 ms.

Windowing

Each speech frame is then multiplied with a suitable window function to minimise the signal discontinuities at the beginning and end of the speech frames. If we represent the speech frame with $s(n)$ and windowing function with $w(n)$, then the windowed signal $y(n)$ can be written as follows:

$$y(n) = s(n)w(n)$$  \hspace{1cm} (2.24)

where $w(n)$ is the Hamming window (as mentioned in Section 2.5.3).

The window length is 160 samples for 8 kHz sampling rate and 320 samples for 16 kHz sampling rate.

Magnitude Spectrum

The next step of the front-end processor is to calculate the power spectrum of the windowed signal. First the Discrete Fourier Transform (DFT) of the windowed signal
2.6. Feature Extraction

is computed as follows:

\[ S[k] = \sum_{n=0}^{N-1} s(n)e^{-j2\pi kn/N}, \ 0 \leq k < N \]  

(2.25)

where \( S[k] \) is a DFT of the \( s(n) \), and \( N \) is the length of the analysis window (DFT length).

Then the power spectrum is computed as \( |S[k]|^2 \) for \( 0 \leq k < \frac{N}{2} \), as the magnitude square of Equation 2.25. Prior to DFT calculation, window length is doubled by zero padding (inserting zeros to the end of the speech signal) to improve the resolution in the frequency domain.

Filter-Bank

The power spectrum values are transformed from the frequency scale into Mel-scale to place less emphasis on higher frequencies [21, 36]. The magnitude spectrum is passed through a bank of filters, by weighting (multiplying) spectrum values with the filter-bank frequency responses. The Mel-scale filter bank is designed to reflect the human ear sound perception ability at different frequencies. The filter bandwidths are also known as critical bands of hearing [21]. The critical bands determine the frequency interval, where human ear perceives two different sounds affecting each other. Hence, the Mel-scale filter-bank performs mapping of linear frequencies to a representation, which simulates human ear perception of sounds. The Mel-scale filter-bank consists of number of overlapping triangular filters. Each triangular filter’s cut-off frequencies are determined by the center frequencies of the two adjacent filters. Filters with their center frequencies less than 1 kHz are linearly spaced and have equal bandwidths. Filters with their center frequencies bigger than 1 kHz are logarithmically spaced and have logarithmically increasing bandwidths. Figure 2.8 shows graphical representations of the triangular Mel-scale filter-bank frequency responses (for 4 kHz sampled signal) and Table 2.1 shows the details of the Mel-scale filter bank [37].

In this study, different set of filter-banks, defined in [38] are used. This filter-bank design represents 4 kHz sampled signal with less number of filters (i.e. 16 filters instead of
Table 2.1: Mel-scale filter bank; Filter indexes are given in first column, beginning, center and end frequencies of each filter are given in columns 2, 3, and 4 respectively.
2.6. Feature Extraction

Mel-Scale Filter-Bank

![Mel-scale filter-bank frequency responses](image)

Figure 2.8: Mel-scale filter-bank frequency responses

without loss of performance. The 24 filters of this filter-bank are located on a non-uniform frequency scale. Each filter's cut-off frequencies are determined by the center frequencies of the two adjacent filters as before. The filter-bank center frequency values, as well as beginning and end frequencies (in Hz) are given in Table 2.2.

Human auditory system resolves the frequencies non-linearly across the audio spectrum. Using Mel-frequency or another filter-bank with similar characteristics, the desired non-linear frequency resolution can be obtained. The variants of filter-bank designs, in general, insignificant from speech and speaker recognition point of view [3].

Log Energy Coefficients

The energy output of each filter is computed as follows:

\[ E_j = \sum_{k=0}^{K-1} \phi_j(k) |s[k]|^2, \quad 0 \leq j < J \]  

(2.26)

where
<table>
<thead>
<tr>
<th>Index</th>
<th>Beginning Freq. (Hz)</th>
<th>Center Freq. (Hz)</th>
<th>End Freq. (Hz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>47</td>
<td>147</td>
</tr>
<tr>
<td>2</td>
<td>47</td>
<td>147</td>
<td>257</td>
</tr>
<tr>
<td>3</td>
<td>147</td>
<td>257</td>
<td>378</td>
</tr>
<tr>
<td>4</td>
<td>257</td>
<td>378</td>
<td>510</td>
</tr>
<tr>
<td>5</td>
<td>378</td>
<td>510</td>
<td>655</td>
</tr>
<tr>
<td>6</td>
<td>510</td>
<td>655</td>
<td>813</td>
</tr>
<tr>
<td>7</td>
<td>655</td>
<td>813</td>
<td>987</td>
</tr>
<tr>
<td>8</td>
<td>813</td>
<td>987</td>
<td>1178</td>
</tr>
<tr>
<td>9</td>
<td>987</td>
<td>1178</td>
<td>1386</td>
</tr>
<tr>
<td>10</td>
<td>1178</td>
<td>1386</td>
<td>1615</td>
</tr>
<tr>
<td>11</td>
<td>1386</td>
<td>1615</td>
<td>1866</td>
</tr>
<tr>
<td>12</td>
<td>1615</td>
<td>1866</td>
<td>2141</td>
</tr>
<tr>
<td>13</td>
<td>1866</td>
<td>2141</td>
<td>2442</td>
</tr>
<tr>
<td>14</td>
<td>2141</td>
<td>2442</td>
<td>2772</td>
</tr>
<tr>
<td>15</td>
<td>2442</td>
<td>2772</td>
<td>3133</td>
</tr>
<tr>
<td>16</td>
<td>2772</td>
<td>3133</td>
<td>3529</td>
</tr>
<tr>
<td>17</td>
<td>3133</td>
<td>3529</td>
<td>3964</td>
</tr>
<tr>
<td>18</td>
<td>3529</td>
<td>3964</td>
<td>4440</td>
</tr>
<tr>
<td>19</td>
<td>3964</td>
<td>4440</td>
<td>4961</td>
</tr>
<tr>
<td>20</td>
<td>4440</td>
<td>4961</td>
<td>5533</td>
</tr>
<tr>
<td>21</td>
<td>4961</td>
<td>5533</td>
<td>6159</td>
</tr>
<tr>
<td>22</td>
<td>5533</td>
<td>6159</td>
<td>6845</td>
</tr>
<tr>
<td>23</td>
<td>6159</td>
<td>6845</td>
<td>7597</td>
</tr>
<tr>
<td>24</td>
<td>6845</td>
<td>7597</td>
<td>8000</td>
</tr>
</tbody>
</table>

Table 2.2: Mel-scale like filter bank; Filter indexes are given in first column, beginning, center and end frequencies of each filter are given in columns 2, 3, and 4 respectively.
2.6. Feature Extraction

\[ E_j = 1 \cdot V_j(k) \quad (2.27) \]

and

where \( J \) represents the number of filters used in the filter-bank, \( V_j(k) \) and \( \phi_j(k) \) represent the frequency response and the magnitude-squared response of the \( j^\text{th} \) filter respectively. The filters are normalised to compensate for the increasing bandwidths. This is followed by calculating the logarithmic (base 10) representation of each filter output by computing \( \log_{10}(E_j) \).

Energy coefficients represent spectral envelope and perform dimensionality reduction (i.e. less number of spectral vectors) [5, 39].

Discrete Cosine Transform (DCT)

The last step of the MFCC extraction method is to find the MFCCs by calculating the DCT of the log-spectral energy vector. The DCT has a decorrelation property i.e. produces decorrelated filter log-energy coefficients. This property simplifies computational costs by allowing the use of diagonal covariance matrices instead of full covariance matrices (decorrelated elements produce correlation values close to zero except for diagonal elements) during Gaussian mixture modeling as described in Chapter 3 [21]. The MFCC vectors are computed as follows:

\[ c_m = \frac{1}{J} \sum_{j=1}^{J} \cos \left( m \frac{\pi}{J} (j - 0.5) \right) \log_{10}(E_j), \quad 0 \leq m < \hat{m} \quad (2.28) \]

where \( c_m \) are MFCC coefficients, \( J \) is number of filters used in the filter-bank and \( \hat{m} \) represents the number of MFCC coefficients and usually 16 coefficients are used for 4 kHz sampled signal. The product of the MFCC extraction method is the set of feature vectors \([c_0, c_1, \ldots, c_{\hat{m}}]\). The coefficient \( c_0 \) provides the average log energy of the spectrum, which is dependent on the intensity (loudness) and varying background noise. Therefore \( c_0 \) is usually not included in the feature vector sets.
Delta Cepstrum Parameters

The source-filter model, defined in Section 2.4.2, describes the human speech production as a process shaped by two different parts, the vocal tract filter (determines the formants and overall spectral shape), and sound source (determines pitch). The speech spectrum can be realised as the product of the filter and the sound source [12]. Cepstral analysis separates the vocal tract filter from the sound source, which eliminates the effect of the source on the filter and allows better speech signal analysis. The cepstral parameters and their variants such as delta coefficients are widely used for speech and speaker recognition applications [5].

Popular parameter types used for speech and speaker recognition applications are delta parameters. Delta Cepstrum and delta-delta cepstrum are the first and second time derivatives (i.e. differences) of the MFCC vectors respectively [3, 40]. These parameters provide temporal information about the speech signal (i.e. how person’s vocal tract changes in time) and can be concatenated to the MFCCs and form a longer feature vectors to improve the recognition performance.

Some other parameters such as Linear Prediction Cepstral Coefficients (LPCC) [9], Linear-Frequency spaced filter-bank Cepstral Coefficients (LFCC) [18], and Perceptual Linear Prediction cepstral Coefficients (PLPC) [41] have been used in the literature for speaker recognition purposes. The pitch can also be used as a feature in speaker recognition tasks. The pitch provides information about the user being male, female or a child. However, the pitch can be affected by the speakers mood, and also can be altered deliberately. High level features such as prosody, phone usage, and pronunciation have been used for speaker recognition tasks and shown to produce promising results [42, 43].

2.7 Speech Signal Enhancement

The speaker recognition task becomes more challenging when the speech samples used for speaker model training and testing are contaminated by background noise, or telephone channel noise. The performance of a recognition system can be severely affected if there is not any precaution taken to minimise the undesired effects of noise. Speech
signal enhancement methods, which aim to remove the undesired effects of noise on the speech signals, can be either performed before the feature extraction process or after the feature vectors are obtained. The speech samples used for speaker recognition applications usually contain background noise such as other people's speech, office or vehicle noises. The effect of noise on speaker recognition systems becomes more visible when the training data and the testing data are collected in different conditions i.e. they are not matched. When the speaker recognition system uses training and test speech samples that are collected from different telephone lines, microphones, and/or different background noise, the speaker models and test speech samples do not match. One important example of this mismatch occurs when clean speech (i.e. no background noise) is used as training data and the noisy speech is used as testing data. The feature vectors obtained from the noisy test speech do not match with the characteristics of the clean speech speaker model created during training process.

Signal enhancement techniques aim to reduce the noise and maximise the Signal-to-Noise Ratio (SNR) of the signal, where SNR is calculated as follows [44]:

\[
SNR = 20 \log_{10} \left( \frac{V_{s(RMS)}}{V_{n(RMS)}} \right) \tag{2.29}
\]

and

\[
V_{i(RMS)} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} v_i^2(t)} \tag{2.30}
\]

where \(V_{i(RMS)}\) is the Root Mean Square (RMS) amplitude of the signal \(v_i(t)\), \(v_d(t)\) represents the desired signal, and \(v_n(t)\) represents the noise.

### 2.7.1 Pre-processing

Additive noise such as vehicular noise, has separate spectral components to the speech signal. Spectral subtraction [45, 46], Wiener filtering [47], and Minimum Mean Square Error (MMSE) spectral amplitude estimation [48, 49] methods are popular examples of speech enhancement techniques, which remove additive noise from the speech signal.
using Short-Time Spectral Amplitudes (STSA) (please refer to Chapter 6 for noise cancellation). Statistical model-based speech enhancement methods [50] use noise models for speech signal enhancement. These methods are applied before feature extraction process.

Another factor that affects the speaker recognition performance is the presence of the convolutional distortion in speech such as distortion introduced by microphone transfer functions, or transmission channels. The Cepstral Mean Subtraction (CMS) removes stationary convolutional distortion [9, 51]. In CMS, telephone channel effects are removed by subtracting the overall average feature vector from each feature vector. Relative Spectral (RASTA) processing of the signal removes time-varying convolutional distortion [51].

2.7.2 Post-Processing

When cepstral coefficients are used as feature vectors, the linear convolutional distortion becomes additive components on the vectors in the cepstrum domain. CMS [9, 51] and RASTA processing [52] applied to feature sets are the most commonly used methods for mismatched training and test conditions. Delta cepstrum, which is described in Section 2.6.2, can be used in a speaker recognition system as a channel invariant feature set [51]. Dynamic features, unlike cepstral features, show the changes in time, rather than in a frame. Hence the effects of linear channel distortion are not reflected on the dynamic features. The noise integration model [53] is a statistical modeling method, which produces a model for speech as well as noise, which is then used directly at the speaker recognition stage. Score normalisation techniques, which are applied at the test stage, such as HNORM and CNORM have also been introduced to minimise the effects of mismatched training/test data [5]. Missing feature detection and removal is another method that is designed to reduce the effect of noise (or mismatched conditions) by detecting and removing the corrupted components of the test feature vectors [54, 55].
2.8 Distance Measures

In speech signal processing or pattern matching applications, the differences between the feature vectors or classes can be found using distance measures (also known as distortion measures). There are many different distance measures that are proposed in the literature [3, 56]. Each distance measure produces different results from the others. A user should select the distance measure, which produces the best results in terms of classification or with respect to some sort of error function. Some of the popular distance measures are listed below:

Mean Squared Error

$$d(x, y) = \frac{1}{P}(x - y)(x - y)^T = \frac{1}{P} \sum_{i=1}^{P} (x_i - y_i)^2$$  \hspace{1cm} (2.31)

Euclidean Distance

$$d(x, y) = \sqrt{\sum_{i=1}^{P} (x_i - y_i)^2}$$  \hspace{1cm} (2.32)

Manhattan Distance

$$d(x, y) = \sum_{i=1}^{P} |x_i - y_i|$$  \hspace{1cm} (2.33)

where $P$ represents the dimension of the feature vectors. $x$ and $y$ represent the data vectors with $i^{th}$ elements of vectors $x_i$ and $y_i$.

Likelihood Ratio Distortion

$$d_{LR}(x, y) = \frac{y^T R_a y}{x^T R_a x} - 1$$  \hspace{1cm} (2.34)

where $x$ and $y$ represent LPC coefficients, $R_a$ represents the Toeplitz autocorrelation matrix [57]
Log-likelihood Distance

\[ d_{LLR}(x, y) = \log(d_{LR}(x, y)) \]  

(2.35)

Since LPC coefficients are not independent, the distance between two LPC vectors can be measured using the likelihood ratio distortion and the log-likelihood distance.

Weighted Cepstral Distance

\[ d_w(x, y) = \sqrt{\sum_{i=1}^{K} [f_i(x_i - y_i)]^2} \]  

(2.36)

where \( x \) and \( y \) represent cepstral feature vectors, \( K \) represents the dimension of the feature vectors, and \( f_i \) represents a weighting function [56].

2.9 Conclusion

In this chapter, biometric recognition methods have been introduced. Examples of the most popular biometric methods have been listed, and application areas for biometrics have been presented. Amongst all biometric methods, voice is a non-intrusive biometric feature, which can be easily captured, and processed without requiring very sophisticated equipment. Also voice can be transmitted over the existing telecommunication networks, thereby allowing remote authentication.

The different types of speaker recognition systems such as text-dependent/-independent systems have been introduced. Speaker identification and verification tasks have been presented. The problems associated with speaker recognition systems have been summarised. Performance evaluation and a description of speech databases of a speaker recognition system have been briefly presented. Ideally, it is desired to have a speaker recognition system, which can provide high recognition performance without being affected by factors such as the length of speech samples, and distortions on the speech database. However, in real-life applications such factors exist and cause degradation in speaker recognition performance.

Later, human and synthetic speech production mechanisms have been presented. The
2.9. Conclusion

Front-end processing for speaker recognition systems have been explained. The typical elements of a front-end processor include the pre-processing part, the feature extraction part and the speech signal enhancement part. The speech signal pre-processing uses some signal processing techniques to prepare speech signal for feature extraction part. The feature extraction part produces feature vectors that represent the speech signal in a compact way, with the desired type of information such as speaker-related characteristics. MFCC and LPC feature vector extraction methods have been explained in detail. The MFCCs are cepstral coefficients and have been used as feature vectors during speaker recognition due to their high performance [18, 58]. The LSF coefficients, which are another representation of the LPCs, have been used during experiments related to speaker recognition using coded speech. In order to improve the speaker recognition performance, the speech signal enhancement techniques have been introduced, which are summarised as pre and post-processing enhancement techniques. This has been followed by a description of distance measures that are used to measure the similarities between data sets. In order to achieve high speaker recognition performance, it is crucial to have a robust and efficient front-end processing part that produces feature vectors, which represent speaker characteristics effectively.
Chapter 3

Speaker Modeling and Recognition

3.1 Introduction

In Chapter 2, the feature extraction process, which produces a sequence of feature vectors that represents the characteristics of the speaker's voice, has been described. In this chapter, the classification process, which is a process for determining the identity of a test speaker using the speaker's feature vectors, is presented. The classification process has two stages, modeling and testing. The speaker modeling part performs speaker enrollment to the recognition system by building a specific model for every speaker. Each individual speaker is represented with a speaker model, which carries information about a particular speaker's voice characteristics, using extracted feature vectors. The speaker testing part determines the identity of the claimant speaker by calculating the utterance score, which is a method of checking the correspondence between the unknown speech utterance and the speaker model.

Chapter 3 is finalised by explaining the details of the experimental setup for text-independent speaker identification and verification baseline systems. Also, the recognition performances for identification and verification system experiments are summarised.
3.2 Parametric versus Non-Parametric Modeling

Figure 3.1 shows the basic structure of speaker recognition systems.

Figure 3.1: Comparison of feature vectors of the unknown speaker with models of the known speakers

The feature vectors of the unknown speaker utterance are used to compute some similarity score (depending on the type of modeling technique) to observe whether the feature vectors are matching with any of the enrolled speaker models. The best matching model from the enrolled speaker models is recognized as the identity of the unknown speaker. There are different methods that speaker recognition systems use for speaker modeling and testing. These methods employ either parametric (or stochastic), or non-parametric (or template) modeling of speakers. Parametric models represent data using a particular distribution type and assume a structure that is characterised by parameters. However, non-parametric models employ minimal assumptions about the probability density function \[22, 59\]. In parametric modeling, the data is restricted to be a certain distribution type, where less data is needed to characterise the model compared to the non-parametric approach.

Some of the most common methods of speaker recognition algorithms are described below.
3.3 Non-parametric Modeling

Non-parametric models learn distribution from the data and avoid making parametric assumptions about the data distribution. However, using no parametric assumptions (unlike parametric models that fit data to a restricted model) require longer training processes to estimate optimal parameters during modeling [58]. The following examples are some of the commonly used non-parametric modeling techniques for speaker recognition applications.

3.3.1 Long-Term Averaging

The long-term averaging method calculates the mean (average) and variance values of each vector component of a large number of feature vectors from known speakers. This produces two sets of vectors for each speaker model, which are the mean vector and the variances vector. The unknown speaker's model is obtained using the same procedure, that is calculating the mean and variances vectors. For identification task, the known speaker with a mean vector that has a minimum distance from the unknown speaker's mean vector is accepted as the unknown speaker's identity. For verification task, the distances between the claimed speaker and the known speaker vectors are calculated. The claimed speaker is accepted if the distances are bigger than the value of empirically determined distance threshold. The variance vectors can be used in the way that the mean vectors are used for clustering or can be used for weighting each component of the mean vectors. This method is designed to be used in text-independent speaker recognition systems. The recognition performance can easily be affected by the duration of the training and the testing utterances. The utterance lengths should be long enough to provide a good diversity of sounds. This provides better modeling of speaker voice characteristics and increases the recognition performance. More information about the long-term average based methods can be found in [21, 60]. This method treats input data as a single cluster of data, and represents the speaker's speech characteristics with a mean and variance vectors for all the sound classes. Long-term averaging may not provide information about different sound classes, i.e. multiple clusters, hence it is not very common for speech and speaker recognition purposes.
3.3.2 Vector Quantization (VQ)

Long-term averaging uses simple speaker models that are constructed by using average and variance vectors calculated over all sounds classes, without distinguishing between different sound classes. It is possible to create better modeling of speakers by computing average vectors over different sound classes. This makes speaker models less vulnerable to phonetic differences in the utterances. Therefore, speaker models can reflect more speaker-dependent information.

VQ is a lossy data compression method that aims to divide a large set of vectors into non-overlapping clusters of vectors. Each cluster is characterized by its centroid i.e. an average vector of all the vectors in the cluster. In speaker recognition, the VQ can be used as a pre-processing stage for other classifiers, for data classification, or for data compression tasks [12].

The quantisation process is a method of reducing the infinite range of sampled vectors into a finite set of possible representative vectors. The feature vectors of a speaker can be realised as a number of data clusters with centroids. The VQ can be used in speaker recognition to represent large clusters of feature vectors with VQ codebooks that are composed of a small number of representative feature vectors. For each known speaker, clustering its feature vectors generates a speaker-specific codebook and eliminates the impracticality of storing each training feature vector, hence reducing computing complexity. There are different clustering algorithms that can be used for VQ such as the Linde, Buzo, and Gray (LBG) algorithm [61] and the k-means algorithm [62]. Speaker-specific codebooks are constructed using the LBG algorithm [61] that generates a \(2^N\) entry quantiser codebook. The LBG algorithm minimises the Weighted Mean Square Error (WMSE) of a quantiser over the training vectors. The algorithm works as follows [61]:

1. Initialisation Step: First codebook's only code-vector \(C_1(0)\) is calculated as the average of the \(M\) vectors in the training database as shown below:

\[
C_1(0) = \frac{1}{M} \sum_{m=1}^{M} x_m
\]  

(3.1)
where $x_m$ is the $m^{th}$ vector from the training database. This is design stage $N = 1$.

2. Splitting Step: Each vector in the codebook $C_N$ is split into two. This generates the codebook $C_{N+1}$:

$$C_{N+1}(k) = (1 + \epsilon)C_N(k)$$

$$C_{N+1}(2^{N-1} + k) = (1 - \epsilon)C_N(k)$$

(3.2)

(3.3)

where $k = 1, \ldots, 2^{N-1}$ and $\epsilon < 1$ is a pre-defined offset. The value of $N$ is increased by 1.

3. Optimisation Step: The codebooks are optimised using a two-step iterative process.

- Partitioning Stage: Each training vector is allocated to a cluster with a code-vector $C_N(k)$, which minimises $\|x_m - C_N(k)\|^2$, where $\|$ is a norm operator.

- Updating stage: The code-vectors are updated as the average of the vectors in each cluster. This reduces quantisation error in each cluster.

The two-step process is repeated until there is no major improvement in the overall quantisation error.

4. Repeat steps 2 and 3 until the codebook with desired size has been obtained.

In the testing part, the recognition of the unknown speaker is performed by comparing the unknown speaker’s feature vectors with the codebooks of the known speakers. The user with a codebook that has a minimum accumulated distortion is chosen as the recognized speaker. For each input vector, VQ selects a single codebook, hence forces that vector to only belong to one class. Usage of non-overlapping classes can limit the performance of the recognition system by forcing a particular vector to be a member of one class only. This non-parametric speaker recognition method can be
used for text-dependent/independent identification and verification processes. More information about VQ methods can be found in [21, 23, 37, 56].

3.3.3 Neural Networks (NNs)

NN (also known as Artificial Neural Networks (ANN)) is an information processing technology that is modeled on the human brain structure [63, 64]. NN is composed of layer(s) of large number of elements called neurons that are tied together with weights. Neurons perform the following operations:

- Receive inputs from other sources
- Combine them by calculating the weighted sum of the inputs
- Perform a non-linear operation on the previous result
- Produce output.

NNs have the ability to model non-linearity, hence non-linear models can be created to obtain better representations of datasets [3]. NNs can be used for data clustering and classification. Typical application areas are pattern, speech, and speaker recognition.

Work done by [65] is described below to demonstrate a simple example for speaker recognition using NNs. In this particular work, a speaker recognition system employs one NN for each speaker.

Training of the NNs is performed by adjusting the weights of the networks so that each NN produces an output value 1 for an input that is coming from the speaker that it represents, and output value 0 for an input that is coming from any other speaker. Like previous methods, the NN of the known speaker that has highest output value for an unknown input sample is chosen as the result of the identification process. For the verification process, the output value of the NN that belongs to the speaker to be verified is compared with a threshold value and the decision is made. Another approach used, employs one large NN to represent all speakers. This NN produces one output value for each speaker. The training and the testing processes of one large NN are the
same as the single NN for one speaker approach.

The benefits of NNs include allowing the use of parallel computation for faster processing, ability to model non-linearity, and having flexible structure that allows adaptive learning (i.e. weights can be changed as the input changes) [3]. Long training time is a major disadvantage for NNs. In order to obtain the desired output value, weights are constantly updated. As the updated weights are computed, output values are recalculated. This process is repeated recursively until the desired model is developed. This empirical nature of the training process increases the computational cost and training times. There are many different factors that play role in the performance and training time of the NNs such as the number of layers, the number of neurons for each layer. It has been shown that the performance of NNs are limited compared to the parametric models [66]. Further information about NNs can be obtained from [5, 12, 63, 64].

3.3.4 Support Vector Machines (SVMs)

SVMs are binary classifiers introduced by Vapnik [67] with the idea of constructing optimal separating hyperplanes to separate classes. This can be achieved by mapping the input vectors into another feature space with high-dimension using some nonlinear mapping [67]. The SVM classifier is a binary classifier obtained by sums of a kernel function $K(.,.)$ and has the following general form:

$$f(x) = \sum_{i=1}^{N} \alpha_i t_i K(x, x_i) + b$$

(3.4)

where $x_i$ are the support vectors, $N$ is the number of support vectors, $t_i$ is a target value for each support vector and can take values 1 and -1 depending on the class that it belongs, $\alpha_i$ and $b$ are coefficients that are the solutions of a quadratic programming problem [67], $\alpha_i \geq 0$ for $i = 1, \ldots, N$ and $\sum_{i=1}^{N} \alpha_i t_i = 0$. The class decision is made by comparing the value of $f(x)$ with a threshold value. In order to obtain better linear separability of the training data with non-linear boundaries, the input space is transformed into a high-dimensional space called feature space. Even though SVMs are linear classifiers, they can be used for non-linear data separation using kernel functions. However, the choice of correct kernel function for different applications is a major
difficulty for researchers. Some of the kernel functions used in the literature are simple dot product, polynomial kernel, and Radial Basis Function (RBF) [68]. The kernel function $K$ is required to satisfy the condition known as Mercer's condition [69]:

For any function $g(x)$

$$\int g(x)^2 dx \text{ is finite} \quad (3.5)$$

$$\int K(x, y)g(x)g(y)dx dy \geq 0. \quad (3.6)$$

If it is assumed that the two-class data to be classified are separable and there is a linear class boundary, then the SVM algorithm classifies the data into two classes by detecting the maximum margin hyperplane. Figure 3.2 shows optimal separating hyperplane in two-dimensional space with a maximum margin. The maximum margin hyperplane is defined as the hyperplane that can separate two clusters of data and located in the middle of the clusters. Support vectors are the points lying on the separating hyperplane boundaries.

![Figure 3.2: Optimal separating hyperplane in two-dimensional space](image)
Some application areas of SVMs are speaker recognition, face recognition, handwritten digit recognition, and language recognition [67, 68, 70, 71]. SVMs have been directly used as classifiers to perform speaker recognition in [70, 72]. Also SVM-GMM hybrid classifiers were reported to have promising results [73, 74, 75, 76]. In these hybrid systems, the SVMs are used to classify and separate the likelihood values of client and impostor speakers [73].

More information about the SVMs can be obtained from [67, 69, 77].

3.4 Parametric Modeling

These models make strong parametric assumptions on the underlying class-conditional probability distribution. Parametric modeling assumes that the data fits into a given statistical distribution, whose parameters are adjusted to fit the input data. Hence the training process is faster compared to non-parametric models. In this section, extra emphasis is given to explain Gaussian Mixture Modeling, since it is used as the main modeling technique in this thesis.

3.4.1 Gaussian Mixture Models (GMMs)

As we speak, different factors such as vocal tract shape, glottal flow, anatomical and fluid dynamical variations affect the way that each of us produce sounds [21]. These variations make speech production non-deterministic, which can be modeled by Gaussian mixtures. The multi-dimensional Gaussian probability density functions (pdf) can be used to probabilistically represent speaker-specific spectral shapes [2, 4, 66]. GMMs are classifiers that can model any distribution. In other words, GMMs do not impose any specific distribution type constraints on the data. Each Gaussian component in a GMM is designed to characterize some broad sound classes and includes information about speaker-specific vocal tract configurations [66]. GMMs generate a probabilistic model of the set of sounds that the speaker can produce.
3.4. Parametric Modeling

Model Description

In GMM based modeling, each speaker is represented by a model $\lambda_s$, which includes parameters of the mixture density i.e. $\mu_i^s$ is the mean vector, $\Sigma_i^s$ is the covariance matrix and the mixture weights $p_i$:

$$\lambda_s = \{p_i^s, \mu_i^s, \Sigma_i^s\}, \quad i = 1, \ldots, M$$  

(3.7)

where $s$ represents a speaker from $S$ enrolled speakers i.e. $s = 1, \ldots, S$.

In speaker recognition, a Gaussian mixture density models the distribution of speaker's feature vectors and mixture models represent the speakers. The Gaussian mixture density is computed as a weighted sum of $M$ component densities as follows:

$$p(x|\lambda_s) = \sum_{i=1}^{M} p_i^s b_i^s(x)$$  

(3.8)

where $x$ is a $D$-dimensional feature vector, $b_i^s(x)$ are component Gaussian densities, $p_i$ are mixture weights, $M$ is number of mixture components, $s$ represents each known speaker and $i = 1, \ldots, M$. This is depicted in Figure 3.3.

The component Gaussian $b_i^s(x)$ is defined as follows:

$$b_i^s(x) = \frac{1}{(2\pi)^{D/2} |\Sigma_i^s|^{1/2}} \exp\left(-\frac{1}{2} (x - \mu_i^s)'(\Sigma_i^s)^{-1}(x - \mu_i^s)\right)$$  

(3.9)

where $\mu_i^s$ is a mean vector, $\Sigma_i^s$ is a covariance matrix and $p_i$ is mixture weight with a constraint $\sum_{i=1}^{M} p_i^s = 1$. $(x - \mu_i^s)'$ represents a vector transpose operation, $(\Sigma_i^s)^{-1}$ represents a covariance matrix inverse operation, $|\Sigma_i^s|$ represents a covariance matrix determinant operation [66].

Covariance matrices used in GMMs can be selected in different ways. Choice of full or diagonal covariance matrix, plus one of the following types of the matrix can be integrated together and used for speaker models:
Chapter 3. Speaker Modeling and Recognition

The diagonal covariance matrices are sufficient for speaker modeling. There are three reasons for that [5]. The speaker model with $M^{th}$ order full covariance can be represented by a model with higher order diagonal covariance. The diagonal covariance matrices are computationally less expensive as they require less calculations (i.e., full matrix inversion is not required anymore) compared to full matrices. It has been shown that models with diagonal covariance matrices can sufficiently represent models with full covariance matrices [4, 66]. In this work, diagonal-nodal covariance matrices are employed.

The GMM method uses each speaker's feature vectors to create a unique model for that speaker. During the training part, the GMM method estimates the optimal values

Figure 3.3: M component Gaussian mixture density

- Nodal covariance - Each Gaussian component has one covariance matrix
- Grand covariance - Each speaker model has one covariance matrix
- Global covariance - All speaker models have one covariance matrix.
of the GMM parameters $p_i$, $\mu_i$, $\Sigma_i$ using the iterative algorithm called Expectation-Maximisation (EM) [78].

Expectation Maximisation

A technique called Maximum Likelihood (ML) estimation is the most widely employed method of the GMM parameter estimation. ML estimates the GMM parameters that can maximise the conditional probability $p(X|\lambda_s)$ (also known as likelihood) of the GMM, where $X$ represents the training feature vectors of the speaker and $X = \{x_1, x_2, \ldots, x_T\}$. With ML estimation, the EM algorithm is employed to obtain an estimate of the GMM parameters by iteratively updating all the parameters until the likelihood of the GMM converges. EM method aims to find the estimates of the correct GMM parameters that can increase the likelihood value of the GMM by iteratively changing the parameter values. In other words, the likelihood of GMM becomes $p(X|\lambda_s^{k+1}) \geq p(X|\lambda_s^k)$ for each EM iteration, where $k$ is the iteration number. Values of the GMM parameters of each iteration can be found as follows [59, 66]:

For each GMM component i.e. $i = 1, \ldots, M$ of the speaker $S$

- Mixture weights:

$$p_i^t = \frac{1}{T} \sum_{t=1}^{T} p(i|x_t, \lambda_s)$$  \hspace{1cm} (3.10)

- Means:

$$\mu_i^t = \frac{\sum_{t=1}^{T} p(i|x_t, \lambda_s) x_t}{\sum_{t=1}^{T} p(i|x_t, \lambda_s)}$$  \hspace{1cm} (3.11)

- Variances:

$$\sigma_i^2 = \frac{\sum_{t=1}^{T} p(i|x_t, \lambda_s) x_t^2}{\sum_{t=1}^{T} p(i|x_t, \lambda_s)} - \mu_i^t$$  \hspace{1cm} (3.12)
and

- A posteriori probability

\[
p(s|x_t, \lambda_s) = \frac{p_t^s b_t^q(x_t)}{\sum_{k=1}^{M} p_k^s b_k^q(x_t)}
\]

where \( T \) is the number of training vectors, \( x_t \) is the arbitrary element of feature vector \( x_t \), \( \mu_t \) is the arbitrary element of mean vector \( \mu_t \), \( \sigma_t^2 \) is the arbitrary element of feature vector \( \sigma_t^2 \), and \( x_t \) is the vector from the set of training vectors \( X = \{x_1, \ldots, x_T\} \). The algorithm terminates if the value of \( p(X|\lambda_s^{k+1}) - p(X|\lambda_s^k) \) is equal to a convergence threshold or the maximum number of iterations defined by user. When the likelihood value is converged the EM algorithm stops, and these updated parameter values represent the speaker's model. Generally 5-10 iterations are adequate for parameter convergence.

Parameter Initialization

The GMM parameters must be initialized before EM. The mean vector initialization can be done in two steps:

- Step 1: Randomly select \( M \) feature vectors from the training database.
- Step 2: Apply one iteration of the k-means [62] algorithm to obtain the initial mean vectors.

In this work, the k-means algorithm is implemented as follows:

1. Initialise the mean vectors to be \( M \) randomly chosen training vectors.
2. Allocate training vectors to their closest mean vector.
3. Calculate the new mean vector of each class. If there is any class with no elements, then replace the mean vector of the empty class with a training vector randomly chosen from another class.
4. Repeat steps 2, and 3 until desired number of iterations or convergence is reached.

The covariance initialization is performed by using the identity matrix as an initial covariance matrix i.e. setting each diagonal element to 1 and each off-diagonal element to 0. The mixture weights can be initialized to be equally likely by setting each weight to \( \frac{1}{M} \), obtaining equally probable weights. It has been shown that the method of parameter initialization given above can provide similar recognition performance compared to other more complicated methods of parameter initialization such as HMMs to segment phonetic classes [66].

**Variance Limitation**

There is a possibility of obtaining very small variance values while training the speaker models [66]. This can be caused by noisy data or insufficient amount of training data. Use of small variance values in the speaker modeling, can affect the GMM likelihood function and reduce the recognition performance. Therefore, it is necessary to apply a variance limiting procedure during the nodal variance training process. Variance limiting can be done as follows:

\[
\sigma_i^2 = \begin{cases} 
\sigma_i^2, & \text{if } \sigma_i^2 > \sigma_{\text{min}}^2 \\
\sigma_{\text{min}}^2, & \text{if } \sigma_i^2 \leq \sigma_{\text{min}}^2 
\end{cases} \quad (3.14)
\]

where \( \sigma_i^2 \) represents the \( i^{th} \) element of the variance vector \( \sigma_i^2 \), and \( \sigma_{\text{min}}^2 \) represents the minimum variance value. The \( \sigma_{\text{min}}^2 \) value is determined empirically and typically selected to be a value between 0.01 and 0.1 for MFCCs [66]. The variance limiting must be done carefully. Too high a variance limiting value may cause masking (blocking the actual values with the defined limiting value) the variance values. This would lead to obtain poor models and degrade the performance. However, too low variance limiting value may not be sufficient to achieve the required limiting. The variance limiting process must be performed for the updated variance values obtained from EM iterations.
Model Order

Another important factor for the speaker modeling using GMMs is the determination of a model order, i.e. the number of GMM components. Wrong choice of the mixture model order might cause recognition performance degradation. A small number of components might produce poor speaker models due to improper representation of speaker characteristics. However, a large number of GMM components with limited amount of training data might not be able to reflect the speaker's characteristics and result in poor modeling [66].

The training process is followed by the testing process, which involves matching unknown input test vectors to the model of the known speakers. Speaker identification and verification processes are explained in the following sections.

Speaker Identification

A speaker identification process determines the identity of the individuals from a group of speakers. Feature vectors obtained from the unknown speaker's speech signal are modeled with each known speaker's GMM parameters. The speaker model that gives the highest likelihood value is accepted as the unknown speaker's identity. The block diagram of the speaker identification process is depicted in Figure 3.4.

Maximum a posteriori probability (MAP) classification is the method of calculating the likelihood of each known speaker. The likelihood of each speaker model given the feature vectors by Bayes' rule can be written as follows:

\[
\hat{S} = \arg \max_{1 \leq k \leq S} \Pr(\lambda_k | X) = \arg \max_{1 \leq k \leq S} \frac{p(X | \lambda_k)}{p(X)} \Pr(\lambda_k) \quad (3.15)
\]

where \( \hat{S} \) is the recognized speaker, \( X \) is the set of training vectors \( X = \{x_1, \ldots, x_T\} \), \( \Pr(\lambda_k) \) is the priori probability of speaker \( \lambda_k \), \( p(X) \) is the priori probability of the training vectors \( X = \{x_1, \ldots, x_T\} \).

In our experiments, we assume equal speaker a priori probabilities for all speakers (i.e. \( \Pr(\lambda_k) = \frac{1}{S} \), where \( S \) is the number of speakers), and also assume equal probabilities...
for training data $X$ for all speaker models (i.e. $p(X) = \frac{1}{S}$, where $S$ is the number of speakers). Then Equation (3.15) becomes:

$$\hat{S} = \arg \max_{1 \leq k \leq S} p(X|\lambda_k)$$  \hspace{1cm} (3.16)

$p(X|\lambda_k)$ can be calculated by assuming that each frame is independent from the other frames, that is the product of the likelihoods for each frame gives the likelihood of the unknown speaker:

$$p(X|\lambda_k) = p(x_1, \ldots, x_T|\lambda_k) = \prod_{t=1}^{T} p(x_t|\lambda_k)$$  \hspace{1cm} (3.17)

Then by taking the logarithm of the above equation we obtain:

$$\hat{S} = \arg \max_{1 \leq k \leq S} \sum_{t=1}^{T} \log p(x_t|\lambda_k)$$  \hspace{1cm} (3.18)

which gives us the identity of the unknown speaker.

The performance of the identification system is measured by the identification error rate as follows:

$$\% \text{ Error} = \frac{N_e}{N} \times 100$$  \hspace{1cm} (3.19)

where $N_e$ is the number of misclassified speaker tests and $N$ is the total number of speaker tests.

![Figure 3.4: Speaker identification system](image-url)
Speaker Verification

Speaker verification process requires correctly determining the identity of a claimed speaker (also known as the hypothesized speaker). Speaker verification systems aim to decide if the unknown speaker's feature vectors are matching with the claimed speaker's model. The recognition decision is a binary decision with possible outcomes of speaker acceptance or rejection.

There are two hypotheses defined in the verification process. Assuming that there is a set of test feature vectors \( X = \{x_1, \ldots, x_T\} \) that is extracted from the unknown speaker. The first hypothesis \( H_0 \) states:

- \( H_0 : X \) is from the claimed speaker

and the second hypothesis \( H_1 \) states:

- \( H_1 : X \) is not from the claimed speaker

It is then the result of the following likelihood ratio test that determines the binary decision of the system:

\[ \text{Likelihood ratio} = \frac{p(X|H_0)}{p(X|H_1)} \left\{ \begin{array}{ll} \geq \theta & \text{accept } H_0 \\ < \theta & \text{reject } H_0 \end{array} \right. \]  

(3.20)

where \( p(X|H_0) \) and \( p(X|H_1) \) are the likelihoods of hypotheses \( H_0 \) and \( H_1 \) respectively, and \( \theta \) is the decision threshold.

Equation (3.20) can be rewritten by replacing hypotheses notations with GMM speaker model notations as follows:

\[ \text{Likelihood ratio} = \frac{p(X|\lambda_c)}{p(X|\lambda_d)} \]  

(3.21)

where \( X \) represents the set of feature vectors obtained from the test utterance, \( p(X|\lambda_c) \) represents the likelihood of the \( X \) given that it is from the claimed speaker, \( p(X|\lambda_d) \)
represents the likelihood of the \( X \) given that it is not from the claimed speaker.

The log-likelihood ratio can be written as follows:

\[
\Lambda(X) = \log[p(X|\lambda_c)] - \log[p(X|\lambda_o)]
\] (3.22)

The elements of this subtraction are formulated as:

\[
\log[p(X|\lambda_c)] = \frac{1}{T} \sum_{t=1}^{T} \log p(x_t|\lambda_c)
\] (3.23)

\[
\log[p(X|\lambda_o)] = \log \left( \frac{1}{m} \sum_{k=1}^{m} p(x_t|\lambda_k) \right)
\] (3.24)

where \( p(X|\lambda_c) \) represents the likelihood of the test feature vectors given that it belongs to the claimed speaker, \( p(X|\lambda_o) \) represents the likelihood of the test feature vectors set given that it belongs to the impostor set, \( m \) represents the number of background speakers. The speaker verification process is depicted in Figure 3.5.

![Figure 3.5: Speaker verification system](image_url)

**Background Speaker Selection**

The speaker verification process requires obtaining the models for the hypothesized speakers and the alternative speakers. The background speakers must be chosen care-
fully in order to represent the possible alternative speakers. There are two main methods to create the alternative hypothesis modeling in the verification task. The first method uses a set of speaker-specific known speaker models to obtain the alternative model. This method requires specific set of background speakers to be used for each hypothesized speaker in the database. When the speaker database is large, this method is not very practical. It requires large storage space and increased computational complexity. For the first method, there are different approaches to choose alternative model speakers [4, 79, 80]. An example from [4] constructs an alternative speaker model from the enrolled speakers. In this particular example, the alternative speaker model is created by using combination of speakers who have a similar and dissimilar voice characteristics to the hypothesized speaker. The background speaker selection process is performed as follows [4]:

- Generate the GMMs of all speakers in the database
- Calculate the Pair-wise distance between each GMM, where Pair-wise distance can be calculated as:
  \[
  d(\lambda_i, \lambda_j) = \log \frac{p(X_i|\lambda_j)}{p(X_i|\lambda_i)} + \log \frac{p(X_j|\lambda_j)}{p(X_j|\lambda_i)}
  \]  
  (3.25)
- Collect \(n\) closest speakers and \(n\) farthest speakers for speaker to be verified
- Use \(m\) closest speakers and \(m\) farthest speakers that are maximally spread from each other \((m < n)\)

The final two stages of background speakers obtained from above process are called as maximally spread close (msc) set and maximally spread far (msf) sets respectively. Number of speakers used in the background speaker model, that is \(m\) must be carefully chosen. This leads to effective representation of the possible impostor group, while minimising computing requirements. The result of the speaker verification process is calculated using a likelihood ratio test and the claimed speaker is either accepted or rejected.
Universal Background Model (UBM)

The second method, uses one generalized alternative model for all hypothesized speakers. This speaker-independent model is constructed using a number of different speakers to represent an alternative hypothesis for all enrolled speakers. This model is called the UBM [81].

When background speaker model is represented by one large alternative speaker model, the speech used for training the model must be chosen so that it can represent the existing speaker features. Depending on the application, multiple background models can be generated. The distribution of the training and the testing data should be carefully considered in order to form the UBM. If the experiments will be gender-dependent, two single-sex UBMs i.e. male speech UBM and female speech UBM are needed. If the experiments will be gender-independent, one mixed-sex UBM i.e. male and female speech UBM is needed. Different UBMs can be used that are tailored to reflect the characteristics of the data presented in a database. This will allow better modeling of the speakers and reduce the data mismatch between training and testing speech. Generally, a UBM with a model order between 512-2048 mixtures can sufficiently represent the desired speech characteristics. Large databases are best represented with large order UBMs. Unfortunately, there is not any general method to create UBMs. UBMs are generally constructed by pooling speech from sets of speakers that can reflect the distribution of general speech feature characteristics. It is important to ensure that the UBM is not balanced favoring any subpopulation. For example, if gender-independent UBM is required, equal number of male and female speakers should be used in order to prevent a gender-biased UBM [82].

Adaptation of Speaker Model:

Large, well trained UBM provides a good representation of speech features in general. This model can be altered to represent the hypothesized speakers. The UBM parameters can be adapted using what is called MAP estimation (also known as Bayesian Learning/Adaptation) and the training speech of the speaker to model the hypothesized speaker [83]. Obtaining a hypothesized speaker model by adapting UBM parameters
provide strong coupling between these two models. This coupling provides higher recognition performance and brings a scoring method that simplifies the speaker scoring time as explained in the following lines.

The hypothesized speaker model can be obtained from the UBM in two steps:

- **Step 1:** Calculate the estimates of the count, the first moment and the second moment of the hypothesized speaker’s training data for each UBM mixture.

- **Step 2:** Adapt the model using a combination of the new estimated statistics at the first step with old statistics from UBM.

The first step allows mapping speaker’s training data probabilistically onto the UBM mixtures. The second step calculates the adapted model parameters by using the UBM parameters and training data statistics. Figure 3.6 shows the adapted speaker model process.

![Figure 3.6: Speaker model adaptation from UBM](image)

The speaker adaptation process is described as follows [82]:

Count, first moment, and second moment statistics of a particular hypothesized speaker with feature vectors \(X = \{x_1, \ldots, x_T\}\) and a UBM, are calculated as below:
3.4. Parametric Modeling

\[ n_i = \sum_{t=1}^{T} Pr(i|x_t) \]  
\[ E_i(x) = \frac{1}{n_i} \sum_{t=1}^{T} Pr(i|x_t)x_t \]  
\[ E_i(x^2) = \frac{1}{n_i} \sum_{t=1}^{T} Pr(i|x_t)x_t^2 \]

where \( n_i \) is the count, \( E_i \) is the first moment, \( E_i^2 \) is the second moment, and \( Pr(i|x_t) \) is the probability of the UBM mixture component \( i \) for the vector \( x_t \).

\( Pr(i|x_t) \) can be found as:

\[ Pr(i|x_t) = \frac{p_i^0f_i(x_t)}{\sum_{j=1}^{M} p_j^0f_j(x_t)} \]

Equation (3.26), Equation (3.27), and Equation (3.28) provide statistical information about the location of the training vectors on the UBM mixtures. On the second leg of the process, the adapted weights, means and covariance vectors are calculated as below:

\[ \hat{p}_i = \left[ \frac{\alpha_i n_i}{T} + (1 - \alpha_i)p_i \right] \gamma \]

\[ \hat{\mu}_i = \alpha_i E_i(x) + (1 - \alpha_i)\mu_i \]

\[ \hat{\sigma}_i^2 = \alpha_i E_i(x^2) + (1 - \alpha_i)(\sigma_i^2 + \mu_i^2) - \hat{\mu}_i^2 \]

where \( \alpha_i \) is the adaptation coefficient, and \( \gamma \) is a scaling factor.

\( \alpha_i \) is calculated as follows:

\[ \alpha_i = \frac{n_i}{n_i + r} \]
where $r$ is an empirically determined fixed relevance factor with values ranging between 8 and 20 \[82\]. $\gamma$ is a normalisation factor that ensures adapted weights sum to one. The adaptation coefficient is a data-dependent variable that controls the balance between old and new estimates. The model adaptation is a data dependent process. Only the mixture components of UBM that have sufficient correspondence with speaker’s training data will be adapted. As [82] indicates, a UBM is a model that represents speaker-independent wide range of speech sounds and model adaptation is a fine-tuning process that modifies UBM to represent speaker-dependent speech classes observed from training speech.

As mentioned before in Equation (3.22), the log-likelihood ratio of the hypothesized speaker is calculated using hypothesized speaker model and UBM. However, we know that in the UBM approach, the model of the hypothesized speaker is an adapted version of the UBM. This brings the use of method called fast-scoring technique. When the new test vector is used for recognition, only a small number of mixtures of large UBM will be close enough to shape the overall result of the likelihood value. Since the adapted GMM of a particular speaker is obtained from UBM, the mixtures that represent the speaker in its large mixture model will be the corresponding ones in the UBM as well. Using limited number of mixture components i.e. the top $C$ best scoring ones, will be sufficient to calculate the likelihood result. With this technique, speaker’s log-likelihood value can be found as follows:

- Calculate the likelihood values of the UBM mixtures,
- Calculate the UBM likelihood result by only using the highest scoring $C$ mixtures
- Calculate the adapted speaker model likelihood result using corresponding $C$ mixtures
- Calculate the speaker’s likelihood value

In experiments typical value used for $C$ is 5 \[82\]. As an example, this would speed up the calculation time by only requiring $M + C$ computations, instead of $M \times 2$ computations for $M^{th}$ order UBM (reducing the computation time to almost half the original computation time).
3.4. Parametric Modeling

Decision Making and Error Measures

Once the speaker’s likelihood value is obtained, the recognition decision is made. There are two possible outcomes from this decision: Acceptance or rejection of the speaker. The verification system compares the threshold value to a speaker’s likelihood value as shown in Figure 3.5. If the speaker’s likelihood value is greater than the threshold $\theta$, then the claimant speaker is accepted, otherwise the speaker is rejected. There are two types of errors for speaker verification process; these are False Acceptance (FA) and False Rejection (FR). FA is a case when impostor is accepted as a true speaker and FR is a case when a true speaker is rejected as an impostor by the recognition system.

False Acceptance Rate (FAR) and False Rejection Rate (FRR) are defined as follows:

\[
FAR = \frac{I_A}{I_T} \quad (3.34)
\]

\[
FRR = \frac{C_R}{C_T} \quad (3.35)
\]

where $I_A$ is the number of false acceptances, $I_T$ is the number of impostor verification attempts, $C_R$ is the number of false rejections and $C_T$ is the number of claimant verification attempts.

When the threshold $\theta$ is going to be selected, it is important to consider choosing a value which is going to minimise the total error score of the system. Depending on the application, different threshold values can be used i.e. if the application requires very high level of security, then $\theta$ is chosen to minimise the FA occurrences.

The well known statistic called Equal Error Rate (EER) is one way of reporting the verification score [14]. The EER value is obtained by choosing $\theta$ so that the rate of false acceptances is equal to the rate of false rejections. One of the most common ways of representing the trade-off between FAR and FRR is the Detection Error Trade-off (DET) curve [13]. An example curve is shown in Figure 3.7. The DET curve is obtained from a speaker verification experiment using male speech only. The details of the experiment can be found in Section 3.5.4.
Male Speaker Detection Performance

Figure 3.7: Detection Error Trade-off curve

3.4.2 Hidden Markov Models (HMMs)

Hidden Markov Modeling is a statistical method that can characterise the stationary and temporal properties of a signal. HMM assumes that the speech signal can be characterised as a parametric random process, and that the parameters of this random process can be estimated accurately [3]. HMMs model both the speech sounds and the temporal sequencing among these sounds.

HMMs model the speech feature vectors as a group of processes. A hidden Markov Chain (i.e. not directly observable, hence hidden process), and an observable process are two stochastic processes that are performed by HMMs. The probability of following any transition is based on the present state of the system and is not affected by the past observations, as defined by the Markov property. When constructing a model, a hidden Markov chain deals with temporal variations and an observable process deals with the spectral variations in the speech signal. The main structure of HMM is defined as a number of states with transitions between each state [56]. Changes in the speech signal are modeled by a set of states with their observation probabilities \( \left( B_i \right) \) and a sequence
of transition probabilities (\(A_{ij}\)) by a Markov chain \([84]\). This is depicted in Figure 3.8, which is an example of a three state left-to-right HMM. The spectral variability of speech signal is modeled by transition probabilities. For each state, a probability density function (pdf) such as multi-dimensional Gaussian pdf (explained in Section 3.4.1) is employed to statistically represent the feature vectors.

There are different HMM types (also known as topologies) such as ergotic HMMs, and left-to-right HMMs \([3, 85]\). Ergotic HMMs (also known as fully connected HMMs) have a property that each state is connected to all other states and it is possible to reach any other state in 1 step. The state transition probabilities of ergotic HMMs are non-zero. Another type of HMM is called left-to-right model as shown in Figure 3.8. In this model, the states move from left to right as time increases. Left-to-right HMMs are good for modeling signals with varying properties in time, such as speech. The state transition coefficients have the following property:

\[
a_{ij} = 0, \quad j < i
\]  

This means that there is no transition to states with lower indices than the current state.

HMMs can be used for text-dependent/independent identification and verification pro-
cesses. For text-dependent speaker recognition, left-to-right HMMs can be sufficient to model the speech, since both training and testing utterances are same. On the other hand, text-independent systems require a model that has a characteristic to allow the ordering of speech events to be more flexible, providing text-independency. Models with more transitions such as ergotic or circular HMMs can be used for text-independent recognition. Further details regarding to HMMs and their applications can be obtained from [3, 85, 86].

3.5 Experimental Setup

It was targeted to implement a text-independent speaker recognition system for identification and verification tasks that can provide high recognition performance and can be used as a baseline system for the work carried out in the following chapters of this thesis. As mentioned in Section 3.4.1, GMMs are used for speaker modeling and evaluations. Following sections give details of the speaker databases, the feature vector extraction, the speaker modeling and the evaluation procedures for the identification and the verification systems.

3.5.1 Speaker Databases

In order to develop and evaluate the recognition systems, it is necessary to have a speech database. There are widely used databases in the literature such as TIMIT [15], NTIMIT [16], Switchboard [87], and YOHO [19]. The National Institute of Standards and Technology (NIST) [88] in United States conducts yearly speaker recognition evaluations since 1996. It aims to provide a fair measurement ground to the researchers who want to test their system performance using a certain criterion [89] (defined by NIST), determine best speaker recognition methods, provide calibration of technical issues, and present information about the direction of ongoing research. The NIST provides yearly updated speech databases such as “2005 NIST Speaker Recognition Evaluation Corpus” to its participants. Speech corpora evaluated in this thesis are a standard American English databases TIMIT (TIMIT : Texas Instruments/Massachusetts In-
stitute of Technology) clean speech database and NTIMIT telephone quality speech database provided by the Linguistic Data Consortium (LDC) [90].

TIMIT Corpus

The TIMIT [15, 91] database contains 630 speakers from 8 different dialect regions of the United States. It was designed to provide a large speaker database with diverse speaker population and rich variety of phonemes. There are 438 male and 192 female speakers (70% male and 30% female) stored in two large folders named “train” and “test”. Each speaker has 10 speech files with an average length of 3 seconds per file. These 10 speech files are divided into 3 different groups named sa (dialect sentences), sx (phonetically-compact sentences), and si (phonetically-diverse sentences) files. There are 2 sa files per speaker i.e. sa1 and sa2. sa1 and sa2 files for same speaker are different from each other. But the sentence uttered in sa1 file and the sentence uttered in sa2 file are the same for all speakers. There are 3 si files and 5 sx files for each speaker. These files are all different from each other, and are different for each speaker. The speech files are recorded in a quiet environment with high quality microphones. All speech data are recorded in one session i.e. no intersessional variability. The speech files are recorded with a sampling frequency of 16kHz. This database provides “almost ideal conditions” to examine the performance of the recognition systems.

NTIMIT Corpus

The NTIMIT [16] database is the telephone bandwidth (≈ 300-3400 Hz) version of the TIMIT database. All speech files in the TIMIT database were transmitted through a telephone network (over local and long-distance Public Switched Telephone Network (PSTN) channels) using a carbon-button telephone handset. Hence the NTIMIT database provides speech samples that are degraded from the effects of microphone and the telephone transmission. The population of speakers and the file distribution for each speaker is same as the TIMIT database.
3.5.2 Front-End Feature Extraction

The front-end feature extraction process starts with silence detection and removal. The silence parts of the speech must be removed prior to the feature extraction process. Otherwise instead of modeling the speaker, the environment will be modeled. The TIMIT and the NTIMIT databases provide transcripts of the speech files uttered by the speakers. These transcripts include details of the uttered speech providing voice and silence intervals (i.e. speech samples for speech and silence are marked). Therefore without requiring separate Voice Activity Detection (VAD), silence parts can be removed easily from the speech files. Once silence intervals are removed from the speech files, the speech signal is analysed using 20 ms (i.e. 320 samples per window for 16kHz sampled speech signal, 160 samples per window for 8kHz sampled speech signal) speech analysis windows with 10 ms frame update rate. Each speech analysis window is multiplied with the Hamming window, which is described in Section 2.5.3, to minimize signal discontinuities in the time domain. After windowing, MFCC feature extraction, which is described in Section 2.6.2, is performed as follows:

The window length is increased from $N$ to $2N$ by zero padding i.e. adding zeros at the end of the signal to improve the frequency domain resolution. The windowed signal is transformed into the frequency domain by performing a DFT using Equation (2.25). The energy coefficients, defined in Equation (2.26), are calculated by computing the inner product of the energy spectrum values with the Mel-filter-bank coefficients. Then the logarithms of the energy coefficients are computed. Then the MFCC vectors are obtained using Equation (2.28) by evaluating the DCT of the log spectral energy vectors.

The TIMIT database was evaluated for two sampling frequencies: 16kHz version named as TIMIT16k and 8kHz version named as TIMIT8k. The NTIMIT database was evaluated for 8kHz only as its the required sampling frequency for telephone transmission. This particular version of NTIMIT is named as NTIMIT8k. Down-sampling from 16kHz signal to 8kHz signal is performed by using a software called “SoX - Sound eXchange” [92].

For 16kHz sampled speech database experiments 24-dimensional MFCC vectors, covering frequency interval of 0 to 8000 Hz, are extracted per speech window. For 8kHz
sampled speech database experiments, 16-dimensional MFCC vectors, covering frequency interval of 0 to 4000 Hz, are extracted per speech window. The NTIMIT8k experiments were carried out using only 12-dimensional MFCC vectors to cover the telephone passband of 300-3400 Hz. The zeroth MFCC component is discarded from the feature vector sets (please refer to Section 2.6.2).

3.5.3 Speaker Identification Experiments

The following section describes the baseline speaker identification system and its performance.

Identification System

The identification system experiments are performed using the closed-set approach with the TIMIT16k, the TIMIT8k and the NTIMIT8k databases. The aim of these experiments is to show the baseline system performance for closed set speaker identification system using clean wide-band and narrow-band speech and degraded narrow-band speech. All 630 speakers were used in the experimental database. For each speaker, a total of 10 sentences were used as follows:

- Five speech files concatenated together providing 15 seconds long speech for training using all sa and si files.
- Five speech files concatenated together providing 15 seconds long speech for testing using all sx files.

Initialisation

For each individual, the speaker model was constructed using 32 Gaussian mixtures. 32 mixtures are sufficient to adequately model the speakers and obtain high recognition performance [66]. The mixture weights $p_i$ were all set to be $\frac{1}{M}$, where $M$ is the number of Gaussian mixtures i.e. 32. The diagonal-nodal covariance matrices were used. Initial matrix values were set to be 1 i.e. identity matrix. The variance limitation value used was 0.01. The initial mean vectors were calculated by:
Chapter 3. Speaker Modeling and Recognition

- Randomly selecting 32 MFCC vectors from the training feature vectors.
- Performing single iteration of k-means algorithm for initial clustering.

The EM parameter estimation was used to iteratively estimate the model parameters. The maximum number of EM iterations were set to be 10. Generally 10 iterations are sufficient for the likelihood function convergence [66].

**System Performance**

After the training process is completed, model parameters of each speaker are stored in a file to construct a database of the speaker models. Once the modeling of the speakers is completed, we evaluate the system performance with speaker testing. The testing process starts with the unknown speaker’s feature vector extraction. The feature vector extraction process is performed as described in Section 2.6.2. Each speaker’s stored model parameters are used with the unknown speaker’s test vectors and the log-likelihood value in Equation (3.18) is calculated. The speaker model that gives the highest log-likelihood value is the best matching model for the unknown speaker, and the speaker of this best matching model is accepted as the identity of the unknown speaker. This task is repeated for all speakers in the database. The speaker identification performances for TIMIT16k, TIMIT8k, and NTIMIT8k databases are shown in Table 3.1.

<table>
<thead>
<tr>
<th>Database</th>
<th>TIMIT16k</th>
<th>TIMIT8k</th>
<th>NTIMIT8k</th>
</tr>
</thead>
<tbody>
<tr>
<td>% Identification Performance</td>
<td>99.8</td>
<td>98.6</td>
<td>68.3</td>
</tr>
</tbody>
</table>

Table 3.1: Results of baseline identification system

The TIMIT database has a clean, near-ideal, phonetically rich, read speech files with no intersessional variability. Consequently, the TIMIT16k database has an excellent recognition performance with 99.8% identification rate. The TIMIT8k database follows this result closely with a similar performance rate of 98.6%. The small performance drop is due to the loss of high frequency components.
The degraded speech of NTIMIT8k database has an identification performance of 68.3%.

The large drop in the identification performance rate of the NTIMIT8k database is caused by the transmission effects, which are band-limitation, noise addition, and nonlinear distortion introduced by handset microphone effects. The band-limiting of speech, the added noise (i.e. background noise and the noise generated in the microphone) and the training and the testing condition mismatches altogether affect the recognition performance. When the training and the testing data are collected from different environments i.e. speech transmitted through a different PSTN channels, the recognition system suffers from an acoustic mismatch between training models and testing vectors. The data mismatch is an important factor that reduces the speaker recognition performance and will be explained in the following chapter in detail. All these results are consistent with the results obtained in the literature [4, 93].

3.5.4 Speaker Verification Experiments

The baseline speaker verification system and its performance are described in the following sections:

Verification System

The verification system experiments are performed using the TIMIT8k and the NTIMIT8k databases. These experiments show the baseline system performance of the speaker verification system using clean and degraded narrow-band speech. The speech databases used in speaker verification experiments are prepared as follows:

The verification system uses the UBM technique described in section 3.4.1 to model the background speakers for the alternative hypothesis modeling.

Both the TIMIT8k and the NTIMIT8k databases were processed using the same approach as described below. For each speaker, a total of 10 sentences are used as follows:
• Eight speech files concatenated together providing approximately 24 seconds long speech for the speaker modeling using all sa and si files, and first three sz files.

• Two speech files, each approximately 3 seconds long were used separately to perform the verification tests using last two sz files.

This provides a total of 336 male and female claimant speaker tests (56 female and 112 male speakers with 2 test sentences each). For each claimant speaker, the impostor attacks were performed using every other (i.e. impostor speaker) speaker’s two test sentences. This gives total of 31024 male and female impostor attack tests (for female speakers 56x55 per test sentence, and for male speakers 112x111 per test sentence).

Background Speaker Selection

In these experiments, we use two gender-dependent background models for our verification system. The test folder of the TIMIT (and also the NTIMIT) database does not contain equal number of male and female speakers. Therefore, separate male and female verification experiments are performed. Having two separate UBM models for male speakers and female speakers ensures that the final models are not biased towards favoring either gender. These two GMM-UBMs are created using one hour of speech per gender. It has been shown by [82] that using one hour of speech to create background speaker model is satisfactory. 120 male and 120 female speakers were used (using all 10 speech files of each speaker) to model the GMM-UBMs. One GMM-UBM is made of 1024 Gaussian mixtures. The speech files for training the GMM-UBM models are taken from the “train” folder of the TIMIT and NTIMIT databases for each experiment. The “test” folder of the TIMIT and the NTIMIT databases have 112 male speakers and 56 female speakers each.

Initialisation

Each gender-dependent UBM was constructed using 1024 Gaussian mixtures. 1024 mixtures are sufficient to adequately model the alternative speakers and obtain high recognition performance [39, 82]. The mixture weights $p_l$ were all set to be $\frac{1}{M}$, where
3.5. Experimental Setup

M is the number of Gaussian mixtures (1024). The diagonal-nodal covariance matrices were used. Initial diagonal matrix values were set to 1. The variance limitation value used was 0.01. The initial mean vectors were calculated as follows:

- Randomly select 1024 MFCC vectors from the training feature vectors.
- Perform single iteration of k-means algorithm for further clustering.

The iterative EM parameter estimation was used to estimate the UBM parameters. For such a large model, the parameter training requires larger number of the EM iterations for the likelihood value convergence. The maximum EM iterations were set to be 20 for the model parameter estimation. Usually 20 iterations are sufficient for the UBM likelihood function convergence [39].

System Performance

After obtaining two gender-dependent UBMs for male and female speakers, each speaker model was created by adapting the UBM parameters as described in section 3.4.1. The speaker models stored in a file to construct a database of speaker models. At the testing stage, the unknown speaker’s feature vectors were extracted. The fast scoring technique described in section 3.4.1 was used. The top 5 scoring mixture components were used for the speaker likelihood value computation. The verification results are reported as the EER values and the DET curves for male and female speakers separately. The verification performance obtained using the TIMIT8k and the NTIMIT8k for male speakers are shown in Table 3.2.

<table>
<thead>
<tr>
<th>Database</th>
<th>TIMIT8k</th>
<th>NTIMIT8k</th>
</tr>
</thead>
<tbody>
<tr>
<td>% EER</td>
<td>1.34</td>
<td>11.16</td>
</tr>
</tbody>
</table>

Table 3.2: EER results of baseline verification system for male speech using TIMIT8k and NTIMIT8k databases

The verification performance results using TIMIT8k and NTIMIT8k for female speakers are shown in Table 3.3.
Table 3.3: EER results of baseline verification system for female speech using TIMIT8k and NTIMIT8k databases

<table>
<thead>
<tr>
<th>Database</th>
<th>TIMIT8k</th>
<th>NTIMIT8k</th>
</tr>
</thead>
<tbody>
<tr>
<td>% EER</td>
<td>1.79</td>
<td>12.50</td>
</tr>
</tbody>
</table>

The DET curves of male and female speech speaker verification using the TIMIT8k and the NTIMIT8k databases are depicted in Figure 3.9 and Figure 3.10 respectively.

Figure 3.9: DET curve of baseline verification system for male speech using TIMIT8k and NTIMIT8k databases

The verification system produces high recognition performance with the EER values of 1.34% for male and 1.79% for female speech using the clean speech database TIMIT8k. The telephone quality NTIMIT8k database has the EER values of 11.16% for male and 12.50% for female speech. The EER value drop of the NTIMIT database is caused by the band-limiting (compared to the TIMIT8k), the noise and the training and testing condition mismatching introduced by the microphone and telephone transmission.
3.6 Conclusion

This chapter has presented the most common speaker modeling techniques used for speaker identification and verification systems. The GMMs are well known for their high performance speaker recognition rates [5, 14, 66, 82]. Since the research work presented in this thesis is using GMMs as a modeling and a matching technique, speaker recognition using the GMM method is explained in detail. The speaker model description, initialisation and training processes, and some practical issues such as variance limitation have been presented. The speaker identification and verification processes of GMM method have been described. The verification process using adapted GMM method, and speaker selection process for UBM were also mentioned, which is followed by the description of the decision making strategy and some error measures.

Later, the baseline identification and verification systems have been described. The TIMIT and the NTIMIT corpora, their content and the experimental preparation stages have been explained. Also the recognition performance of the baseline iden-
tification and the verification systems have been presented. Both the identification
and the verification systems produce very high recognition rates with clean speech.
However, recognition performance drops drastically when the telephone/noisy speech
is used. The training and the testing environment mismatching problem have been
introduced and will be explained in the following chapter.
Chapter 4

Coded Speech Speaker Recognition

4.1 Introduction

The increasing interest in mobile communications leads to a higher demand for speaker recognition applications which use coded speech. Typical application examples, where the coded speech is used in speaker recognition systems, include transaction authentication such as telephone banking, and law enforcement for identifying suspects. In these applications, there is a loss in speaker recognition performance due to the speech compression carried out by the speech coders. This recognition performance loss increases when the quality of the speech coder (i.e. bit rate) decreases [94, 95].

Speaker recognition systems perform well when the training and the testing environmental conditions are identical (known as the matched conditions). On the other hand, different environmental conditions (known as the mismatched conditions) for the training and the testing speech result in a speaker recognition performance degradation [94, 95, 96, 97]. When the mismatched coded speech data is used during speaker modeling and testing (i.e. training and testing speech data are collected from different coders), the degradation in speaker recognition performance increases further.

This chapter investigates the influence of speech coding on text-independent speaker
verification system performance. The speaker verification performance using matched and mismatched coded speech training and testing cases are investigated. A Speech Coder Detection (SCD) system that distinguishes between the speech coder type of coded speech samples is described. The performance of the SCD system is evaluated and factors determining the performance of the SCD system are investigated.

4.2 Data Mismatch

When telephone speech is used to perform speaker recognition, factors such as different acoustic environments, and communication channels can cause mismatch between the training and the testing data. This data mismatch results in a recognition performance loss [94, 95, 96, 97]. One source of data mismatch is caused by the telephone handset variability (i.e. different types of microphone used in different handsets). Different types of handset microphones have different frequency responses. Therefore the effect of spectral shaping applied to the speech signal by each microphone is different from the others. As an example, imagine a scenario where the enrollment speech of a speaker is gathered from an electret microphone handset. The speaker model will of course reflect the speaker voice characteristics that are affected by the electret microphone handset. If the speaker's test speech is collected from a carbon-button microphone handset, the speaker's test vectors will reflect the distortions introduced by a carbon-button microphone handset. Therefore, the characteristics of the model and the test vectors of the speaker will not match. In such cases, the performance degradation in speaker recognition is caused by handset type biased speaker models. These models favor the test vectors collected from same particular handset type, and if the test vectors are collected from a different handset, the recognition performance drops.

Unlike linear channel effects, the handset transducer effects are nonlinear [98, 99]. Hence it is difficult to eliminate these nonlinear effects at the front-end stage using linear channel compensation methods such as CMS (please refer to Section 2.7.2). There are different approaches in the literature in order to reduce the effects of the handset type on the recognition performance such as the handset detector and handset-dependent score normalisation technique of [82].
4.3. Speech Coders

The other effects of the mismatch can be introduced by the telephone line and switching equipment, background noise, and handset microphone noise.

The speech collected from the Public Land Mobile Networks (PLMN) channels suffer from the distortions introduced by the speech coding. The speaker recognition performance deteriorates when the coded speech is used for the speaker modeling and the testing processes [94, 95, 100, 101, 102, 103]. It has been observed that the amount of speaker recognition performance degradation is relative to the speech coder bit rate [94, 95, 101, 103]. The details of the speech coding and the data mismatch introduced by the speech coders are described in the following sections.

4.3 Speech Coders

In mobile communication systems such as cellular telephony, the speech compression methods allow efficient use of the limited bandwidth (using 8 kHz sampled speech). The speech coding is simply a process of compressing/decompressing the speech signal for transmission using speech coders (also known as codec). At the encoder, the speech signal is compressed to reduce the required transmission channel bandwidth or the storage capacity. At the decoder, the received signal is decompressed to reconstruct the lossy version of original speech signal. Different speech coding algorithms operate at different bit rates. The speech coders with medium to high bit rates (such as 8-16 kb/s) can produce high quality speech, while coders with low bit rates (such as below 1 kb/s) produce low quality robotic sound. The work presented in this chapter uses four different speech coders with different bit rates. With this approach, it is aimed to show the effects of speech coders with various bit rates on speaker recognition performance, covering a range of different applications. The speech coders used in this work are MELP (2.4 kb/s) [104], G723.1 (5.3 kb/s) [105], G729 (8kb/s) [106], and GSM-AMR (12.2 kb/s) [107]. The following sections provide brief information about each speech coder. The details given here are very brief, providing short description of each coder for our speaker verification and coder detection experiments.
4.3.1 Mixed-Excitation Linear Predictive (MELP) Speech Coder

The MELP speech coder [104, 108, 109] is the U.S. Federal Standard coder operating at a bit rate of 2.4 kb/s. It is classified as a very low bit rate speech coder. The MELP coder uses traditional LPC analysis to model the short-term spectrum. It has additional features such as mixed pulse and noise excitation (as the name mixed excitation implies), aperiodic pulses, adaptive spectral enhancement, pulse dispersion filtering, and Fourier magnitude modeling [109] to improve its performance. The coder uses mixed-voicing decision. This is achieved by splitting the input frame into sub-bands, and measuring the long-term correlation in each frequency band to distinguish between voiced and unvoiced sections. Some of the MELP coder characteristics are as follows:

- Bit rate: 2.4 kb/s
- Frame size: 22.5 ms
- Sampling Rate: 8 kHz
- High-pass filtering at the beginning of the encoder with a cutoff frequency of 60Hz
- 10<sup>th</sup> order linear prediction analysis at the encoder
- Generally used by military and commercial communication systems that do not require high quality speech while communicating

4.3.2 G723.1 Speech Coder

The G723.1 speech coder [105] is the standard coder of the International Telecommunication Union (ITU). This coder can operate in two bit rates of 5.3 kb/s and 6.3 kb/s. The 5.3 kb/s version of the coder is an Algebraic Code Excited Linear Prediction (ACELP) coder [23, 110]. The 6.3 kb/s version of the coder is Multi-Pulse LPC (MPLPC) coder [23, 110].

Some of the G723.1 coder characteristics are as follows:
4.3. Speech Coders

- Bit rates: 5.3 and 6.3 kb/s
- Frame size: 30 ms with four 7.5 ms sub-frames
- Sampling Rate: 8 kHz
- DC element of the input speech is removed by using high-pass filtering at the encoder with 30 Hz cut-off frequency
- 10th order linear prediction analysis at the encoder for both rates
- Post-filtering at the end of decoder to improve the quality of the synthesized signal. G723.1 has an adaptive post-filter that performs formant and pitch post-filtering
- Generally used for video conferencing and Voice-over-Internet applications

4.3.3 G729 Speech Coder

The G729 speech coder [106] is the standard coder of the International Telecommunication Union (ITU). The coder employs Conjugate-Structure ACELP (CS-ACELP) [23, 110] at the encoder.

Some of the G729 coder characteristics are as follows:

- Bit rate: 8 kb/s
- Frame size: 10 ms
- Sampling Rate: 8 kHz
- Pre-processing component performs signal scaling (at the encoder the speech signal is scaled to reduce the possible overflows in the implementation) and high-pass filtering (with 140 Hz cut-off frequency) before the encoding process.
- 10th order linear prediction analysis at the encoder
- Post-processing component performs adaptive post-filtering, high-pass filtering (with 100 Hz cut-off frequency) and signal up-scaling (the up-scaling is performed to restore the input signal level) operations.
Chapter 4. Coded Speech Speaker Recognition

- Generally used for digital cellular communications and Voice-over-Internet applications

4.3.4 GSM-AMR Speech Coder

The Global System for Mobile Communications-Adaptive Multi-Rate (GSM-AMR) speech coder is a standardised coder introduced by the 3rd Generation Partnership Project (3GPP) for digital cellular applications [107, 111]. The variable bit-rate allows the coder to perform bit allocation according to the channel noise. If the channel is noisy, more bits are allocated for channel coding than the bits allocated for speech coder. The GSM-AMR coder uses ACELP to encode the speech signal using one of 8 different bit rates. Some of the GSM-AMR coder characteristics are as follows:

- Bit rates: 4.75, 5.15, 5.90, 6.7, 7.4, 7.95, 10.2, 12.2 kb/s
- Frame size: 20 ms with four 5ms sub-frames
- Sampling Rate: 8 kHz
- The 12.2 kb/s coder is the same as the GSM Enhanced Full-Rate (EFR) coder [112, 113] of the European Telecommunications Standards Institute (ETSI). 10th order linear prediction analysis is performed utilising two asymmetric 30 ms wide analysis windows (please refer to Section 5.3.3)
- Uses 10th order linear prediction analysis at the encoder utilising one analysis window for the other bit rates
- Generally used for applications that require high quality speech such as digital cellular communications

4.4 Speaker Recognition Performance Using Coded Speech

This section aims to demonstrate the degradation in the speaker recognition performance introduced by the use of coded speech during speaker modeling and testing.
Four different speech coders with various bit rates were used. The AMR coder (12.2 kb/s), the G729 coder (8 kb/s), the G723.1 coder (5.3 kb/s), and the MELP coder (2.4 kb/s) were chosen in order to create a speaker verification scenario using various bit rates and speech coder qualities. All these coders use 8 kHz speech as an input. Therefore we used down-sampled version of the TIMIT database (TIMIT8k) during our experiments. The clean speech files of the TIMIT8k were coded (i.e. encoding and decoding) with each of the four speech coders. The coded versions of the TIMIT8k database are named as follows:

- TAMR : AMR 12.2 kb/s coded TIMIT8k database
- TG729 : G729 8 kb/s coded TIMIT8k database
- TG723.1 : G723.1 5.3 kb/s coded TIMIT8k database
- TMELP : MELP 2.4 kb/s coded TIMIT8k database

4.4.1 Matched and Mismatched case Definitions for Speech Coding

Figure 4.1 shows a speaker recognition system, which is accessed by different clients using various networks. In this scenario, the speaker recognition system is accessed by customers through various networks such as PSTN and PLMN. If a speaker is enrolled using his mobile phone, but he uses a land-line phone during authentication, a mismatch between training and testing speech due to different coding algorithms, and different communication channels is unavoidable.

In real life applications, there are number of different combinations of coded and uncoded speech that can be used during the training and the testing processes.

There are two main conditions that can be encountered during recognition processes. These conditions are namely the matched condition and the mismatched condition. The matched condition indicates that the training speech (both the UBM speech and the adaptation model speech) and the testing speech of a particular speaker are derived from the coded speech using same coder.
The mismatched condition indicates that the training and testing speech samples of a particular speaker are derived from different sources (i.e. one from uncoded speech and the other from coded speech). The mismatched condition can take different forms. In a partially mismatched situation, the background and the claimant speaker models are obtained using speech samples collected from the same source and the test data collected from a different source. In a fully mismatched situation, the background speaker model is obtained using speech collected from a different source, where the claimant speaker models and the test data are samples collected from the same source. Table 4.1 summarises the possible cases of mismatched conditions of training and testing data of a particular speaker.

For the fully mismatched conditions, the background speaker model and the claimant speaker models are collected from different sources, hence a higher degree of mismatch is present. It is expected to obtain a higher drop in speaker recognition performance when compared to the partially mismatched conditions [94].
4.4. Speaker Recognition Performance Using Coded Speech

Table 4.1: Possible versions of the mismatched conditions for speaker modeling and testing

<table>
<thead>
<tr>
<th></th>
<th>Partially Mismatched Case</th>
<th>Fully Mismatched Case</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speech</td>
<td>UBM</td>
<td>Claimant</td>
</tr>
<tr>
<td>Case 1</td>
<td>Uncoded</td>
<td>Uncoded</td>
</tr>
<tr>
<td>Case 2</td>
<td>Coded</td>
<td>Coded</td>
</tr>
</tbody>
</table>

In our experiments the following define the matched and the mismatched cases:

- **Matched Case**: The UBM speech, the adaptation model speech and the test speech of the speaker are all coded speech and derived from the same coder.

- **Mismatched Case**: The UBM speech and the adaptation model speech of the speaker are derived from the uncoded speech. The test speech of the speaker is derived from the coded speech.

**Verification Performance for Coded Speech**

The speaker verification experiments for all four coders are performed using the matched conditions and the mismatched conditions defined above. The resultant EER values are compared with the EER value of the baseline verification system. The baseline system is trained and tested using uncoded speech for the UBMs, the adaptation models and the speakers test data. The speaker verification experiments are carried out using the same methodology as described in Section 3.5.4. The values of EER for male and female speaker verification experiments for the matched and mismatched cases are shown in Figure 4.2 and Figure 4.3. Also the EER values of the baseline systems are included for better comparison.
Chapter 4. Coded Speech Speaker Recognition

Male Speakers

Baseline System
Matched Case
Mismatched Case

Figure 4.2: EER values for male speaker verification experiments using four different coders with matched and mismatched conditions

Female Speakers

Baseline System
Matched Case
Mismatched Case

Figure 4.3: EER values for female speaker verification experiments using four different coders with matched and mismatched conditions
The results indicate that the speaker verification performance is related to the bit rate of the coder that is used for coded speech. It can be observed from Figures 4.2 and 4.3 that when the speech coder bit rate decreases, the speaker verification performance decreases too. The only exceptional result here appears to be the MELP coder. The MELP coder EER values of male and female speaker experiments are lower compared to the G723.1 speech coder EER values of male and female speakers. We believe that in the mismatched case experiments, the low-pass filtering applied by both G729 and G723.1 coders cause an increase in the level of mismatch between training and test data, effectively reducing the verification performance. In the matched case, however, it is possible that this performance difference is due to the analysis frame size difference between these two coders. The G723.1 speech coder has a 30 ms analysis frame window. But the MELP coder has a 22.5 ms analysis frame window. The shorter analysis windows allows capturing changes in the speech signal more efficiently. Hence the MELP speech coder can achieve better speaker recognition performance compared to the G723.1 speech coder. Similar results are also obtained by [39]. It has been observed that apart from the bit rate, the spectral shaping of each speech coder plays an important role determining the SCD system performance. This issue is explained in the following sections.

4.5 Data Mismatch of Multi-Coder Situation

The results obtained in the previous section confirm that the speaker recognition performance deteriorates when coded speech is used for speaker modeling and testing processes. But this is not the only effect imposed by the speech coders on speaker recognition performance. As mentioned before, the coder mismatch causes further reduction in recognition performance. An example scenario can be described as the speaker model (including both the UBM and the claimant speaker model) derived from the GSM-AMR coded speech while the speaker test data derived from the MELP coded speech. This combination of course produces lower performance compared to the matched case of both the speaker model and the speaker test data derived from the GSM-AMR coded speech. The effect of the coder mismatch problem gets even worse when the speaker
recognition system trained for one particular coder type uses test speech obtained from a number of different speech coders [39, 114, 115]. Ergun et al. proposed a method called the Speech Coder Detection (SCD) system to reduce/eliminate the verification performance degradation due to the coder mismatch problem [114]. It has been shown that the use of the SCD in a multi-coder environment provides speaker recognition performance similar to the performance of the matched coder-type environment.

### 4.6 Speech Coder Detection System

The SCD system determines the type of the speech coder used for the coded test speech. This allows the use of appropriate coded speech UBM and the claimant speaker model (i.e., the same coder used for input test speech) during verification. The SCD process is based on the coded speech modeling. The individual coder model is created by constructing a UBM, using speech samples coded with that particular coder. Each coder model is denoted by the notation \( \{\lambda_{\text{UBM},k}\}_{k=1}^{K} \) where \( K \) is the number of coders to be detected by the SCD system. The SCD system is designed to perform detection over \( K \) different coder types. This allows SCD implemented speaker verification system to be accessed by \( K \) different networks using \( K \) different speech coders. Figure 4.4 shows a combined SCD/speaker verification system.

The speaker verification system receives the coder type information from the SCD system and uses coder-dependent UBM and claimant speaker models to calculate the log-likelihood ratio. The SCD system employs \( K \) different UBMs to represent \( K \) different speech coders. The UBMs are denoted as \( \{\lambda_{\text{UBM,1}}, \lambda_{\text{UBM,2}}, \ldots, \lambda_{\text{UBM,K}}\} \). Each claimant speaker \( s_i \) has also \( K \) different claimant speaker models. The claimant speaker models are denoted as \( \{\lambda_{C,1}, \lambda_{C,2}, \ldots, \lambda_{C,K}\} \). Each one of these models represents a particular speech coder for the speaker \( s_i \). The coder type of a coded test speech is calculated as shown in Equation (4.1):

\[
k_{\text{opt}} = \arg \max_k p(X|\lambda_{\text{UBM},k}), \quad k = 1, \ldots, K.
\]

where \( X = \{x_1, \ldots, x_T\} \) is the coded speech feature vector with length \( T \).
4.6. Speech Coder Detection System

Speech Coder Detector

Speech Coding

Speech Coder Detector

Coder-Dependent Reference UBMs

AMR
G729
G723.1
MELP

Feature Extraction

Select the coder with maximum likelihood

Detected Coder Label

(UBM)_{MAX}

(Claimant Speaker Models)_{MAX}

Speaker Verification System

Accept Speaker
if \( A > \theta \)

Figure 4.4: Combined speech coder detection and speaker verification system
Once the coder type of the test speech is detected, the speaker verification system uses the claimant speaker model and the UBM model of that coder for the log-likelihood calculation. The log-likelihood value for a coded speech test vector $X$ is computed as follows:

$$ A(X) = \log \left[ p(X|\lambda_{C,\text{opt}}) \right] - \log \left[ p(X|\lambda_{\text{UBM,\text{opt}}}) \right] $$

(4.2)

The UBMs used in the SCD system are the same UBMs created for the matched case coded speech speaker verification experiments. These large-order UBMs represent broad acoustic classes including the speech coder characteristics. Therefore, it is sufficient to use the same UBMs for the SCD system.

### 4.7 Speech Coder Detection System Performance

The SCD system performance is calculated using four different speech coders i.e. $K = 4$. The coders used were the AMR coder (12.2 kb/s), the G729 coder (8 kb/s), the G723.1 coder (5.3 kb/s), and the MELP coder (2.4 kb/s). Coder detection experiments were performed for male and female speakers separately. The male and the female speaker UBMs were created using the concatenated speech of 120 male and 120 female speakers respectively. Each UBM was constructed using 1024 mixtures. The number of actual claimant male and female speakers were 112 and 56 respectively. The feature vectors used for model training and testing were MFCCs. The databases used were the TAMR, the TG729, the TG723.1 and the TMELP (i.e. coded versions of the TIMIT8k database). The experimental setup is identical to the one described in the Section 3.5.4. The SCD system performance is depicted in Figure 4.5.

Figure 4.5 shows the detection rate of each speech coder for male and female speakers separately. The average coder detection performance for male and female speakers are 95.4% and 98% respectively. When Figure 4.5 is analysed, it is not possible to observe any direct relationship between the coder detection performance and the coder bit rate. [114] suggests that the speech coder detection performance is better when the bit rates
4.8 Speaker Verification Experiments with Speech Coder Detection System

The combined SCD/speaker verification system is constructed as shown in Figure 4.4. During the verification process, the coder type of a coded speech test vector is determined by the SCD system. Then the speaker log-likelihood value given in Equation (4.2) is calculated using the detected coder type UBM and claimant speaker models. Figure 4.6 and Figure 4.7 depict the average speaker verification performance of the pool of four speech coders for male and female speakers respectively.

Figure 4.5: Male and female speaker speech coder detection rate in four-coder environment using TIMIT8k database.

are lower. This relationship is true for female speakers. But the same conclusion cannot be drawn for the male speakers. At this stage, we believe that further investigation of the speech coder detection system is essential in order to determine the factor that allows the SCD to distinguish amongst different coders. This issue will be covered in Section 4.9 in detail.

4.8 Speaker Verification Experiments with Speech Coder Detection System

The combined SCD/speaker verification system is constructed as shown in Figure 4.4. During the verification process, the coder type of a coded speech test vector is determined by the SCD system. Then the speaker log-likelihood value given in Equation (4.2) is calculated using the detected coder type UBM and claimant speaker models. Figure 4.6 and Figure 4.7 depict the average speaker verification performance of the pool of four speech coders for male and female speakers respectively.
Chapter 4. Coded Speech Speaker Recognition

Figure 4.6: Male speaker DET curves of matched case verification, mismatched case verification, and combined speech coder detection/verification system using TIMIT8k database

Figure 4.7: Female speaker DET curves of matched case verification, mismatched case verification, and combined speech coder detection/verification system using TIMIT8k database
The matched case results are obtained by calculating the coded test speech likelihood value using the same coder type for the UBM, the claimant speaker model and the test speech. The mismatched case results are obtained by calculating the coded test speech likelihood value using uncoded speech for the UBM, and the claimant speaker model. The coder detection case results are obtained by calculating the coded test speech likelihood value using the speech coder type provided by SCD system for the UBM and the claimant speaker model. Each DET curve is obtained by plotting the false alarm and miss probabilities using a speaker and coder independent threshold parameter.

Both Figure 4.6 and Figure 4.7 show that the speaker verification performance of the combined SCD/verification system is very similar to the performance of the matched case verification. The EER values for both the combined SCD/verification system experiment and matched case verification experiment are 3.2% for the male speakers and 3.3% for the female speakers (the EER values obtained by [114] for both the combined SCD/verification system experiment and matched case verification experiment are 4.1% and 4.0% for the male and female speakers respectively). These results indicate that the effect of the coder mismatch case can be eliminated by using integrated SCD system in the speaker verification system.

4.9 Analysis and Observations

The speech coder detection experiment results presented in Section 4.7 are not conclusive in terms of any direct relationships between the coder detection performance and speech coder bit rates. The research work produced by [114] suggests that the speech coder detection performance is better when the coder bit rates are lower. This relationship might seem partially true for female speakers. In Figure 4.5, the female speaker coder detection performance of the 2.4 kb/s MELP coder provides the highest detection rate compared to the other three coders. The 5.3 kb/s G7231 coder also provides better female speaker coder detection performance than other two higher bit rate coders. However, for the 8kb/s G729 coder the female speaker coder detection rate is lower than the coder detection rate of 12.2 kb/s AMR coder. Also there is no directly
observable relationship between the coder detection performances of male speakers with the coder bit rates. In order to fully understand how the coder recognition system can distinguish between different speech coders, we performed further experimentation on the coder detection system.

Initially we decided to use only two coders in the coder recognition system to simplify the problem. We started our analysis using the G729 and the G723.1 coders. These coders are both ACELP coders, and their coding algorithms are very similar. This should minimise the effects of different coding methods on the result and allow us to analyse the experimental results easier.

We performed several different experiments and obtained some interesting findings. These findings are listed below as three observations recorded during our experiments:

4.9.1 Observation One: Effect of Two Coding Algorithms

Our initial approach was to observe the behavior of the SCD system when the transcoded input speech is obtained by encoding/decoding the speech using two speech coders one after the other (i.e. first transcode a clean speech through the G729 coder and then transcode this coded speech through the second coder G723.1). Table 4.2 shows the speech coder detection performance when the clean speech is first transcoded using the G729 coder and then transcoded using the G723.1 coder. Table 4.3 shows the speech coder detection performance when the clean speech is first transcoded using the G723.1 coder and then transcoded using the G729 coder.

<table>
<thead>
<tr>
<th>Detected Coder Type</th>
<th>G729</th>
<th>G723.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCD Performance (%)</td>
<td>14.3</td>
<td>85.7</td>
</tr>
</tbody>
</table>

Table 4.2: Clean female speech transcoded using the G729 coder and then transcoded using G723.1 coder

The results in Table 4.2 and Table 4.3 show that speech files that are transcoded twice mainly reflect the characteristics of the second coder. This shows that the processes performed by second coder removes the effects of first coder. The next stage is to investigate the processes performed in the coding algorithms.
4.9. Analysis and Observations

<table>
<thead>
<tr>
<th>Detected Coder Type</th>
<th>G729</th>
<th>G723.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCD Performance (%)</td>
<td>92.9</td>
<td>7.1</td>
</tr>
</tbody>
</table>

Table 4.3: Clean female speech transcoded using the G723.1 coder and then transcoded using G729 coder

4.9.2 Observation Two: Effect of Coder Components

Our next strategy was to identify the components in a coder that are detected by the SCD system, allowing coder recognition. Our approach was to keep one coder unmodified and remove a signal processing component from the other coder. The components that we decided to remove during our experiments were:

- High-Pass Filter (HPF) at the encoder (removes the DC element from the input speech)
- Post-filter at the end of decoder (improves the quality of the synthesized signal)
- Vector quantiser in the encoder (performs LSP vector quantisation)

These signal processing components are basic yet important components of a speech coder. The high-pass filter component affects the speech signal that is about to be coded. The post-filter component modifies the speech signal at the output stage of a coder. Also the vector quantiser of a coder has an important effect on the output speech signal. Figure 4.8 shows the SCD system performance of each coder in two-coder environment using female speech.

The speech coder detection performance is depicted, showing each speech coder component that is removed from the coder. As it can be seen from Figure 4.8, there is no major performance degradation in the speech coder detection rate introduced by the removal of the HPF, the post-filter and the VQ. But it can be observed that the post-filtering is the most effective component on coder detection rate.
Figure 4.8: Effect of speech coder component removal on female speaker speech coder detection rate in two coder environment using TIMITsk database

4.9.3 Observation Three: Effects of Filtering on the Coded Speech

The G729 coder has a post-processing component at its decoder. This post-processing component performs post-filtering, high-pass filtering and signal up-scaling (at the encoder the speech signal is scaled to reduce the possible overflows in the implementation, hence the up-scaling is performed to restore the input signal level) operations. The synthesized speech output of G729 coder is high-pass filtered at 100 Hz. However there is no such high-pass filtering at the decoder of the G723.1 coder. We believe that the removal of certain spectral information might play a role in the speech coder detection process. It is possible that the bandwidth of the coded speech signal may influence the speech coder detection performance. This can be experimentally observed by using different filters on the speech spectrum. The filter types that we decided to use were:

- Band-Pass Filter (BPF) with cutoff frequencies at 100 Hz and 3500 Hz
- High-Pass Filter (HPF) with cutoff frequency at 100 Hz
- Low-pass Filter (LPF) with cutoff frequency at 3500 Hz
4.9. Analysis and Observations

Two-Coder Environment:

The SCD system performance experiment in a two-coder environment was performed using three different versions of the G723.1 coded speech as a test data. Each different input test data was obtained by using one of the filters described above. One set of test data was obtained by filtering the G723.1 transcoded speech files with band-pass filter (cut-off frequencies at 100 Hz and 3500 Hz). The second set of test data was obtained by filtering the G723.1 transcoded speech files with high-pass filter (cut-off frequency at 100 Hz). The third set of test data was obtained by filtering the G723.1 transcoded speech files with low-pass filter (cut-off frequency at 3500 Hz). The three separate test input vectors were extracted from these filtered speech files. The coder detection experiments were performed for male and female speakers separately. The gender-dependent speaker UBMs were constructed using the concatenated speech of 120 male/female speakers. Each UBM was created using 1024 mixtures. The number of claimant male and female speakers were 112 and 56 respectively. The experimental setup is identical to the one described in the Section 3.5.4.

Table 4.4 and Table 4.5 show the SCD system performance for G723.1 coder using male and female speech.

<table>
<thead>
<tr>
<th>Actual Coder Type</th>
<th>Detected Coder Type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>G723.1</td>
</tr>
<tr>
<td>G723.1</td>
<td>96.0</td>
</tr>
<tr>
<td></td>
<td>1.8</td>
</tr>
<tr>
<td></td>
<td>1.8</td>
</tr>
<tr>
<td></td>
<td>96.0</td>
</tr>
</tbody>
</table>

Table 4.4: G723.1 coded male test speech SCD performance (in %) in two coder environment using G723.1, and G729 coders with different filtering applied to coded speech

Table 4.4 shows the SCD system performance in two-coder environment using G723.1 coded test input speech, which is filtered with BPF, HPF and LPF, respectively. When no filtering applied to the input test speech, correct coder type detection rate is 96.0%.
When the G723.1 coded test input speech is band-pass filtered, the correct coder type detection rate is reduced to 1.8%. The SCD system detected the band-pass filtered G723.1 coded speech files as G729 coded speech. Also the coder type of the high-pass filtered G723.1 coded test input speech files are detected as the G729 coder. On the other hand, coder type of the low-pass filtered G723.1 coded test input speech files are detected as the G723.1 coder.

Similar results were observed for female speaker experiments. When there is not any filtering applied to the input test speech, correct coder type detection rate is 100.0%. When the G723.1 coded test input speech is band-pass filtered or high-pass filtered, the SCD system detected the filtered G723.1 coded speech files as G729 coded speech. Applying LPF to the input speech prior to speech coder detection is not effecting the result of coder detection task.

<table>
<thead>
<tr>
<th>Actual Coder Type</th>
<th>Detected Coder Type</th>
<th>Filtering</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>G723.1</td>
<td>G729</td>
</tr>
<tr>
<td>G723.1</td>
<td>100.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>1.8</td>
<td>98.2</td>
</tr>
<tr>
<td></td>
<td>0.9</td>
<td>99.1</td>
</tr>
<tr>
<td></td>
<td>99.1</td>
<td>0.9</td>
</tr>
</tbody>
</table>

Table 4.5: G723.1 coded female test speech SCD performance (in %) in two coder environment using G723.1, and G729 coders with different filtering applied to coded speech

The results shown in Table 4.4 and Table 4.5 indicate that the bandwidth of the input speech is critical in the coder type detection. The SCD system utilises the differences in the bandwidth of speech signals to decide the tested coder type. The following two sections provide the SCD system performance in three-coder and four-coder environments. The aim of these sections is to confirm the importance of the speech signal bandwidth in speech coder detection, which was shown for two-coder environment.
Three-Coder Environment:

The SCD system performance experiment in three-coder environment was performed using three different sets of the AMR coded speech: band-pass filtered set, high-pass filtered set, and low-pass filtered set. The three separate test input vectors were extracted from these filtered speech files. The experimental setup is identical to the one described in the Section 4.9.3 with exception that coder number used is three.

Table 4.6 and Table 4.7 show the SCD system performance for AMR coder using male and female speech, respectively.

Table 4.6 shows the SCD system performance in three-coder environment using AMR coded test input speech, where different type of filtering is applied to the input speech prior to coder detection task.

<table>
<thead>
<tr>
<th>Actual Coder Type</th>
<th>Detected Coder Type</th>
<th>Filtering</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>G723.1</td>
<td>G729</td>
</tr>
<tr>
<td>G723.1</td>
<td>1.8</td>
<td>2.2</td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>92.8</td>
</tr>
<tr>
<td>AMR</td>
<td>0.0</td>
<td>93.3</td>
</tr>
<tr>
<td></td>
<td>4.9</td>
<td>5.4</td>
</tr>
</tbody>
</table>

Table 4.6: AMR coded male test speech SCD performance (in %) in three coder environment using G723.1, G729, and AMR coders with different filtering applied to coded test speech

When there is not any filtering applied to the AMR coded test input speech, the correct coder type detection rate is 96.0%. Applying band-pass filtering to AMR coded test input speech reduces the correct coder detection to 6.7%. Majority of the band-pass filtered AMR coded speech files are detected as G729 coded speech. Also the coder type of the high-pass filtered AMR coded test input speech files are detected as the G729 coder. Whereas the majority of the low-pass filtered AMR coded test input speech files are detected as the AMR coder.
Chapter 4. Coded Speech Speaker Recognition

The results obtained for the female speaker experiments follow similar trend as male speaker experiments.

<table>
<thead>
<tr>
<th>Actual Coder Type</th>
<th>Detected Coder Type</th>
<th>Filtering</th>
</tr>
</thead>
<tbody>
<tr>
<td>G723.1</td>
<td>2.7</td>
<td>0.0</td>
</tr>
<tr>
<td>G729</td>
<td>0.9</td>
<td>96.4</td>
</tr>
<tr>
<td>AMR</td>
<td>0.9</td>
<td>96.4</td>
</tr>
<tr>
<td></td>
<td>1.8</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Table 4.7: AMR coded female test speech SCD performance (in %) in three coder environment using G723.1, G729, and AMR coders with different filtering applied to coded test speech.

When there is no filtering applied to the AMR coded test input speech, the correct coder type detection rate is 97.3%. Applying band-pass filtering, and high-pass filtering, to the AMR coded test input speech caused test speech coder type to be detected as G729 coder. Coder type of the low-pass filtered input speech is detected as AMR coder.

Another experiment performed in a three-coder environment was filtering the AMR test input speech with high-pass filter with a cut-off frequency at 60 Hz. The AMR coder has a high-pass filter at its encoder part. Hence in these experiments we aim to increase the coder detection rate of AMR test speech by high-pass filtering it at 60 Hz, instead of 100 Hz. Table 4.8 and Table 4.9 show the SCD system performance of male and female speakers using high-pass filtered AMR speech at 60 Hz respectively. As it can be seen in both tables, the detection rate for AMR speech is improved.

It can be concluded from the results shown in Tables 4.6, 4.7, 4.8, and 4.9 that the speech bandwidth is very important for coder type detection, which allows SCD to distinguish between different coders.
4.9. Analysis and Observations

<table>
<thead>
<tr>
<th>Actual Coder Type</th>
<th>Detected Coder Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>G723.1</td>
<td>2.3</td>
</tr>
<tr>
<td>G729</td>
<td>77.2</td>
</tr>
<tr>
<td>AMR</td>
<td>20.5</td>
</tr>
</tbody>
</table>

Table 4.8: HPF (60 Hz cut-off freq.) AMR coded male test speech SCD performance (in %) in three coder environment using G723.1, G729, and AMR coders

<table>
<thead>
<tr>
<th>Actual Coder Type</th>
<th>Detected Coder Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>G723.1</td>
<td>3.6</td>
</tr>
<tr>
<td>G729</td>
<td>78.5</td>
</tr>
<tr>
<td>AMR</td>
<td>17.9</td>
</tr>
</tbody>
</table>

Table 4.9: HPF (60 Hz cut-off freq.) AMR coded female test speech SCD performance (in %) in three coder environment using G723.1, G729, and AMR coders

Four-Coder Environment:

The coder type detection performance experiment in a four-coder environment was performed using three different sets of the MELP coded speech: band-pass filtered set, high-pass filtered set, and low-pass filtered set. The three separate test input vectors were extracted from these filtered speech files. The experimental setup is identical to the one described in the Section 4.9.3 with the exception that the coder number used is four.

Table 4.10 and Table 4.11 show the SCD system performance using MELP coded male and female speech, respectively.

Table 4.10 shows the SCD system performance in four-coder environment using MELP coded test input speech, where different filtering is applied prior to coder detection. The correct coder type detection rate for the MELP coded test input speech is 94.2%. The correct coder type detection rate is dropped to 83.5%, when band-pass filtered MELP coded test speech is used as an input to the SCD system. The SCD system detected 16.5% of the the band-pass filtered MELP coded speech as G729 coded speech. 15.2% of the high-pass filtered MELP coded test input speech files are detected as G729 coded speech. Also 92.9% of the low-pass filtered MELP coded test input speech files...
are detected as the MELP coded speech.

<table>
<thead>
<tr>
<th>Actual Coder Type</th>
<th>Detected Coder Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>MELP</td>
<td>G723.1</td>
</tr>
<tr>
<td>94.2</td>
<td>0.0</td>
</tr>
<tr>
<td>83.5</td>
<td>0.0</td>
</tr>
<tr>
<td>84.8</td>
<td>0.0</td>
</tr>
<tr>
<td>92.9</td>
<td>1.3</td>
</tr>
</tbody>
</table>

Table 4.10: MELP coded male test speech SCD performance (in %) in four coder environment using MELP, G723.1, G729, and AMR coders with different filtering applied to coded test speech.

Table 4.11 shows the SCD system performance using MELP coded female speech. When there is no filtering applied, the correct coder type detection rate for MELP coded test speech is 100%. The SCD system performance using band-pass filtered MELP coded test speech is dropped to 69.8%. The SCD system detected 40.2% of the band-pass filtered MELP coded speech as G729 coded speech. Also 92.8% of the high-pass filtered MELP coded test speech files are detected correctly. Finally, 93.8% of the low-pass filtered MELP coded test speech are detected as the MELP coded speech.

Similar to two- and three-coder environments, applying different filters to the coded test input speech causes speech coder detection to misidentify the correct coder type.

Summary of Observations

It has been previously believed that the SCD uses algorithmic differences of each coder to determine the coder type. However, the results given above show that the bandwidth differences on the speech signals are used by the SCD to detect the coder type. These bandwidth differences in each coder are introduced by the high-pass filters used in coder/decoder parts. It has been observed that this information is very
### Table 4.11: MELP coded female test speech SCD performance (in %) in four coder environment using MELP, G723.1, G729, and AMR coders with different filtering applied to coded test speech

<table>
<thead>
<tr>
<th>Actual Coder Type</th>
<th>Detected Coder Type</th>
<th>MELP</th>
<th>G723.1</th>
<th>G729</th>
<th>AMR</th>
<th>Filtering</th>
</tr>
</thead>
<tbody>
<tr>
<td>MELP</td>
<td></td>
<td>100.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>No Filtering</td>
</tr>
<tr>
<td></td>
<td></td>
<td>59.8</td>
<td>0.0</td>
<td>40.2</td>
<td>0.0</td>
<td>BPF</td>
</tr>
<tr>
<td></td>
<td></td>
<td>92.8</td>
<td>0.9</td>
<td>6.3</td>
<td>0.0</td>
<td>HPF</td>
</tr>
<tr>
<td></td>
<td></td>
<td>93.8</td>
<td>0.9</td>
<td>5.3</td>
<td>0.0</td>
<td>LPF</td>
</tr>
</tbody>
</table>

Table 4.11: MELP coded female test speech SCD performance (in %) in four coder environment using MELP, G723.1, G729, and AMR coders with different filtering applied to coded test speech

useful for speaker recognition tasks, where there is a coder mismatch possibility. The speaker recognition performance can be improved by eliminating/minimising the coder mismatch problem using the SCD system combined with speaker recognition system. When the combined SCD/verification system is used in a multi-coder environment, the speaker verification performance is very similar to the performance of the matched case verification, providing the EER values for both the combined SCD/verification system experiment and matched case verification experiment as 3.2% for the male and 3.3% for the female speakers.

### 4.10 Conclusion

In this chapter, effects of the data mismatch on speaker recognition performance have been described. Four different speech coders namely the AMR (12.2 kb/s), the G729 (8 kb/s), the G723.1 (5.3 kb/s) and the MELP (2.4 kb/s) have been briefly described. It has been observed that speaker verification performance decreases relative to the speech coder bit rate. The coder mismatch problem in the multi-coder environment has been explained. Speech coder detection system that distinguishes coder types used for test speech has been presented. It has been demonstrated that the speech coder detection system embedded in the speaker verification system improves the speaker verification performance in a multi-coder environment. Further analysis has been performed on
speech coder detection system to fully understand the coder detection process. It has been observed that the speech coder detection system uses not only each coder’s particular algorithmic differences but also the bandwidth characteristics of the speech signal to make distinction between the coders. Speaker recognition performance can be further improved by using this information to minimise/eliminate the differences in the speaker models and test vectors that are coded by different speech coders.
Chapter 5

Application of Improved LSF Extraction Through Anti-Aliasing Filtering

5.1 Introduction

Applications such as transaction authentication may require speaker recognition systems to operate on compressed speech transmitted over mobile phone networks. However, speech compression degrades speech quality, and hence causes a reduction in speaker recognition performance [94, 95, 116]. It has been shown that the classic technique for extraction of LSF parameters in speech coders is prone to aliasing distortion [117]. The use of a low-pass filtering on up-sampled LSF vectors has been shown to alleviate this problem, therefore improving speech quality. In this chapter, the effect of this Non-Aliased LSF (NA-LSF) extraction method on speaker recognition performance is observed using GSM-EFR coded speech.
5.2 LSF Smoothing Through Anti-Aliasing Filtering

Speaker recognition systems use coded speech in applications such as telephone banking. However, speech coders perform compression of speech for transmission purposes. Speech model parameters are estimated at a given transmission rate over an analysis window with the assumption that speech is stationary within that analysis window. Such an assumption is used to validate the use of long-term concepts for short-term signal segments (i.e. Fourier transform [27]). As a natural signal, speech is generally known to be non-stationary. However, it can be treated as locally stationary signal for an analysis window [118]. For example LSF parameters are typically extracted at every 10 ms, which corresponds to 100 Hz. This assumes that the spectral envelope of the speech varies no faster than 50 Hz. Hence any variation faster than 50 Hz will cause spectral overlapping and aliasing distortion. It has been shown by [117] that the stationary assumption of speech in an analysis window is not an entirely true assumption.

To establish this, in our experiments we extracted LSF parameters at 8 kHz instead of 100 Hz i.e. LSF extraction is carried out by shifting the analysis window at every sample. Figure 5.1 shows the LSF extraction process at every sample.

![Figure 5.1: Over-sampled LSF parameter extraction](image)

The bold straight line curves shown in Figure 5.1 represent the classic parameter extraction, where LSF parameters are extracted at every 10ms. The thin dashed line

---

*Figure 5.1: Over-sampled LSF parameter extraction*
curves shown in Figure 5.1 represent the over-sampled parameter extraction, where LSF parameters are extracted by shifting the analysis window at every sample. When oversampled, it has been observed that the LSF parameters contain high frequency variations, which cause some aliasing noise in the LSF parameters [117]. In Figures 5.2 and 5.3, the oversampled LSF energy tracks are shown.

Figure 5.2: Spectra of LSF tracks extracted at every sample

Figure 5.3: Concentrating on region of interest (i.e. vector transmission rate at 50 Hz) of the spectra of LSF tracks extracted at every sample
It can be observed from these figures that there are frequency components above the system vector transmission rate i.e. 50 Hz. These unwanted frequency components can be removed by employing an anti-aliasing filter. The filter’s cut-off frequency is relative to the vector transmission rate of the coder. Since this method removes unwanted aliasing effects, it can be used for speech coders that employ LSF parameter calculation in their algorithm.

The anti-aliasing filtering method that is introduced in this chapter, operates within the speech coder without modifying its design. The experiments are performed using GSM-EFR [112] as it is the most widely used standard coder for mobile communications in Europe. Nevertheless, the results are also expected to be applicable to AMR [107], as its principles are similar to GSM-EFR.

The use of LSF parameters in the GSM-EFR coder with removed spectral aliasing effect, is expected to improve the speaker recognition performance.

### 5.3 Anti-Aliasing Filtering Process

#### 5.3.1 Sampling Theory

Sampling is a process of transforming a continuous-time signal into a discrete-time signal. The discrete-time signal values are attained from continuous-time signal at certain intervals according to the sampling theory. The sampling theory [119] states that the band-limited signal can be reconstructed from its samples without aliasing if the sampling frequency $F_S$ is equal to or greater than twice the maximum frequency component $f_{max}$ of the signal i.e.

$$F_S \geq 2f_{max}$$

#### 5.3.2 Sampling Rate Conversion

The sampling rate of a signal can be changed from one value to another. This process is known as sampling rate conversion. The new sampling rate is represented as:
where \( D > 0 \). The sampling rate conversion process is called interpolation (also known as up-sampling) if \( D \) is greater than one. If \( D \) is less than one, then the process is called decimation (also known as down-sampling). Interpolation can be performed by adding \( D \) zeros between original samples. This operation increases the sample number and the sampling rate of the signal by a factor of \( D \). The low-pass filtering operation after interpolation is essential in order to select the spectrum that corresponds to the up-sampled signal. Decimation can be performed by choosing every \( \frac{1}{D} \)th sample of the signal. This operation decreases the sample number and the sampling rate of the signal by a factor of \( D \). Before the decimation process, the signal must be low-pass filtered with a filter that has a cut-off frequency relative to the new sampling rate. This will eliminate the possibility of spectral overlapping and consequently aliasing.

5.3.3 LSF Extraction From a Decimation Perspective

In this section, the effects attributed to the lack of low-pass filtering during LSF parameter extraction in GSM-EFR speech coder are investigated. The GSM-EFR coder performs LP analysis [9] twice for each speech frame using autocorrelation, utilising two asymmetric 30 ms wide analysis windows that are different in shape. These windows are designed in such a way that look-ahead delay is not required. The first window is constructed from the two halves of Hamming windows that have different sizes. The first window's weight is concentrated at the second sub-frame of the coder analysis window, bearing in mind that the GSM-EFR algorithm divides each analysis window into 4 sub-frames. The second window is constructed from a Hamming window and a quarter of a cosine function cycle. The weight of this window is concentrated at the fourth sub-frame. The two analysis windows used in the GSM-EFR coder are shown in Figure 5.4 [112].

The frequency responses of the GSM-EFR coder analysis windows are shown in Figure 5.5. It can be observed in Figure 5.5 that the second LP analysis window has larger
Figure 5.4: Time plots of the GSM-EFR coder LP analysis windows

main lobe compared to the first window. The large main lobe of the second window produces high frequency leakage, consequently producing more noisy speech.

Each one of these windows is then used to produce 10 LSF vectors per speech frame. In order to analyse the effects of aliasing on LSF parameters, the LSF extraction is performed at a higher sampling rate (i.e. parameter extraction at every sample) than the system rate. LSF tracks show the LSF parameter evolution over time and they are obtained by plotting each parameter value in time using over-sampled LSF vectors. When down-sampling is performed on the LSF tracks at the system rate (i.e. the rate of vector transmission), the LSF vectors which are generated are identical to the original LSF extraction method. During the down-sampling process, any LSF track that contains spectral components at frequencies greater than half of its vector transmission frequency causes spectral overlapping. This produces some aliasing noise in the extracted LSF parameters. In order to remove the high frequency variations observed
5.3. Anti-Aliasing Filtering Process

in the LSF track spectra, a pre-processing stage was proposed in [117] that involves the use of low-pass filtering before LSF vector decimation. We employed this anti-aliasing filtering approach in the LSF parameter extraction section of the GSM-EFR. It will be shown that the use of NA-LSF parameters improve the quality of the synthesized speech produced by the GSM-EFR. Therefore, GSM-EFR speech coding with NA-LSF parameter extraction has been shown to provide more efficient speaker modeling and testing processes and ultimately better speaker recognition performance.

5.3.4 Low-Pass Filtering

In our experiments, the low-pass filtering has been applied as follows:

1. Extract two sets of LSF vectors \( f^1(n) = f_1^1(n), \ldots, f_p^1(n) \) and \( f^2(n) = f_1^2(n), \ldots, f_p^2(n) \) from the two sets of LPC vectors \( l^1(n) = l_1^1(n), \ldots, l_p^1(n) \) and \( l^2(n) = \ldots \).
Chapter 5. Application of Improved LSF Extraction Through Anti-Aliasing Fil.

1. Compute \( l_p^1(n), \ldots, l_p^2(n) \) at every sample for each analysis window of the GSM-EFR, where \( p \) is the LP filter order and \( n \) is time.

2. Construct two sets of LSF tracks using LSF vectors \( (l_p^1, l_p^2) \) for each analysis window obtained from the first step.

3. For each LSF track \( (l_p) \), perform low-pass filtering in the frequency domain with a cut-off frequency that is chosen according to the vector transmission rate.

In order to demonstrate the effect of low-pass filtering on the LSF tracks, sets of LSF tracks were obtained using speech samples collected from the 8 kHz down-sampled version of TIMIT database (TIMIT8k) [15].

Figure 5.6 (a), Figure 5.7 (a), Figure 5.8 (a), and Figure 5.9 (a) show the 1\(^{st}\), 4\(^{th}\), 7\(^{th}\) and 10\(^{th}\) LSF tracks of the original and the NA-LSF parameters obtained from the first LP analysis window of the GSM-EFR coder. Figure 5.6 (b), Figure 5.7 (b), Figure 5.8 (b), and Figure 5.9 (b) show the 1\(^{st}\), 4\(^{th}\), 7\(^{th}\) and 10\(^{th}\) LSF tracks of the original and the NA-LSF parameters obtained from the second LP analysis window of the GSM-EFR coder.

It can be observed from these figures that the NA-LSF tracks follow a smoother behavior compared to the original tracks. The original LSF tracks contain a large amount of variation. This variation in the original LSF tracks are more prominent in intervals where there is a transition between voiced and unvoiced speech. The distortions on the LSF tracks are more evident with the higher order LSF parameters (i.e. 10\(^{th}\) LSF parameter). Also it can be observed in these figures that the amount of distortion on the original LSF tracks for the second LP analysis window is much higher compared to the the first LP analysis window. This is caused by the use of two different window compositions as described in Section 5.3.3. The LSF parameters of the second LP analysis window of the GSM-EFR are more distorted as a result of the weighting applied by this unusually-shaped asymmetric window.
Figure 5.6: Variations in the 1st LSF track for original $f_1$ and low-pass filtered $g_1$ LSFs
Figure 5.7: Variations in the 4th LSF track for original $f_4$ and low-pass filtered $g_4$ LSFs
5.3. Anti-Aliasing Filtering Process

(a) LSF tracks $f_7$ and $g_7$ for the first analysis window

(b) LSF tracks $f_7$ and $g_7$ for the second analysis window

Figure 5.8: Variations in the 7th LSF track for original $f_7$ and low-pass filtered $g_7$ LSFs
Chapter 5. Application of Improved LSF Extraction Through Anti-Aliasing Fil.

Figure 5.9: Variations in the 10th LSF track for original $f_{10}$ and low-pass filtered $g_{10}$ LSFs

(a) LSF tracks $f_{10}$ and $g_{10}$ for the first analysis window

(b) LSF tracks $f_{10}^2$ and $g_{10}^2$ for the second analysis window
5.4 Speaker Verification Using NA-LSFs in GSM-EFR

The speaker verification system performance using coded speech was assessed in Chapter 4. It was shown that the speaker verification performance degrades when coded speech is used for speaker training and testing processes. As a result of speech coding effects, the verification performance decreases under matched training and testing conditions (i.e. the training and the testing data are collected from the same coder) [95]. Under mismatched conditions (i.e. the training data are collected from the clean speech and the testing data are collected from the coded speech), the verification performance degradation was found to be even higher due to the mismatch between the training models and the test vectors [95, 114]. Different methods such as Score Normalisation [95] and Speech Coder Recognition [114] have been used to reduce the loss in speaker recognition performance using coded speech under matched and mismatched training and testing conditions.

In this work, matched and mismatched conditions are used to demonstrate the benefit of the NA-LSF parameter extraction method. The following sections describe the experimental setup of the NA-LSF extraction process for the GSM-EFR coder and the performance evaluation of the speaker verification system using the GSM-EFR coded TIMIT8k database (TGSM database) with NA-LSFs employed in the coder.

The NA-LSF extraction is performed as described in Section 5.3.4. The FFT window size is chosen to be large enough in order to avoid the effects of the large side lobes of the rectangular window. The cut-off frequency of the low-pass filter used for the two LP windows is at 25 Hz as this corresponds to a 10 ms vector transmission rate of the GSM-EFR coder (keeping in mind that there are two LP analysis windows shifted by 20 ms every frame).

5.4.1 Speaker Verification Experiments

Speech databases NA-LSF TGSM and Org-LSF TGSM represent the GSM-EFR coded TIMIT database using the NA-LSF and original LSF methods, respectively. MFCCs were used as feature vectors for model training and speaker testing, which are de-
scribed in Chapter 2. 16 MFCCs were extracted at every 10 ms using 20 ms speech frame length. A Gaussian Mixture Model - Universal Background Model (GMM-UBM) speaker verification system [82] is used to perform experiments for male and female speakers separately. The gender-dependent background models were created using the concatenated speech of 120 male and 120 female speakers separately. Each UBM was constructed using 1024 mixtures. The claimant speaker models were derived from the gender-dependent UBMs using Bayesian adaptation. The number of claimant male and female speakers were 112 and 56, respectively. The verification score of a claimed speaker was determined by the log-likelihood ratio calculation. The results are reported as EER values. Table 5.1 and Table 5.2 show the EER values of the speaker verification system using different combinations of the training and the testing data for male and female speakers, respectively.

<table>
<thead>
<tr>
<th>Row</th>
<th>Training Speech</th>
<th>Testing Speech</th>
<th>EER (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Uncoded</td>
<td>Uncoded</td>
<td>1.34</td>
</tr>
<tr>
<td>B</td>
<td>Uncoded</td>
<td>Org-LSF TGSM</td>
<td>3.59</td>
</tr>
<tr>
<td>C</td>
<td>Uncoded</td>
<td>NA-LSF TGSM</td>
<td>3.14</td>
</tr>
<tr>
<td>D</td>
<td>Org-LSF TGSM</td>
<td>Org-LSF TGSM</td>
<td>2.23</td>
</tr>
<tr>
<td>E</td>
<td>NA-LSF TGSM</td>
<td>NA-LSF TGSM</td>
<td>1.75</td>
</tr>
<tr>
<td>F</td>
<td>Org-LSF TGSM</td>
<td>NA-LSF TGSM</td>
<td>1.86</td>
</tr>
<tr>
<td>G</td>
<td>NA-LSF TGSM</td>
<td>Org-LSF TGSM</td>
<td>1.77</td>
</tr>
</tbody>
</table>

Table 5.1: EER values of verification system for male speech using TIMIT8k, Original-LSF TGSM, and NA-LSF TGSM databases.

Tables 5.1 and 5.2 show that the use of NA-LSFs in the GSM-EFR coder reduces the amount of loss in speaker verification performance.

By employing NA-LSF extraction instead of the classical LSF extraction method in the GSM-EFR coder, the speaker verification EER values reduce from 3.59% to 3.14% for male speakers and 6.25% to 5.39% for female speakers in the mismatched training and testing conditions. Also when the NA-LSFs are used in the GSM-EFR coder,
5.4. Speaker Verification Using NA-LSFs in GSM-EFR

<table>
<thead>
<tr>
<th>Row</th>
<th>Training Speech</th>
<th>Testing Speech</th>
<th>EER (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Uncoded</td>
<td>Uncoded</td>
<td>1.79</td>
</tr>
<tr>
<td>B</td>
<td>Uncoded</td>
<td>Org-LSF TGSM</td>
<td>6.25</td>
</tr>
<tr>
<td>C</td>
<td>Uncoded</td>
<td>NA-LSF TGSM</td>
<td>5.39</td>
</tr>
<tr>
<td>D</td>
<td>Org-LSF TGSM</td>
<td>Org-LSF TGSM</td>
<td>2.52</td>
</tr>
<tr>
<td>E</td>
<td>NA-LSF TGSM</td>
<td>NA-LSF TGSM</td>
<td>1.83</td>
</tr>
<tr>
<td>F</td>
<td>Org-LSF TGSM</td>
<td>NA-LSF TGSM</td>
<td>1.92</td>
</tr>
<tr>
<td>G</td>
<td>NA-LSF TGSM</td>
<td>Org-LSF TGSM</td>
<td>1.79</td>
</tr>
</tbody>
</table>

Table 5.2: EER values of verification system for female speech using TIMIT8k, Original-LSF TGSM, and NA-LSF TGSM databases.

the speaker verification EER values reduce from 2.23% to 1.75% for male speakers and 2.52% to 1.83% for female speakers in the match training and testing conditions. Experimental results also show that using NA-LSF coded speech in only training or the testing process (e.g., the training speech is collected from the original-LSF GSM-EFR coder and the testing speech is collected from the NA-LSF GSM-EFR coder) improves the speaker verification performance. The performance increase is the result of using the NA-LSF parameter extraction method which removes the unwanted LSF track components in the frequency domain. More stable coefficients are obtained using the low-pass filtering operation, producing higher quality synthesized speech compared to the original LSF extraction of the GSM-EFR coder. As a result, the speaker verification performance on average is improved by 12.5% and 21.5% for male, and 13.8% and 27.4% for female speakers, under mismatched and matched conditions respectively. For female speakers, the EER value given in the row G of Table 5.2 is better than the EER value given in the row E. It is not clear why this particular EER value is smaller than the result of NA-LSF training/testing speech experiment. Although this method requires extra computational cost and time delay, as shown in [117] the method improves the synthesized speech quality, while providing easier quantisation compared to the original LSF extraction methods. Initial experiments indicate that the NA-LSF parameter extraction method can be used with any speech coder that employs LP analysis in its
5.5 Conclusion

In this chapter, the NA-LSF parameter extraction process for speaker recognition applications has been presented. It was shown that LSF vectors obtained with classical extraction methods contain undesired frequency components. These components cause some aliasing noise in the LSF parameters. The NA-LSF parameter extraction approach has been introduced in order to remove the undesired frequency components on the LSF tracks of the GSM-EFR coder. The speaker verification system experiments were performed using GSM-EFR coded speech in mismatched and matched conditions. The results obtained from these experiments show that the use of NA-LSF parameter extraction in the GSM-EFR coder increases speaker verification performance and by employing the NA-LSF method in the GSM-EFR coder reduces the speaker verification error by 12.5% and 21.5% for male, and 13.8% and 27.4% for female speakers under mismatched and matched conditions respectively. The NA-LSF parameter extraction process may be employed in the existing standard speech coders to improve the synthesised speech quality as well as the speaker recognition performance. Since NA-LSF parameters also provide performance improvement when used only on the training data or testing data, it is possible to use NA-LSF parameter extraction at the training process only. During the enrollment process, NA-LSFs can be used, and the users still would be able to use their existing handsets at the testing stage. The proposed method is fully compatible with the existing standard speech coders, and thus it does not require any modification to existing infrastructures.
Chapter 6

Noise Cancellation for Speaker Verification

6.1 Introduction

Background noise is one of the most important problems of communication systems, which causes disruption in the speech signal characteristics and affect the proper operation of the subsystems such as speech coders, and speaker recognition systems. In real-life scenarios, there are different background noise types, such as vehicular noise, and babble noise, which have different characteristics. Unless removed, background noise might severely reduce the performance of the targeted application such as quality of the coded speech, or the performance of the speaker recognition systems. It is desired to have a noise cancellation technique, which can operate under different noise conditions. In this chapter, we have been concentrated on using a particular noise suppression technique called the Minimum Mean Square Error - Log Spectral Amplitudes (MMSE-LSA) noise cancellation algorithm. The effect of background noise on speaker verification performance using clean and coded speech is demonstrated. The MMSE-LSA system is used as a pre- and post-processing technique, to remove the background noise. The chapter is finalised by showing the effects of different noise types and SNR levels on speaker recognition performance, followed by demonstrating the effectiveness of the MMSE-LSA algorithm under various combinations of matched and mismatched
training and testing conditions.

6.2 Background Noise and MMSE-LSA Noise Suppression

The presence of background noise in speech, where it is used for applications such as speech coding or speaker recognition, may cause unwanted effects, influencing the system outputs such as poor quality coded speech or lower recognition performance. In the case of speaker recognition applications, it is not only the presence of background noise that degrades the speaker recognition performance, but also the mismatch between the training and testing data caused by unknown background noise characteristics. Another aspect, which is worth investigating, is the effect of the background noise on speaker recognition performance when the speech is coded for mobile voice communication systems, and the usability of the MMSE-LSA noise canceller for this particular scenario.

Most of the speech enhancement methods operate in the frequency domain. The noise suppression operation is performed by dividing the noise corrupted signal into frequency bins i.e. different spectral components. In order to suppress the noise in each bin, a frequency-dependent gain function is used. This gain function applies different amounts of attenuation to different frequency bins, imposing different amount of noise suppression depending on the amount of noise present in each bin.

There are two different approaches used for gain function modification, known as hard-decision and soft-decision. In the hard-decision modification approach, there is only one speech/non-speech decision is made for the whole speech segment. The noise characteristics are adapted only when the speech segment is voice-inactive. The voicing decision is made by a VAD, therefore if the VAD process fails to perform correct speech detection due to heavy noise conditions, the hard-decision approach might suffer low performance. Another approach called soft-decision modification has been introduced to improve the noise suppression by modifying the frequency-dependent gain function depending on the priori probability of the speech absence in each frequency bin [120].

Accurate background noise power spectrum estimation plays a very important role on
6.2. Background Noise and MMSE-LSA Noise Suppression

The effectiveness of noise suppression methods. The noise spectrum estimate is updated during the non-speech intervals of the noisy speech using a VAD. However, considering that under heavy noise conditions VAD might fail to correctly determine the speech segments, it is crucial to continue updating the noise power spectrum estimation and noise adaptation process during the speech intervals, to fully capture the changes in noise spectrum.

The most of the popular noise-cancellation techniques such as Wiener filtering [47, 121] and spectral subtraction, which are designed to deal with uncorrelated additive noise, use Short-Time Spectral Amplitude (STSA) of the speech signal for noise suppression. Therefore, the noise cancellation technique that is used in our work is based on MMSE-LSA, which is an STSA based technique also known as Ephraim-Malah filter [48, 49, 122]. The noise-canceller used in this work is an adapted version of the improved MMSE-LSA technique developed at the Centre for Communication Systems Research (CCSR).

The MMSE-LSA noise-canceller (will be used interchangeably with Ephraim-Malah filter) is an STSA estimator based method, which performs modification on the noisy speech spectral amplitude without changing its phase. The phase enhancement is a complex process with a small gain in performance, hence the enhanced signal is reconstructed by using phase information of the noisy speech with the modified spectral amplitudes.

The amplitude estimator of the model can be summarised as follows:

Assuming that the noisy speech signal $y(t)$ is composed of speech signal $x(t)$ and additive noise $d(t)$. The noisy speech signal can be written as follows:

$$y(t) = x(t) + d(t), \quad 0 \leq t \leq T. \quad (6.1)$$

where $t$ represents the time. The noise-canceller is designed to produce enhanced signal $\hat{x}(t)$ for a given noisy signal $y(t)$, assuming that the noise $d(t)$ is uncorrelated with speech signal $x(t)$. The frequency domain representation of the Equation (6.1) can be written as follows:
\[ Y_k = X_k + D_k \]  
(6.2)

where \( k \) represents the frequency bin index, \( Y_k = R_k \exp(j \vartheta_k) \), \( X_k = A_k \exp(j \alpha_k) \), and \( D_k \) represent the \( k^{th} \) spectral component of the signal \( y(t) \), \( x(t) \), and \( d(t) \) in an interval \([0, T]\), respectively.

It is assumed that the speech and noise DFT coefficients are independent Gaussian random variables [122].

The MMSE-LSA estimator of the logarithm of the STSA \( A_k \) of the clean speech is used to derive the estimate of the clean speech magnitude as follows [122]:

\[ \hat{A}_k = E\{A_k|Y_k, H_k^k\} P(H_k^k|Y_k) \]  
(6.3)

where \( H_k^k \) represents the speech presence in the \( k^{th} \) bin, \( E\{A_k|Y_k, H_k^k\} \) represents the expected speech spectrum under speech presence, and \( P(H_k^k|Y_k) \) represents the soft-decision modification under the speech signal presence hypothesis [122].

The enhanced speech amplitude estimator is written in terms of gains as follows:

\[ \hat{A}_k = G_m(k) G_{lsa}(k) R_k \]  
(6.4)

where \( G_m(k) \) represents the soft-decision gain modification, \( G_{lsa} \) represents the log-spectral amplitude gain function, and \( R_k \) represents the noisy input speech.

### 6.2.1 Soft-Decision Gain

The soft-decision gain \( G_m(k) \) is obtained by using Bayes' rule to Equation (6.3) [122]:

\[ G_m(k) = P(H_k^k|Y_k) = \frac{P(Y_k|H_k^k)P(H_k^k)}{P(Y_k|H_0^k)P(H_0^k) + P(Y_k|H_1^k)P(H_1^k)} \]  
(6.5)
which can be written as follows:

\[ G_m(k) = P(H_k^1 | Y_k) = \frac{\Lambda(k)}{1 + \Lambda(k)} \]  

(6.6)

where \( \Lambda(k) \) represents the likelihood ratio:

\[ \Lambda(k) = \frac{P(Y_k | H_k^1)}{P(Y_k | H_k^0)} \]  

(6.7)

where \( \mu_k \) is written as:

\[ \mu_k = \frac{P(H_k^1)}{P(H_k^0)} = \frac{1 - q_k}{q_k} \]  

(6.8)

where \( q_k \) represents the a priori probability of speech absence in the \( k^{th} \) bin.

### 6.2.2 Log-Spectral Amplitude Gain

The second component of the Equation (6.4), which is Log-Spectral Amplitude (LSA) gain is computed as follows:

\[ G_{lsa}(\xi_k, \gamma_k) = \frac{\xi_k}{1 + \xi_k} \exp\left( \frac{1}{2} \int_{-\infty}^{\infty} e^{-t} dt \right) \]  

(6.9)

where \( \gamma_k \) represents the a posteriori SNR, \( \eta_k \) represents the a priori SNR, and \( q_k \) represents the prior probability of speech absence, for the \( k^{th} \) frequency bin.

The a posteriori SNR \( \gamma_k \) and the a priori SNR \( \eta_k \) can be computed as follows:

\[ \gamma_k = \frac{R_k^2}{\lambda_d(k)} \]  

(6.10)

and

\[ \eta_k = \frac{\lambda_x(k)}{\lambda_d(k)} \]  

(6.11)
where

\[ u_k = \frac{\xi_k}{1 + \xi_k} \eta_k \]  
(6.12)

\[ \xi_k = \frac{\eta_k}{1 - q_k} \]  
(6.13)

\[ \lambda_d(k) = E \left\{ |D_k|^2 \right\} \]  
(6.14)

\[ \lambda_x(k) = E \left\{ |X_k|^2 \right\} \]  
(6.15)

where \( \lambda_d \) represents the estimated noise power spectrum, and \( \lambda_x \) represents the speech power spectrum.

By using the equations above, Equation (6.7) can be rewritten as follows:

\[ \Lambda(k) = \mu_k \frac{e^{\alpha_k}}{1 + \xi_k} \]  
(6.16)

The accuracy of the MMSE-LSA noise-canceller is dependent on the input signal amplitude observations, prior probability of speech absence estimation, and the noise power spectrum adaptation.

### 6.2.3 A Priori Probability Estimation

A priori probability estimation is very important process for the noise-canceller, since it is used for both the gain and noise power spectrum adaptation computations. One can use fixed probability of speech absence in all bins. But when voiced speech is considered (i.e. quasi-harmonic signal), where the speech energy may not be present in every bin, this assumption is not valid. By allowing different probability value in each bin, the noise power spectrum estimation can be updated when speech is present. \( q_k \) is computed as follows:
6.2 Background Noise and MMSE-LSA Noise Suppression

\[ q_k(l) = \alpha_q q_k(l-1) + (1 - \alpha_q) I_k(l) \]  
(6.17)

where \( \alpha_q \) is a tuning factor, \( I_k(l) \) represents a VAD test index function, and \( l \) is the frame notation.

\( I_k(l) \) provides a VAD test results of each bin as follows:

\[ I_k = \begin{cases} 0 & \gamma_k \geq \gamma_{th} \\ 1 & \gamma_k < \gamma_{th} \end{cases} \]  
(6.18)

where \( \gamma_{th} \) is a pre-defined threshold value.

6.2.4 Noise Spectrum Adaptation

Another important process for noise cancellation process is the estimation of a noise power spectrum, since it controls the level of signal suppression in the noise-canceller. Noise spectrum is generally estimated at the intervals by smoothing of \( R_k^2 \), where the speech is absent, by using a VAD. The background noise power spectrum can be computed as follows:

\[ \lambda_d(k, l) = \alpha_d(l) \lambda_d(k, l-1) + (1 - \alpha_d(l)) R_k^2 \]  
(6.19)

with

\[ \alpha_d(l) = 1 - 0.2|\gamma(l-1) - 1|, \quad 0.8 \leq \alpha_d(l) \leq 0.98 \]  
(6.20)

where \( \gamma(l-1) \) represents the average a posteriori SNR computed up to the previous frame.

The process of the noise power spectrum adaptation, as shown in Equation (6.19), is performed recursively with a dynamic smoothing factor to account for the possible spectral changes in the non-speech intervals (considering the non-stationarity of the noise). In Equation (6.20), \( \gamma \) indicates if the noise spectrum changes faster than the present noise power spectrum adaptation rate. One also needs to consider the possible
changes of the noise power spectrum during intervals of speech presence. Therefore $\alpha_d$ can be modified in order to update the noise power spectrum during intervals of speech presence in the bins where the current SNR value is very low as follows:

$$\alpha_d(t) = 1 - 0.2|\gamma(l - 1) - 1|q_h, \quad 0.8 \leq \alpha_d(t) \leq 0.98 \quad (6.21)$$

6.2.5 Voice Activity Detector (VAD)

The VAD operation is also very critical part of the noise-canceller, since it is actively used during the gain computations. The VAD used in MMSE-LSA noise-canceller system is produced from the work of [123].

6.3 Speaker Verification using Noise-Canceller

The performance of speaker recognition systems degrade in the presence of a background noise [53, 124, 125]. In this work, our aim is to use MMSE-LSA noise-cancellation technique in order to remove the unwanted effects of the background noise on the speech signal, consequently improving speaker recognition performance. The experimental results described in the following sections were obtained from a speaker verification system using speech collected from male and female speakers for different combinations of training and testing processes.

6.3.1 Speaker Training and Test Data

During speaker verification experiments, following versions of the TIMIT@k (8 kHz version of TIMIT database) speech database were used:

- clean
- noisy (15 dB SNR, noisy speech corrupted with vehicular noise)
- noise-cancelled (noise-cancelled noisy speech using Ephraim-Malah filter)
- coded (coded speech using AMR 12.2 kb/s speech coder)
6.3. Speaker Verification using Noise-Canceller

6.3.2 Experimental Setup

MFCCs were used as feature vectors for speaker model training and testing (described in Chapter 2). 16 MFCCs were extracted at every 10 ms using 20 ms speech analysis frame length. A Gaussian Mixture Model - Universal Background Model (GMM-UBM) speaker verification system [82] is used to perform experiments for male and female speakers separately. The gender-dependent background models were created using the concatenated speech of 120 male and 120 female speakers separately. Each UBM was constructed using 1024 mixtures. The claimant speaker models were derived from the gender-dependent UBMs using Bayesian adaptation. The number of claimant male and female speakers were 112 and 56, respectively. The verification score of a claimed speaker was determined by the log-likelihood ratio calculation. The results are reported as averaged EER values for male and female speakers. The separate male and female EER values are presented in Appendix A in order to reduce the number of tables to be shown in Chapter 6 for simplicity and clarity purposes.

6.3.3 Verification Experiments

The training and testing processes were performed in several different combinations to analyse the effect of background noise on speaker verification performance. First part of Table 6.1 shows the averaged EER values of the speaker verification system for male and female speech verification experiments.

The results shown in Table 6.1 shows that the verification performance degrades when noisy speech is used for both the training and the testing processes (at row A-B of Table 6.1). The verification system EER value increases from 1.57% to 3.39%. When the MMSE-LSA noise-canceller is used as a pre-processing stage prior to speaker verification task to remove the noise from the noisy training and testing data, the EER value becomes 3.57% (row C of Table 6.1). The performance difference between the row B and row C of Table 6.1 might be caused by two possible reasons. It is possible that the noise-canceller removes some of the speech information, while removing the background
### Table 6.1: Averaged male and female EER values of verification system using TIMIT8k, Noisy TIMIT8k corrupted with vehicle noise (15 dB SNR), and noise-cancelled speech using MMSE-LSA noise canceller.

<table>
<thead>
<tr>
<th>Row</th>
<th>Training Speech</th>
<th>Testing Speech</th>
<th>EER (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Clean</td>
<td>Clean</td>
<td>1.57</td>
</tr>
<tr>
<td>B</td>
<td>Noisy</td>
<td>Noisy</td>
<td>3.39</td>
</tr>
<tr>
<td>C</td>
<td>Noise-Cancelled</td>
<td>Noise-Cancelled</td>
<td>3.57</td>
</tr>
<tr>
<td>D</td>
<td>Clean</td>
<td>Noisy</td>
<td>8.30</td>
</tr>
<tr>
<td>E</td>
<td>Clean</td>
<td>Noise-Cancelled</td>
<td>6.18</td>
</tr>
<tr>
<td>F</td>
<td>Clean-Coded</td>
<td>Clean-Coded</td>
<td>2.38</td>
</tr>
<tr>
<td>G</td>
<td>Noisy-Coded</td>
<td>Noisy-Coded</td>
<td>3.82</td>
</tr>
<tr>
<td>H</td>
<td>Noise-Cancelled-Coded</td>
<td>Noise-Cancelled-Coded</td>
<td>3.80</td>
</tr>
<tr>
<td>I</td>
<td>Clean-Coded</td>
<td>Noisy-Coded</td>
<td>7.69</td>
</tr>
<tr>
<td>J</td>
<td>Clean-Coded</td>
<td>Noise-Cancelled-Coded</td>
<td>6.38</td>
</tr>
<tr>
<td>K</td>
<td>Noisy-Coded-Noisy_Cancel_</td>
<td>Noisy-Coded-Noisy_Cancel_</td>
<td>3.87</td>
</tr>
<tr>
<td>L</td>
<td>Clean-Coded</td>
<td>Noisy-Coded-Noisy_Cancel_</td>
<td>8.04</td>
</tr>
</tbody>
</table>
6.3. Speaker Verification using Noise-Canceller

Noise, or the noise information is utilised by the speaker model, increasing the match between the training and testing data, which in effect might help to improve speaker recognition. However, when the speaker model training is done on a clean speech, assuming that there is no prior knowledge about the type of the background noise (to generate the speaker training models using that background noise), the noise-cancelled test speech data produces lower speaker verification EER values than the noisy speech files (shown in row D and E of Table 6.1). This result indicates that the noise-canceller in fact, removes the undesired effects of the background noise from the speech, allowing speaker recognition process to work more effectively.

6.3.4 Noise Cancellation and Speech Coding

Considering fast growth of interest in the mobile voice communication systems and their usage in applications that require user authentication, it is crucial to provide noise-robust speaker recognition in these systems to minimise possible frauds. Therefore, this section of Chapter 6 investigates the use of the MMSE-LSA noise canceller in a mobile voice communication system using GSM-AMR speech coder. The following experiments aim to demonstrate the effects of background noise, and the noise-cancellation process when the noise-corrupted speech is coded.

Noise-Canceller as a Pre-Processor

The verification experiments were performed for male and female speakers using the GSM-AMR coded versions of the clean, the noisy (corrupted with vehicle noise, 15 dB SNR), and the noise-cancelled TIMIT8k speech database. The averaged EER values are shown in second part of Table 6.1. As explained in Chapter 4, the speaker recognition performance degrades when the coded speech is used for speaker recognition tasks. This is shown in row F of Table 6.1, where speaker verification error rate increases from 1.57% to 2.38% when the AMR coded speech is used for speaker recognition, instead of clean speech. When the noisy speech is used for speech coding, the speaker verification performance degrades further, producing EER value of 3.82%, which is shown in row G of Table 6.1. This result
indicates that the GSM-AMR speech coder is affected by the presence of background noise, which is also reflected on the speaker verification performance producing higher error rates. It can be observed from the results shown in rows G and H of Table 6.1 that when the noise-canceller is used prior to the coding process for the matched training and the test data, the speaker verification EER values are not affected by the noise-cancellation process. The EER value of speaker verification task using noise-cancelled coded speech is 3.80%. Hence it can be concluded from these results that the noise-cancellation process is not effective in speaker recognition performance for matched training and testing conditions when the coded speech is used for speaker recognition. However, it can also be observed that when the AMR-coded clean speech is used for speaker training process, the use of noise-canceller on a noisy test speech, prior to speech coding at the testing process, produces lower EER value compared to the noisy AMR-coded test speech. Mismatched condition of clean coded training speech and the noise-cancelled AMR coded test speech verification produces EER value of 6.38%, whereas the noisy AMR coded test speech verification EER value is 7.69% (shown in row I and J of Table 6.1).

**Noise-Canceller as a Post-Processor**

Further experiments were carried out by applying the noise-cancellation process as a post-processing stage to the speech coding process instead of using it as a pre-processing stage. The third part of Table 6.1 shows the averaged male and female speaker EER values when the MMSE-LSA noise-canceller is used as a post-processor after the speech coding process. When the matched training and the test data are used for verification, using noise-canceller as a post-processor after the speech coding is shown not to have any major effect on recognition performance. This experiment produces EER value of 3.87% as shown in row K of Table 6.1, which is very similar to the performance of the recognition system when the noise-canceller is used as a pre-processing stage (e.g. 3.80% EER). On the other hand, when the mismatched training and the test data are used for verification, using noise-canceller on a coded noisy speech as a post-processor is shown to reduce the recognition performance. This is shown in row L of Table 6.1, with EER value of 8.04%, where clean coded speech is used for training, and noisy coded
6.3. Speaker Verification using Noise-Canceller

Speech followed by noise-cancellation process is used for testing tasks. This result is even lower than the result shown in row I, where noisy coded test speech is used for mismatched speaker verification task. This result indicates that the noise-cancellation process should be performed prior to speech coding, preventing speech quality degradation during speech coding, ultimately improving speaker recognition performance.

6.3.5 Noise Cancellation for Different Background Noise Characteristics

Considering the experimental results demonstrated in Table 6.1, the MMSE-LSA noise-canceller is shown to improve speaker recognition performance only when the training and testing data are mismatched. It appears that there is no improvement in recognition performance when the training and test data are matched. However, the matched case (i.e. training and test data are both collected from the environments, where the background noise types and SNRs for both data sets are same) considered in this scenario is not very realistic. In real-life scenarios, on most occasions it is not possible to collect training and test data with matching background noise types and SNR values. A typical example to such a scenario can be shown as the speaker accessing remote authentication services through his mobile phone while traveling on a train, whom enrolled using a land-line phone from the comfort of his quiet house environment. Therefore further experiments have been carried out to show the effectiveness of the MMSE-LSA noise-canceller on speaker recognition performance, where the mismatch of the training and the test data is introduced by the different background noise types and SNR values.

Noisy Speech Verification

Table 6.2 shows averaged male and female speaker EER values for speaker verification experiments, where the training data are noise corrupted speech (15 dB SNR, vehicular noise) and the test data are noise corrupted speech with different background noise
Chapter 6. Noise Cancellation for Speaker Verification

types and SNR values.

<table>
<thead>
<tr>
<th>Background Noise Type</th>
<th>Vehicle</th>
<th>Babble</th>
<th>Gaussian</th>
</tr>
</thead>
<tbody>
<tr>
<td>Row</td>
<td>SNR (dB)</td>
<td>EER (%)</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>5</td>
<td>24.20</td>
<td>24.56</td>
</tr>
<tr>
<td>B</td>
<td>15</td>
<td>3.39</td>
<td>12.56</td>
</tr>
<tr>
<td>C</td>
<td>25</td>
<td>5.39</td>
<td>9.75</td>
</tr>
</tbody>
</table>

Table 6.2: Averaged male and female speaker verification EER values for noisy training speech (15 dB SNR, vehicular noise) and noisy test speech with different noise types and SNRs

Table 6.2 aims to show that when the background noise conditions are not matched, the degradation in speaker recognition performance is in fact very significant. In this table, test speech materials are corrupted with three different background noise types, namely vehicular, babble, and Gaussian noises with varying SNR values of 5 dB, 15 dB, and 25 dB. The training data is corrupted with vehicular noise of 15 dB SNR value. As Table 6.2 shows, the speaker verification performance is strongly affected by the varying background noise types and SNR values. The only case where the speaker recognition performance is not degraded is when both the training and test data are collected from noisy speech with 15 dB SNR corrupted by vehicular noise. This is the matched case described in Table 6.1. It can be observed from Table 6.2 that lower the SNR value of the test speech, higher the degradation in speaker verification performance. It can also be observed that as the noise types differ, the performance degradation gets higher with decreasing SNR values. The only exception is the matched case of test data corrupted by vehicular noise (with 15 dB SNR) as shown in row B of Table 6.2. It can be concluded from the results presented in Table 6.2 that the background noise mismatch is a very important problem for speaker recognition systems and it is necessary to use a noise-canceller prior to speaker recognition to minimise speaker recognition performance degradation.
6.3. Speaker Verification using Noise-Canceller

Mismatched Conditions and Noise Cancellation

Since it is not realistic to expect to collect speech material with matched training and test data in terms of noise conditions due to the environmental changes, the concept of mismatched conditions in relation with noise cancellation task was further investigated by performing experiments with different combinations of mismatched training and test data with and without the noise-cancellation process. Table 6.3 shows averaged male and female speaker EER values for speaker verification experiments using noise-cancelled mismatched training and test data.

<table>
<thead>
<tr>
<th>Background Noise Type</th>
<th>Vehicle</th>
<th>Babble</th>
<th>Gaussian</th>
</tr>
</thead>
<tbody>
<tr>
<td>Row</td>
<td>SNR (dB)</td>
<td>EER (%)</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>5</td>
<td>19.08</td>
<td>24.55</td>
</tr>
<tr>
<td>B</td>
<td>15</td>
<td>3.57</td>
<td>6.90</td>
</tr>
<tr>
<td>C</td>
<td>25</td>
<td>6.44</td>
<td>6.10</td>
</tr>
</tbody>
</table>

Table 6.3: Averaged male and female speaker verification EER values for noise-cancelled training speech (15 dB SNR, vehicular noise) and noise-cancelled test speech with different noise types and SNRs.

The results presented in Table 6.3 show that using MMSE-LSA noise canceller on a noisy training and test data produces lower recognition error rates compared to the case where there is no noise-canceller used prior to the verification task. When compared with results shown in Table 6.2, the improvement in error rates is higher where SNR of the test speech is 15 dB. Noise-cancellation on noisy speech corrupted with babble noise is shown to produce higher improvement in recognition performance degradation, followed by Gaussian and vehicle noises. General trend of the EER values shown in Table 6.3 indicate that the noise-cancellation is in fact a very useful process prior to speaker recognition. The speaker verification performance on average is improved by 16.98%, when the noise-canceller is used for both training and test processes compared to noisy training and test processes. As these results indicate, the noise-suppressed
speech material can be better modelled, ultimately producing higher speaker recognition rates.

Table 6.4 depicts averaged male and female speaker EER values for speaker verification experiments using mismatched noise-cancelled training data and noisy test data.

<table>
<thead>
<tr>
<th>Background Noise Type</th>
<th>Vehicle (dB)</th>
<th>Babble</th>
<th>Gaussian</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Row</strong></td>
<td><strong>SNR (dB)</strong></td>
<td><strong>EER (%)</strong></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>5</td>
<td>30.13</td>
<td>26.43</td>
</tr>
<tr>
<td>B</td>
<td>15</td>
<td>6.48</td>
<td>11.34</td>
</tr>
<tr>
<td>C</td>
<td>25</td>
<td>6.79</td>
<td>6.57</td>
</tr>
</tbody>
</table>

Table 6.4: Averaged male and female speaker verification EER values for noise-cancelled training speech (15 dB SNR, vehicular noise) and noisy test speech with different noise types and SNRs.

Table 6.4 demonstrates that when the noise-canceller is only used on the training data but not on the test data, as expected, the speaker recognition performance degradation is unavoidable. Similar error rates with noisy training and noisy test data (as depicted in Table 6.2) obtained in this table show that it is required to use the noise-canceller in both training and test processes.

It was shown in Table 6.1 that when the noise-canceller is used on the noisy test speech, where clean speech is used for the training process, speaker recognition rates improve. For the completeness of the results, speaker verification experiments were performed using clean speech as a training data, and noise-cancelled and noisy speech as two different sets of test data.

Table 6.5 and Table 6.6 show averaged male and female speaker EER values for speaker verification experiments using noise-cancelled and noisy test data respectively, where clean speech is used as training data for both tables.

It can be observed from Table 6.5 and Table 6.6 that using the noise-canceller on the test data improves verification performance when the clean speech is used for speaker model training. When compared with results shown in Table 6.6, the highest improvement
6.3. Speaker Verification using Noise-Canceller

Table 6.5: Averaged male and female speaker verification EER values for clean training speech and noise-cancelled test speech with different noise types and SNRs

<table>
<thead>
<tr>
<th>Row</th>
<th>SNR (dB)</th>
<th>Vehicle</th>
<th>Babble</th>
<th>Gaussian</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>5</td>
<td>23.66</td>
<td>22.54</td>
<td>35.41</td>
</tr>
<tr>
<td>B</td>
<td>15</td>
<td>6.18</td>
<td>5.12</td>
<td>10.97</td>
</tr>
<tr>
<td>C</td>
<td>25</td>
<td>2.73</td>
<td>1.88</td>
<td>4.69</td>
</tr>
</tbody>
</table>

Table 6.6: Averaged male and female speaker verification EER values for clean training speech and noisy test speech with different noise types and SNRs

<table>
<thead>
<tr>
<th>Row</th>
<th>SNR (dB)</th>
<th>Vehicle</th>
<th>Babble</th>
<th>Gaussian</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>5</td>
<td>29.65</td>
<td>24.38</td>
<td>39.68</td>
</tr>
<tr>
<td>B</td>
<td>15</td>
<td>8.30</td>
<td>7.38</td>
<td>18.26</td>
</tr>
<tr>
<td>C</td>
<td>25</td>
<td>3.34</td>
<td>3.05</td>
<td>5.86</td>
</tr>
</tbody>
</table>
Chapter 6. Noise Cancellation for Speaker Verification

in error rates shown in Table 6.5 is at row B, where SNR of the test speech is 15 dB, followed by row C with 25 dB and A with 5 dB. Noise-cancellation on noisy test speech corrupted with babble noise is again shown to produce higher improvement in recognition performance degradation, followed by Gaussian and vehicle noises. The speaker verification performance on average is improved by 23.46%, for the recognition system trained on clean speech and tested on noise-cancelled speech compared to noisy test speech.

6.4 Conclusion

The overall results can be summarised as follows:

Presence of the background noise degrades the speaker recognition performance for all noise types and SNR values.

When the background noise conditions are matched, the noise-canceller does not improve speaker recognition performance, which can be observed in the following results:

- The use of the MMSE-LSA noise-canceller, when used both for training and testing data that are corrupted with matching noise conditions, produces slightly higher error rate compared to the matched case noisy speaker training and testing data. We believe that the performance difference is introduced by the fact that the noise-canceller removes some speech as well as background noise during noise-cancellation process, and the background noise is utilised by the speaker model, which helps improving speaker recognition for matched background noise conditions only.

- When the noise-canceller is used prior to speech coding process on the training and the testing data with matched noise conditions, the noise-canceller produces error rate that is very similar to the noisy coded-speech training and testing verification experiments. This indicates that the use of noise-canceller prior to the speech coding, for the matched training and test data, is not beneficial for speaker recognition process.
When the background noise conditions of training and test data are mismatched in terms of noise type and SNR values, the degradation in speaker recognition performance increases further. The noise-canceller significantly improves speaker recognition performance when the background noise conditions are mismatched, which can be observed in the following results:

- When the noise-canceller is used at the testing process only, while using clean speech during training process (assuming that there is not any prior knowledge about the type of background noise), the noise-cancelled test speech verification experiments provide lower error rate than the noisy test speech.

- When the noise-canceller is used prior to speech coding process at the testing process only, while using AMR-coded clean speech during training process (assuming that there is not any prior knowledge about the type of background noise), the noise-cancelled AMR-coded test speech verification experiments provides lower error rate than the AMR-coded noisy test speech. This result indicates that the noise-canceller can be used to improve the speaker recognition performance for noisy coded speech, when the training and test conditions are mismatched.

- Using noise-canceller as a post-processor following the speech coding process, is not effective on speaker recognition performance for matched case. However, it reduces speaker verification performance when used for mismatched conditions. This shows that the speech coder is affected by the noisy input speech, consequently reducing speaker recognition performance. Therefore it can be concluded that the noise-canceller should be used prior to speech coding process for effective noise-cancellation for speech coding applications.

- The speaker verification performance improves (in our experiments by 16.98%) when the noise-canceller is used for both training and test processes compared to noisy training and test processes. Noise-canceller used on noisy speech corrupted with babble noise is shown to produce highest improvement in recognition performance degradation, followed by Gaussian and vehicle noises.

- The speaker verification performance improves (in our experiments by 23.46%)
when the noise-canceller is used for the test process only, for the recognition system trained on clean speech compared to clean training and noisy test processes. Noise-cancellation on noisy test speech corrupted with babble noise is shown to produce highest improvement in recognition performance degradation, followed by Gaussian and vehicle noises.

It can be concluded from the results summarised above that the noise cancellation is a very useful process to reduce the speaker recognition performance degradation caused by the mismatched background noise conditions. In real systems, the noise cancellation process should be performed prior to speech coding in order to avoid reducing the quality of the coded speech, and consequently improving speaker recognition performance.
Chapter 7

Conclusions

7.1 Concluding Overview

This chapter summarises the main research achievements presented in the previous chapters. The thesis has mainly focused on enhancement techniques for improving the speaker recognition performance. The work presented can be broadly divided into four main areas:

- **Baseline GMM Recognition:**
  Baseline speaker identification and verification systems using GMMs for classification have been analysed for clean and noisy telephone speech. Identification and verification system performances have been shown to be very high for clean speech and degraded considerably for noisy telephone speech. The main reason for speaker recognition performance degradation of the noisy telephone speech has been shown to be the training and the testing environment mismatches.

- **Speech Coder Detection:**
  As the interest in mobile voice communication systems grows, the demand for speaker recognition applications, which use coded speech, increases. However, in these applications the speaker recognition performance degrades due to the speech compression carried out by the speech coders. Recognition performance
loss increases when the quality of the speech coder decreases. Speaker recognition systems perform well when the training and the testing environmental conditions are identical. However, mismatched environmental conditions cause speaker recognition performance degradation due to the acoustical mismatches in the speaker models and the test speech feature vectors. In Chapter 4, effects of the data mismatch on speaker recognition performance introduced by the speech coding process have been described. It has been observed that the speaker verification performance decreases relative to the speech coder bit rate. The coder mismatch problem in the multi-coder environment has been introduced. The speech coder detection system to distinguish the coder type of the speech prior to testing process has been presented. It has been shown that the speech coder detection system embedded in the speaker verification system improves the speaker verification performance in a multi-coder environment, producing error rate very similar to the matched case. A thorough investigation has been performed on speech coder detection system to fully understand the coder detection process. Previously, it was believed that the speech coder detection system uses each coder's algorithmic differences to detect the coder type. In our experiments, it has been observed that the speech coder detection system mainly uses the bandwidth characteristics of the speech signal to make distinction between the coders.

- **Application of NA-LSF Parameter Extraction:**

  People have overlooked the fundamental signal processing details that the input signal is wide sense stationary. However, there is a small leakage, which may not be so significant in speech coding as there is already a lot of loss of information at low bit rates. However, in the case of speaker recognition even small variations in the signal quality may affect the performance of the recognition system. In Chapter 5, the NA-LSF parameter extraction process for speaker recognition applications has been presented. It has been previously shown that LSF vectors obtained with classical extraction methods contain undesired frequency components, which cause some aliasing noise in the LSF parameters. The NA-LSF parameter extraction approach has been introduced to remove the undesired
frequency components on the LSF tracks of the GSM-EFR coder. The speaker verification system performance has been analysed using GSM-EFR coded speech in mismatched and matched conditions. It has been shown that when the NA-LSF parameter extraction method is used in the GSM-EFR coder, the speaker verification performance degradation reduces for both mismatched and matched conditions (in our experiments by 12.5% and 21.5% for male, and 13.8% and 27.4% for female speakers under mismatched and matched conditions respectively). The proposed method is fully compatible with the existing standard speech coders, and thus it does not require any modification to existing infrastructures.

*Noise Cancellation for Speaker Verification:*

Speech corrupted with background noise, where it is used for applications such as speech coding or speaker recognition, may influence the quality of the system performance, consequently producing poor quality coded speech or, lower recognition performance. In speaker recognition, the presence of background noise as well as the mismatch between the training and testing data caused by unknown background noise characteristics degrade the speaker recognition performance. Chapter 6 has detailed an investigation into the effects of background noise on speaker recognition performance. It has been shown that the presence of the background noise reduces the speaker recognition performance considerably. When the background noise conditions of the training and the testing data are mismatched in terms of noise level and type, the degradation in speaker recognition has been observed to be even higher than the matched conditions. STSA based MMSE-LSA noise-canceller has been used as a pre-processor prior to recognition task for noise suppression. The noise-canceller has been shown not to be effective on speaker recognition performance when the training and the testing data background noise conditions are matched. However, it has been shown that the noise-canceller significantly improves the speaker recognition performance when the training and the testing data background noise conditions are mismatched. It has been also observed that, for the real-systems, the noise-canceller should be performed prior to speech coding process to achieve improvement in the speaker recognition performance.
7.2 Future Work

Possible future work is listed as follows:

- In Chapter 4, the coder mismatch problem was introduced. The coder detection system was used to reduce the performance degradation caused by mismatched coder characteristics during speaker modeling and testing. Another approach to improve the speaker recognition performance in a multi-coder environment would be to use coder independent UBMs, large background models that are designed to represent the various coder type characteristics as well as the background speaker characteristics.

- Chapter 5 described the NA-LSF parameter extraction process for speaker recognition applications. The NA-LSF parameters are estimated at every sample, which is then filtered and decimated. Parameter estimation at every sample brings extra computation load. Further work should involve determining the NA-LSF extraction rate that is less frequent than every sample, while preserving the advantages introduced by the NA-LSF method. Another possible future work is to employ NA-LSF parameter extraction method in other existing standard speech coders, which is expected to provide similar improvement in the speaker recognition performance.

- It was shown in Chapter 6 that the presence of background noise and varying noise characteristics affect the speaker recognition performance, producing high error rates. Several other approaches that might improve the speaker recognition performance when the background noise is present are as follows:
  
  - Development of UBM that can represent the possible background noise characteristics for different noise types and SNR levels.
  
  - Development of noise-robust features that can minimise the effect of noise on speaker recognition.
  
  - Development of a speaker model that can possibly contain additional information about the background noise characteristics as well as the speaker's voice characteristics.
7.2. Future Work

- Current front-end feature extraction and speaker modeling techniques of speaker recognition systems are not robust against the mismatches in the training and the testing data, and variabilities in the speech and speaker characteristics. Development of features that can decouple the speaker characteristics and channel/environmental characteristics are required. Further research on using high-level features [43], such as prosodic features, combined with low-level features, such as cepstral parameters, should lead to better speaker modeling, consequently better speaker recognition performance. With efficient front-end feature extraction and development of speaker recognition techniques without speaker modeling [126] should minimise sensitivity of recognition systems against data variabilities and mismatches.

- Another approach to improve the speaker recognition performance is to design a system that can actually mimic human speech recognition, which performs much better speaker recognition than its machine counterpart [127]. With the guidance of human speech recognition knowledge, design of such a system should provide very efficient speaker recognition robust against background noise, variabilities in the speech and speaker characteristics, and data mismatches. However the amount of available information about the human speech recognition is very limited and requires further research.
Appendix A

Male and Female Speaker Verification Experiments for Chapter 8

The tables depicted in Appendix A is provided for the interested readers, which show separate tables for EER values of male and female speaker verification experiments that are presented in Chapter 6: "Noise Cancellation for Speaker Verification".
The first part of Table A.1 shows the EER values of clean, noisy, and noise-cancelled male speaker verification experiments. The second part of Table A.1 shows the EER values of clean-coded, noisy-coded, and noise-cancelled coded (using GSM-AMR 12.2 kb/s speech coder) male speaker verification experiments. The third part of Table A.1 shows the EER values of the coded noisy male speech when the noise-canceller is used as a post-processor, following the speech coding process.

<table>
<thead>
<tr>
<th>Row</th>
<th>Training Speech</th>
<th>Testing Speech</th>
<th>EER (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Clean</td>
<td>Clean</td>
<td>1.34</td>
</tr>
<tr>
<td>B</td>
<td>Noisy</td>
<td>Noisy</td>
<td>3.21</td>
</tr>
<tr>
<td>C</td>
<td>Noise-Cancelled</td>
<td>Noise-Cancelled</td>
<td>3.57</td>
</tr>
<tr>
<td>D</td>
<td>Clean</td>
<td>Noisy</td>
<td>8.55</td>
</tr>
<tr>
<td>E</td>
<td>Clean</td>
<td>Noise-Cancelled</td>
<td>6.29</td>
</tr>
<tr>
<td>F</td>
<td>Clean-Coded</td>
<td>Clean-Coded</td>
<td>2.23</td>
</tr>
<tr>
<td>G</td>
<td>Noisy-Coded</td>
<td>Noisy-Coded</td>
<td>4.07</td>
</tr>
<tr>
<td>H</td>
<td>Noise-Cancelled-Coded</td>
<td>Noise-Cancelled-Coded</td>
<td>4.02</td>
</tr>
<tr>
<td>I</td>
<td>Clean-Coded</td>
<td>Noisy-Coded</td>
<td>8.42</td>
</tr>
<tr>
<td>J</td>
<td>Clean-Coded</td>
<td>Noise-Cancelled-Coded</td>
<td>6.61</td>
</tr>
<tr>
<td>K</td>
<td>Noisy-Coded-Noise_Canc</td>
<td>Noisy-Coded-Nois_Canc</td>
<td>4.05</td>
</tr>
<tr>
<td>L</td>
<td>Clean-Coded</td>
<td>Noisy-Coded-Nois_Canc</td>
<td>8.93</td>
</tr>
</tbody>
</table>

Table A.1: EER values of verification system for male speech using TIMIT8k, Noisy TIMIT8k corrupted with vehicle noise (15 dB SNR), and noise-cancelled speech using MMSE-LSA noise canceller.
Appendix A. Male and Female Speaker Verification Experiments for Chapter 8

The first part of Table A.2 shows the EER values of clean, noisy, and noise-cancelled female speaker verification experiments. The second part of Table A.2 shows the EER values of clean-coded, noisy-coded, and noise-cancelled coded (using GSM-AMR 12.2 kb/s speech coder) female speaker verification experiments. The third part of Table A.2 shows the EER values of the coded noisy female speech when the noise-canceller is used as a post-processor, following the speech coding process.

<table>
<thead>
<tr>
<th>Row</th>
<th>Training Speech</th>
<th>Testing Speech</th>
<th>EER (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Clean</td>
<td>Clean</td>
<td>1.79</td>
</tr>
<tr>
<td>B</td>
<td>Noisy</td>
<td>Noisy</td>
<td>3.57</td>
</tr>
<tr>
<td>C</td>
<td>Noise-Cancelled</td>
<td>Noise-Cancelled</td>
<td>3.57</td>
</tr>
<tr>
<td>D</td>
<td>Clean</td>
<td>Noisy</td>
<td>8.04</td>
</tr>
<tr>
<td>E</td>
<td>Clean</td>
<td>Noise-Cancelled</td>
<td>6.06</td>
</tr>
<tr>
<td>F</td>
<td>Clean-Coded</td>
<td>Clean-Coded</td>
<td>2.52</td>
</tr>
<tr>
<td>G</td>
<td>Noisy-Coded</td>
<td>Noisy-Coded</td>
<td>3.57</td>
</tr>
<tr>
<td>H</td>
<td>Noise-Cancelled-Coded</td>
<td>Noise-Cancelled-Coded</td>
<td>3.57</td>
</tr>
<tr>
<td>I</td>
<td>Clean-Coded</td>
<td>Noisy-Coded</td>
<td>6.95</td>
</tr>
<tr>
<td>J</td>
<td>Clean-Coded</td>
<td>Noise-Cancelled-Coded</td>
<td>6.14</td>
</tr>
<tr>
<td>K</td>
<td>Noisy-Coded-Noise_Cancel</td>
<td>Noisy-Coded-Noise_Cancel</td>
<td>3.69</td>
</tr>
<tr>
<td>L</td>
<td>Clean-Coded</td>
<td>Noisy-Coded-Noise_Cancel</td>
<td>7.14</td>
</tr>
</tbody>
</table>

Table A.2: EER values of verification system for female speech using TIMIT8k, Noisy TIMIT8k corrupted with vehicle noise (15 dB SNR), and noise-cancelled speech using MMSE-LSA noise canceller.
Table A.3 shows male speaker EER values for speaker verification experiments, where the training data are noise corrupted speech (15 dB SNR, vehicular noise) and the test data are noise corrupted speech with different background noise types and SNR values.

<table>
<thead>
<tr>
<th>Background Noise Type</th>
<th>Vehicle</th>
<th>Babble</th>
<th>Gaussian</th>
</tr>
</thead>
<tbody>
<tr>
<td>Row SNR (dB)</td>
<td>EER (%)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A 5</td>
<td>21.43</td>
<td>14.29</td>
<td>38.91</td>
</tr>
<tr>
<td>B 15</td>
<td>3.21</td>
<td>6.59</td>
<td>18.75</td>
</tr>
<tr>
<td>C 25</td>
<td>5.36</td>
<td>7.66</td>
<td>7.59</td>
</tr>
</tbody>
</table>

Table A.3: Male speaker verification EER values for noisy training speech (15 dB SNR, vehicular noise) and noisy test speech with different noise types and SNRs

Table A.4 shows female speaker EER values for speaker verification experiments, where the training data are noise corrupted speech (15 dB SNR, vehicular noise) and the test data are noise corrupted speech with different background noise types and SNR values.

<table>
<thead>
<tr>
<th>Background Noise Type</th>
<th>Vehicle</th>
<th>Babble</th>
<th>Gaussian</th>
</tr>
</thead>
<tbody>
<tr>
<td>Row SNR (dB)</td>
<td>EER (%)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A 5</td>
<td>26.97</td>
<td>34.82</td>
<td>46.43</td>
</tr>
<tr>
<td>B 15</td>
<td>3.57</td>
<td>18.53</td>
<td>26.01</td>
</tr>
<tr>
<td>C 25</td>
<td>5.41</td>
<td>11.83</td>
<td>12.61</td>
</tr>
</tbody>
</table>

Table A.4: Female speaker verification EER values for noisy training speech (15 dB SNR, vehicular noise) and noisy test speech with different noise types and SNRs
Table A.5 shows male speaker EER values for speaker verification experiments using noise-cancelled mismatched training and test data.

<table>
<thead>
<tr>
<th>Background Noise Type</th>
<th>Vehicle</th>
<th>Babble</th>
<th>Gaussian</th>
</tr>
</thead>
<tbody>
<tr>
<td>Row SNR (dB)</td>
<td>EER (%)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>5</td>
<td>19.29</td>
<td>16.15</td>
</tr>
<tr>
<td>B</td>
<td>15</td>
<td>3.57</td>
<td>5.85</td>
</tr>
<tr>
<td>C</td>
<td>25</td>
<td>5.70</td>
<td>5.80</td>
</tr>
</tbody>
</table>

Table A.5: Male speaker verification EER values for noise-cancelled training speech (15 dB SNR, vehicular noise) and noise-cancelled test speech with different noise types and SNRs.

Table A.6 depicts male speaker EER values for speaker verification experiments using mismatched noise-cancelled training data and noisy test data.

<table>
<thead>
<tr>
<th>Background Noise Type</th>
<th>Vehicle</th>
<th>Babble</th>
<th>Gaussian</th>
</tr>
</thead>
<tbody>
<tr>
<td>Row SNR (dB)</td>
<td>EER (%)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>5</td>
<td>30.80</td>
<td>17.06</td>
</tr>
<tr>
<td>B</td>
<td>15</td>
<td>6.70</td>
<td>6.70</td>
</tr>
<tr>
<td>C</td>
<td>25</td>
<td>6.25</td>
<td>5.78</td>
</tr>
</tbody>
</table>

Table A.6: Male speaker verification EER values for noise-cancelled training speech (15 dB SNR, vehicular noise) and noisy test speech with different noise types and SNRs.
Table A.7 shows female speaker EER values for speaker verification experiments using noise-cancelled mismatched training and test data.

<table>
<thead>
<tr>
<th>Background Noise Type</th>
<th>Vehicle</th>
<th>Babble</th>
<th>Gaussian</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Row</strong></td>
<td><strong>SNR (dB)</strong></td>
<td><strong>EER (%)</strong></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>5</td>
<td>18.87</td>
<td>32.94</td>
</tr>
<tr>
<td>B</td>
<td>15</td>
<td>3.57</td>
<td>7.95</td>
</tr>
<tr>
<td>C</td>
<td>25</td>
<td>7.18</td>
<td>6.40</td>
</tr>
</tbody>
</table>

Table A.7: Female speaker verification EER values for noise-cancelled training speech (15 dB SNR, vehicular noise) and noise-cancelled test speech with different noise types and SNRs

Table A.8 depicts female speaker EER values for speaker verification experiments using mismatched noise-cancelled training data and noisy test data.

<table>
<thead>
<tr>
<th>Background Noise Type</th>
<th>Vehicle</th>
<th>Babble</th>
<th>Gaussian</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Row</strong></td>
<td><strong>SNR (dB)</strong></td>
<td><strong>EER (%)</strong></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>5</td>
<td>29.46</td>
<td>35.80</td>
</tr>
<tr>
<td>B</td>
<td>15</td>
<td>6.26</td>
<td>15.97</td>
</tr>
<tr>
<td>C</td>
<td>25</td>
<td>7.32</td>
<td>7.36</td>
</tr>
</tbody>
</table>

Table A.8: Female speaker verification EER values for noise-cancelled training speech (15 dB SNR, vehicular noise) and noisy test speech with different noise types and SNRs
Table A.9 shows male speaker EER values for speaker verification experiments using noise-cancelled test data, where clean speech is used as training data.

<table>
<thead>
<tr>
<th>Background Noise Type</th>
<th>Vehicle</th>
<th>Babble</th>
<th>Gaussian</th>
</tr>
</thead>
<tbody>
<tr>
<td>Row</td>
<td>SNR (dB)</td>
<td>EER (%)</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>5</td>
<td>24.11</td>
<td>15.62</td>
</tr>
<tr>
<td>B</td>
<td>15</td>
<td>6.29</td>
<td>3.99</td>
</tr>
<tr>
<td>C</td>
<td>25</td>
<td>2.71</td>
<td>1.89</td>
</tr>
</tbody>
</table>

Table A.9: Male speaker verification EER values for clean training speech and noise-cancelled test speech with different noise types and SNRs

Table A.10 shows male speaker EER values for speaker verification experiments using noisy test data, where clean speech is used as training data.

<table>
<thead>
<tr>
<th>Background Noise Type</th>
<th>Vehicle</th>
<th>Babble</th>
<th>Gaussian</th>
</tr>
</thead>
<tbody>
<tr>
<td>Row</td>
<td>SNR (dB)</td>
<td>EER (%)</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>5</td>
<td>28.05</td>
<td>16.55</td>
</tr>
<tr>
<td>B</td>
<td>15</td>
<td>8.55</td>
<td>4.04</td>
</tr>
<tr>
<td>C</td>
<td>25</td>
<td>3.10</td>
<td>2.67</td>
</tr>
</tbody>
</table>

Table A.10: Male speaker verification EER values for clean training speech and noisy test speech with different noise types and SNRs
Table A.11 shows female speaker EER values for speaker verification experiments using noise-cancelled test data, where clean speech is used as training data.

<table>
<thead>
<tr>
<th>Background Noise Type</th>
<th>SNR (dB)</th>
<th>EER (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vehicle</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Babble</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gaussian</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Row A</td>
<td>5</td>
<td>23.21</td>
</tr>
<tr>
<td>Row B</td>
<td>15</td>
<td>6.06</td>
</tr>
<tr>
<td>Row C</td>
<td>25</td>
<td>2.75</td>
</tr>
</tbody>
</table>

Table A.11: Female speaker verification EER values for clean training speech and noise-cancelled test speech with different noise types and SNRs

Table A.12 shows female speaker EER values for speaker verification experiments using noisy test data, where clean speech is used as training data.

<table>
<thead>
<tr>
<th>Background Noise Type</th>
<th>SNR (dB)</th>
<th>EER (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vehicle</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Babble</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gaussian</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Row A</td>
<td>5</td>
<td>31.25</td>
</tr>
<tr>
<td>Row B</td>
<td>15</td>
<td>8.04</td>
</tr>
<tr>
<td>Row C</td>
<td>25</td>
<td>3.57</td>
</tr>
</tbody>
</table>

Table A.12: Female speaker verification EER values for clean training speech and noisy test speech with different noise types and SNRs
Bibliography


