Audio Signal Classification in Reverberant and Noisy Environments

by
Ahmad Rami Abu-El-Quran, B.Sc., M.A.Sc.

A thesis submitted to the Faculty of Graduate Studies and
Research in partial fulfillment of the requirements for the
degree of

Doctor of Philosophy

Ottawa-Carleton Institute For Electrical and Computer Engineering
Department of Systems and Computer Engineering
Faculty of Engineering and Design
Carleton University
Ottawa, Ontario
April 2009

© Copyright 2009, Ahmad Rami Abu-El-Quran

Library and
Archives Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-52059-8
Our file Notre référence
ISBN: 978-0-494-52059-8

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

Abstract

Audio signal classification (ASC) is the automatic categorization of an audio signal into a certain class (e.g. speech, music, footsteps). ASC categorizes a test audio signal, from an unknown class, by comparing it to training audio signals, whose classes are known. While capturing the audio signals, the surrounding environment may vary; differences can include the level of reverberation and/or the level of noise. Differences in the surrounding environment between the test and the training audio signals can lead to a significant degradation in ASC performance (i.e. system accuracy and robustness in varying environments).

The conventional method to improve the performance of ASC in reverberant and noisy environments is to perform audio enhancement prior to ASC. In this method, training audio signals are obtained in a clean environment (i.e. no reverberation and no noise). The test audio signals, however, are obtained in environments that may contain reverberation and/or noise; de-noising and/or de-reverberation audio enhancements are applied to these signals to reduce the effects of the environment. Often this method enhances the audio signal audibility from the human perspective, but ignores the fact that the enhancement may remove or distort information in the audio signal; this information may be essential to the ASC process which may limit the performance of the ASC.

In this thesis, we introduced a generalized solution for ASC applications that has high performance in reverberant and noisy environments. In particular, instead of trying to

adapt the audio to the ASC system, we adapt the ASC system to the surrounding environment of the captured audio signal. This solution characterizes its environment from the captured audio and adapts the ASC to better suit the environment. In this solution, we introduced novel ASC features, multi-feature fusion, multi-engine classifier fusion, and environment information estimators to characterize the surrounding environment.

Two novel ASC features, Adaptive Pitch Ratio (APR) and Variance of Low-band Energy Ratio (VLER), are introduced. Compared to conventional methods the improvement in accuracy was as high as 34.8% and 31.0% for APR and VLER, respectively. Multi-Feature Selection and Multi-Feature Combination ASC methods were introduced. Compared to conventional methods the improvement in accuracy was as high as 33.0% and 34.0% for Multi-Feature Selection and Multi-Feature Combination, respectively.

Multi-Engine Selection and Multi-Engine Weighted Fusion ASC methods were introduced. Compared to conventional methods the improvement in accuracy was as high as 9.0% and 10.0% for Multi-Engine Selection and Multi-Engine Weighted Fusion, respectively. Environment information estimators are introduced; Reverberation Sensing System (RSS) and Discrete-Valued SNR Estimator, with classification accuracy of 87.5% and 98.4%, respectively.

Acknowledgments

I would like to express my appreciation for the valuable efforts of *Dr. Rafik Goubran* and *Dr. Adrian Chan* in supervising this research. I would also like to sincerely thank *Dr. Goubran* and *Dr. Chan* for their ongoing support and encouragement.

I would like to acknowledge the funding from Mitel, OGS, and NSERC.

I am especially grateful to my professors at Ain Shams University, Cairo, Egypt, Carleton University, Ottawa, Canada and University of Ottawa, Ottawa, Canada, for the strong technical background that I have learned from my undergraduate and graduate courses. Also I would like to thank my colleagues who have helped me during my ongoing studies.

Finally, I would like to thank my family for their support and understanding and the sacrifices they have made through the years of this research. I would like to dedicate this thesis to the memory of my father *Mahmoud*. Also I would like to dedicate this thesis to my mother *Salwa*, my beautiful wife *Ruba*, my sons *Mahmoud* and *Zeyad*, and my little princess *Malak*.

Table of Contents

Abstract.....	iii
Acknowledgments	v
Table of Contents	vi
List of Tables	ix
List of Figures.....	x
List of Abbreviations	xiv
Chapter 1 Introduction.....	1
1.1 Thesis Motivation	1
1.2 Problem Definition.....	3
1.3 Thesis Objectives	5
1.4 Contributions.....	6
1.4.1 Major Contributions.....	6
1.4.2 Minor Contributions	12
1.5 Thesis Outline	15
Chapter 2 Overview of Audio Signal Classification.....	18
2.1 History of ASC.....	18
2.2 Overview of ASC Process.....	19
2.2.1 Audio Capturing Devices.....	22
2.2.2 Feature Extraction.....	23
2.2.2.1 Energy	24
2.2.2.2 Zero Crossing Rate	25
2.2.2.3 Periodicity (Pitch Detection).....	27
2.2.2.4 Mel-Frequency Cepstrum Coefficients (MFCC)	28
2.2.2.5 4 Hz Modulation	30
2.2.2.6 Entropy.....	31
2.2.2.7 Segmentation Count.....	31
2.2.2.8 Spectral Centroid	32
2.2.3 Classifiers.....	33
2.2.3.1 Linear Classifiers	33
2.2.3.2 Non-linear Classifiers	34
2.3 ASC Categories.....	37
2.3.1 Coarse ASC.....	38
2.3.1.1 Speech/Non-Speech Classification	38
2.3.1.2 Speech/Music Classification	39
2.3.2 Fine ASC.....	40
2.3.2.1 Talker Identification.....	40
2.3.3 Combined ASC	41
2.3.3.1 Audio Indexing and Retrieval	41
2.4 ASC in Variable Reverberant and Noisy Environments.....	42
2.4.1 ASC in Reverberant Environment	43

2.4.2	ASC in Noisy Environment	45
2.5	Discussion	51
Chapter 3	Experimental and Simulation Environments Setup	53
3.1	Audio Database	53
3.1.1	Speech Database	54
3.1.2	Speech/Music Database	54
3.1.3	Speech/Non-Speech Database	54
3.2	Environment Models	55
3.2.1	Reverberant Environments	56
3.2.1.1	Reverberation Models (Image Method)	57
3.2.1.2	Reverberation Specifications	60
3.2.2	Noisy Environments	61
3.2.2.1	Noise Models	61
3.2.2.2	Noise Specifications	61
3.3	Simulation Environment (Hardware and Software)	62
3.4	Performance Evaluation Criteria	62
3.4.1	Baseline Comparison Method	62
3.4.2	Classification Accuracy	62
3.4.3	Matthews's Correlation Coefficient	64
Chapter 4	Performance of ASC in Reverberant and Noisy Environments	66
4.1	Performance of ASC in Clean Environments	66
4.1.1	Speech/Non-Speech Classification	67
4.1.2	Speech/Music Classification	73
4.1.3	Talker Identification	74
4.2	Performance of ASC in Reverberant Environments	75
4.2.1	Speech/Non-Speech Classification	75
4.2.2	Talker Identification	77
4.3	Performance of ASC in Noisy Environments	79
4.3.1	Speech/Non-Speech Classification	79
4.3.2	Speech/Music Classification	80
4.3.3	Talker Identification	82
4.4	Discussion	83
Chapter 5	Solutions for ASC in Reverberant and Noisy Environments	85
5.1	ASC in Reverberant and Noisy Environments	85
5.2	Feature Extraction	87
5.2.1	Adaptive Feature Threshold	89
5.2.2	Multi-Feature Fusion ASC	90
5.2.2.1	Multi-Feature Selection	91
5.2.2.2	Multi-Expert Method	92
5.2.2.3	Multi-Feature Combination	97
5.3	Environment Information Estimator	99
5.4	Classifier Engines	100
5.4.1	Multi-Engine Classifier in Variable Environments	100
5.4.1.1	Multi-Engine ASC	101

5.4.1.2	Multi-Engine Selection	102
5.4.1.3	Multi-Engine Weighted Fusion.....	103
5.5	Discussion	105
Chapter 6	Feature Extraction in Variable Environments.....	106
6.1	Resilient and Adaptive Features	107
6.1.1	Adaptive Pitch Ratio algorithm (APR).....	107
6.1.1.1	APR in Variable Reverberant Environment	107
6.1.1.2	APR in Variable Noisy Environment	109
6.1.2	Variance of Low-Band Energy Ratio Algorithm (VLER).....	113
6.1.3	Adaptive Feature Threshold in Noisy Environments.....	118
6.2	Multi-Feature Methods	125
6.2.1	Multi-Feature Selection Method	126
6.2.2	Multi-Feature Combination Method	127
6.3	Discussion	129
Chapter 7	Environment Information Estimators with ASC.....	131
7.1	Reverberation Sensing System (RSS).....	132
7.1.1	RSS Implementation	132
7.2	Multi-Engine Selection Method in Reverberant Environment	136
7.3	Multi-Engine Weighted Fusion Method in Reverberant Environment.....	138
7.4	Discrete-Valued SNR Estimator	140
7.5	Multi-Engine Selection Method in Noisy Environment	142
7.6	Multi-Engine Weighted Fusion Method in Noisy Environment.....	144
7.7	Multi-Engine Selection and Weighted Fusion Methods in Real Environments	146
7.8	Multi-Engine Fusion Methods at Hybrid Reverberant and Noisy Environments	150
7.9	Discussion	154
Chapter 8	Application of ASC Systems in Variable Environments.....	155
8.1	Security System Overview.....	156
8.2	Overall Performance of the Combined ASC System.....	158
8.3	Discussion	160
Chapter 9	Conclusion and Future Work	161
9.1	Conclusion	161
9.2	Future Work.....	163
Appendix A	Microphone Array	166
A.1	Linear Delay-and-Sum Beamformer.....	166
Appendix B	Reverberant Environments Setup	169
References		175

List of Tables

Table 2.2.1 Main audio features	24
Table 2.4.1 ASR classification accuracy improvement with White, and Babble noise....	48
Table 3.1.1 Speech/non-speech audio database.....	55
Table 3.2.1 RT60 simulation values	60
Table 4.1.1 Comparison of Speech/Non-speech classification accuracy using three algorithms in clean environment (%).....	73
Table 4.1.2 Comparison of speech/music classification accuracy using seven algorithms in clean environment.....	73
Table 7.1.1 RSS classification results (%).....	135
Table 7.4.1 Discrete-valued SNR estimator classification results (%).....	142
Table 7.7.1 Real Environment RSS selection and classification accuracy of the individual TI engine.....	148
Table 7.7.2 Talker identification accuracy using different methods in real environments.....	150
Table 8.2.1 Speech/non-speech classification results (%).....	159
Table B.1.1: Reverberation specifications for $d = 0.75$ m	169
Table B.1.2: Reverberation specifications for $d = 0.75$ m continued (1).....	169
Table B.1.3: Reverberation specifications for $d = 0.75$ m continued (2).....	170
Table B.1.4: Reverberation specifications for $d = 0.75$ m continued (3).....	170
Table B.1.5: Reverberation specifications for $d = 0.87$ m.....	170
Table B.1.6: Reverberation specifications for $d = 0.87$ m continued (1).....	170
Table B.1.7: Reverberation specifications for $d = 0.87$ m continued (2).....	171
Table B.1.8: Reverberation specifications for $d = 0.87$ m continued (3).....	171
Table B.1.9: Reverberation specifications for $d = 1.00$ m	171
Table B.1.10: Reverberation specifications for $d = 1.00$ m continued (1).....	171
Table B.1.11: Reverberation specifications for $d = 1.00$ m continued (2).....	172
Table B.1.12: Reverberation specifications for $d = 1.00$ m continued (3).....	172
Table B.1.13: Reverberation specifications for $d = 1.25$ m	172
Table B.1.14: Reverberation specifications for $d = 1.25$ m continued (1).....	172
Table B.1.15: Reverberation specifications for $d = 1.25$ m continued (2).....	173
Table B.1.16: Reverberation specifications for $d = 1.25$ m continued (3).....	173

List of Figures

Figure 1.1.1 Audio signal classification categories	2
Figure 1.2.1 Graphical illustration of how an ASC trained in a clean environment has a classification accuracy that decreases with increasing noise or reverberation.	4
Figure 1.3.1 Introduced ASC solution	5
Figure 2.2.1 Speech waveform of voiced ($\bar{a}$) and unvoiced (sh) types.....	21
Figure 2.2.2 Audio signal classification process	21
Figure 2.2.3 Audio capturing device Arc.....	22
Figure 2.2.4 Unidirectional audio capturing device (Mitel®)	23
Figure 2.2.5 Short time average energy	24
Figure 2.2.6 Zero crossing rate of audio signal	26
Figure 2.2.7 Pitch detection of audio signal	27
Figure 2.2.8 MFCC calculation steps	28
Figure 2.2.9 Nonlinear Mel-scale filters (10 filters designed for 8 Khz audio signal)	28
Figure 2.2.10 4 Hz Modulation Algorithm	30
Figure 2.2.11 Entropy calculation.....	31
Figure 2.2.12 Music segmentation.....	32
Figure 2.2.13 Speech segmentation.....	32
Figure 2.2.14 Linear classifier (two classes and two features).....	33
Figure 2.2.15. Clustering of a single audio class feature vectors	34
Figure 2.2.16. Multi layer perceptron neural network structure	37
Figure 2.4.1 Room Identification System	44
Figure 2.4.2 Mode selection in hearing aids device.....	45
Figure 2.4.3 ASR classification accuracy with White noise.....	47
Figure 2.4.4 ASR classification accuracy with Babble noise	47
Figure 2.4.5 Adaptive VAD with ASR.....	50
Figure 2.4.6 VAD selection algorithm.....	51
Figure 3.2.1 Simulated environment.....	56
Figure 3.2.2 Real environment.....	56
Figure 3.2.3 Reverberation	57
Figure 3.2.4 First order reflection.....	58
Figure 3.2.5 Second order reflection.....	59
Figure 3.2.6 Spatial arrangement of image sources used in the simulation program	59
Figure 3.4.1 Classification accuracy parameters	63
Figure 3.4.2 Confusion table of classification accuracy	64
Figure 4.1.1 Audio segments and frames.....	67
Figure 4.1.2 Pitch Ratio Algorithm.....	69

Figure 4.1.3 (a) PR distribution of speech and non-speech audio segment (b)MCC variation with PR threshold. (c) Speech and non-speech classification accuracies with the threshold.	72
Figure 4.2.1 Comparison of Speech/Non-Speech classification accuracy using three algorithms in reverberant environment (%) (PR parameter $FS = 60$ ms, $OL = 0.85$ and PR Threshold = 0.32).....	77
Figure 4.2.2 Classification accuracy of TI engine (trained in clean environment) in variable reverent environment	78
Figure 4.3.1 Comparison of Speech/Non-Speech classification accuracy using three algorithms in noisy environment (%) (PR parameter $FS = 60$ ms, $OL = 0.85$ and PR Threshold = 0.32)	80
Figure 4.3.2 Comparison of Speech/music classification accuracy using seven algorithms in noisy environment.....	81
Figure 4.3.3 Speech/music classification distributions using ZCR and fixed threshold at different SNR (a) SNR = -5 dB (b) SNR= 2 dB (c) SNR = 40 dB.	82
Figure 4.3.4 Classification accuracy of TI engine trained in clean environment and tested in variable noisy environment	83
Figure 5.1.1 ASC generalization in variable environments.....	87
Figure 5.2.1 (a) Features at clean env. (b) First scenario after env. change with clear decision threshold. (c) Second scenario after env. change without clear decision threshold.	88
Figure 5.2.2 Adaptive Feature Threshold	90
Figure 5.2.3 Multi-Feature ASC	91
Figure 5.2.4 Multi-Feature Selection method	91
Figure 5.2.5 Multi-Feature Selection with Adaptive Feature Threshold.....	92
Figure 5.2.6 Abstract information level (example).....	93
Figure 5.2.7 Rank information level (example).....	94
Figure 5.2.8 Measurement information level (example).	94
Figure 5.2.9 Multi-Feature Combination method	99
Figure 5.4.1 Solution for ASC in variable environment.....	101
Figure 5.4.2 Multi-Engine Selection method.....	102
Figure 5.4.3 Multi-Engine Weighted Fusion method	103
Figure 6.1.1 Speech/non-speech classification accuracy vs. RT60, using adaptive and fixed PR. ($FS = 60$ ms, $OL = 0.85$, and $SD = 450$ ms).....	108
Figure 6.1.2 Comparison of Speech/Non-Speech classification accuracy using four algorithms in reverberant environment (%) (PR parameter $FS = 60$ ms, $OL = 0.85$)	109
Figure 6.1.3 Threshold and corresponding speech / non-speech classification accuracy vs. frame size and overlap (SNR = 30 dB, and $SD = 450$ ms).....	110
Figure 6.1.4 Adaptive threshold and corresponding speech / non-speech classification accuracy vs. frame size and SNR ($OL = 0.85$, and $SD = 450$ ms)	111
Figure 6.1.5 Speech/non-speech classification accuracy vs. SNR, using adaptive and fixed PR. ($FS = 60$ ms, $OL = 0.85$, and $SD = 450$ ms)	112

Figure 6.1.6 Comparison of Speech/Non-Speech classification accuracy using four algorithms in noisy environment (%).....	113
Figure 6.1.7 Variance of low-band energy ratio algorithm (VLER)	114
Figure 6.1.8 Power spectral density of voiced ($\bar{a}$) and unvoiced types (sh) (sampling rate 22.05 kHz).....	116
Figure 6.1.9 Speech/music classification accuracy vs. SNR, using adaptive and fixed VLER	116
Figure 6.1.10 Comparison of Speech/music classification accuracy using nine algorithms in noisy environment.....	118
Figure 6.1.11 (a) 4Hz (b) VLER algorithms distribution at SNR = -20	119
Figure 6.1.12 (a) 4Hz (b) VLER algorithms distribution at SNR = -10 dB	119
Figure 6.1.13 (a) 4Hz (b) VLER algorithms distribution at SNR = 0 dB.....	120
Figure 6.1.14 (a) 4Hz (b) VLER algorithms distribution at SNR = 10 dB.....	120
Figure 6.1.15 (a) 4Hz (b) VLER algorithms distribution at clean environment.....	121
Figure 6.1.16 Speech/music classification accuracy vs. SNR, using adaptive and fixed Entropy (left) and Segmentation (right).....	122
Figure 6.1.17 Speech/music classification accuracy vs. SNR, using adaptive and fixed APR (left) and Centroid (right).....	122
Figure 6.1.18 Speech/music classification accuracy vs. SNR, using adaptive and fixed Low Energy Frames (left) and ZCR (right).....	123
Figure 6.1.19 Classification accuracy of adaptive threshold and classification accuracy gain of adaptive threshold over fixed threshold classification vs. SNR	124
Figure 6.1.20 Adaptive threshold classification accuracy vs. segment duration	125
Figure 6.2.1 Classification accuracy using best-performing feature selection algorithm vs. SNR (dB)	127
Figure 6.2.2 Classification accuracy using Multi-Feature Combination algorithm vs. SNR (dB).....	129
Figure 7.2.1 Talker identification accuracy using Multi-Engine Selection method vs. RT60.....	137
Figure 7.3.1 Talker identification accuracy using Multi-Engine Weighted Fusion method vs. RT60	139
Figure 7.5.1 Talker identification accuracy using Multi-Engine Selection method vs. SNR.....	143
Figure 7.6.1 Talker identification accuracy using Multi-Engine Weighted Fusion method vs. SNR	144
Figure 7.7.1 Real Environment setup (ME4356).....	147
Figure 7.7.2: Illustration of the real environment setups (a) ME 4359 (b) ME 4356.....	147
Figure 7.8.1 Talker identification accuracy using Multi-Engine Weighted Fusion and Multi-Engine Selection methods at different reverberant environments vs. SNR using white noise	152
Figure 8.1.1 Introduced security system	156
Figure 8.2.1 Overall accuracy of the introduced system	159
Figure A.1.1 Delay-and-sum beamformer	167

Figure A.1.2 Incident of plan wave on a linear microphone array	168
Figure A.1.3 Beampattern of six element linear array steered to 90° from the array axis. Radial scale indicates the normalized directive gain	168
Figure B.1.4 Impulse response and energy decay curve at $d=1.00$ m and RT60 = 0.25 sec.....	173
Figure B.1.5 Impulse response and energy decay curve at $d=1.00$ m and RT60 = 1.00 sec.....	174
Figure B.1.6 Impulse response and energy decay curve at $d=1.00$ m and RT60 = 1.80 sec.....	174

List of Abbreviations

APR	adaptive pitch ratio
AR-AGS	autoregressive using arithmetic-geometry sphericity
ARVM	autoregressive vector model
ASC	audio signal classification
ASR	automatic speech recognition
BP	back propagation
BPF	band-pass filter
DCT	discrete cosine transform
DTX	discontinuous transmission
EDC	energy decay curve
EM	expectation maximization
FF	feed forward
FFT	fast Fourier transform
FN	false negative
FP	false positive
FS	frame size
GSM	global system for mobile communications
GMM	Gaussian mixture model
HOS	high order statistics
HMM	hidden Markov model

HPF	high-pass filter
LBG	Linde, Buzo, Gray clustering algorithm
LER	low-band energy ratio
LPC	linear prediction coefficient
LPCC	linear prediction cepstral coefficient
LPF	low-pass filter
LSP	line spectrum pairs
MCC	Matthews correlation coefficient
MFCC	mel frequency cepstral coefficients
MFSC	mel frequency spectral coefficients
MLP	multi layer perceptron
NF	number of frames in one segment
NN	neural network
OL	overlap duration
PDF	probability density function
PR	pitch ratio
RMS	root mean squared
RMSE	root mean squared energy
RSS	reverberation sensing system
RT60	reverberation time 60 sec
SD	segment duration
SNR	signal to noise ratio

SVM	support vector machine
TASI	time assignment speech interpolation
TI	talker identification
TN	true negative
TP	true positive
VAD	voice activity detector
VLER	variance of low-band energy ratio
VQ	vector quantization
WGN	white Gaussian noise
WT	wavelet transform
ZCR	zero crossing rate

Chapter 1

Introduction

1.1 Thesis Motivation

Audio signal classification (ASC) is defined as the automatic categorization of an audio signal into a certain class (e.g. speech, music, environment sounds). ASC categorizes a test audio signal, from an unknown class, by comparing it to training audio signals, whose classes are known.

ASC coarse and fine categories are shown in Figure 1.1.1. The coarse level mainly performs a number of binary classifications (two audio classes), such as speech/non-speech classification. Fine level performs further audio classification after the main audio class is determined. For example, performing talker identification after the coarse level determines that the main audio class of the input audio signal is speech. ASC can be deployed in many applications, such as biomedical [1], [2], audio-video conferencing [3], and multimedia applications [4], [5].

Figure 1.1.1 Audio signal classification categories

An example of a biomedical application is hearing aids. Hearing aids can utilize the ASC to determine the class of the captured audio signal (e.g. music, speech) to customize its parameters to achieve the optimal hearing performance [6], [7].

Modern audio-video conferencing using a microphone array and a steerable camera can utilize the ASC. ASC is utilized to determine whether the sound source is a talker or background noise, enabling the system to avoid steering the camera to undesired locations (e.g. to a noisy fan, instead of a talker) [8].

With the massive amount of digitally stored data, multimedia applications, such as audio transcription and audio indexing and retrieval, need ASC to determine the class of the enquired audio signal [9], [10].

Applications that use ASC, including those mentioned previously, require ASC with high performance (i.e. accuracy and robustness) to have successful functionality. Poor ASC performance would limit utility of these ASC applications.

1.2 Problem Definition

ASC classifiers have two different operating modes: a training mode and a testing mode. The goal of the training mode is to allow the classifier to learn how to discern between different known audio classes. The purpose of the testing mode is the discernment of unknown audio data based on the training.

The performance of most ASC systems is degraded severely when the training and the testing audio are captured in different acoustic environments. Figure 1.2.1 illustrates this graphically, where high performance can be obtained for environments with low level of reverberation and noise; however, as the level of reverberation or noise increases, the accuracy of the ASC decreases. For example, training audio may be captured using a telephone handset and clean environment (no reverberation and no noise), while the test audio are captured using a hands-free talker phone in a non-clean environment. Simultaneously, the effect of the mismatch between training and test acoustic environment of the audio signal has a little research. This mismatch resulted due to the existence of the reverberation and the noise [11], [12].

Figure 1.2.1 Graphical illustration of how an ASC trained in a clean environment has a classification accuracy that decreases with increasing noise or reverberation.

Previous efforts in [11], [13], [14] to improve ASC system performance in reverberant and noisy environments have focused on developing ASC systems to function in a specific environment (*a priori* knowledge of the environment is required). This improvement may not be generalized to variable reverberant and noisy environments; they are tuned to certain environments and ignored the dynamics in the surrounding environments.

In [12], [15], [16] the performance of the ASC is improved using methods that do not rely on *a priori* knowledge of the surrounding environment. These methods perform audio enhancement prior to the ASC. In these methods, training audio signals are obtained in a clean environment (i.e. no reverberation and no noise). The test audio signals are obtained in environments that may contain reverberation and/or noise; denoising and/or de-reverberation audio enhancements are applied to these signals to reduce the effects of the environment. Often this method enhances the audio signal audibility from the human perspective, but ignores the fact that the enhancement may remove or

distort information in the audio signal. This information may be essential to the ASC process which may limit the performance of the ASC [12].

1.3 Thesis Objectives

The main objective of this research is to introduce a generalized solution for ASC applications that has high performance in variable reverberant and noisy environments. We will be developing algorithms to mitigate the poor robustness and the degradation of the ASC performance where the level of noise and/or reverberation varies.

As mentioned in Section 1.2, previous efforts have been trying to adapt the audio to the ASC system to enhance the accuracy performance of the ASC. In this research, we introduce an ASC system that adapts to the surrounding environment of the audio signal. This system characterizes its environment from the captured audio signal and adapts the ASC process to better suit the environment (Figure 1.3.1). The reverberation level is characterized by reverberation time (RT60) while the noise level is characterized by signal-to-noise ratio (SNR). This adaptation enhances the overall performance of the ASC system in variable environments.

Figure 1.3.1 Introduced ASC solution

To create a generalized ASC, with high performance, a study and evaluation of the impact of the reverberant and noisy environments on the behavior of the ASC system in different environments has to be conducted (Figure 1.3.1). Using this evaluation, each entity of the ASC system should be adapted to the surrounding environment. This adaptation should improve the ASC performance. The contributions of this thesis are dedicated to a generalized solution for the ASC that has high performance in variable reverberant and noisy environments.

1.4 Contributions

This thesis has been broken up into multiple contributions to develop and validate the ASC systems in variable environments; specifically, the effects of reverberation and noise are examined. Novel methods that improve the performance of the ASC systems are presented. The major and minor contributions are summarized in the following subsections.

1.4.1 Major Contributions

1. **Introduce the use of two novel adaptive audio features: Adaptive Pitch Ratio (APR) and Variance of Low-band Energy Ratio (VLER). These features are robust in noisy and reverberant environments.**
 - a. **APR** audio feature is used to classify an audio segment as speech or non-speech. The APR algorithm looks at the presence of human pitch ($70\text{Hz} < \text{human pitch} < 280\text{Hz}$) in multiple small frames in a
-

certain audio segment. Then it determines whether or not the audio segment is speech or non-speech. This feature is capable of handling low signal-to-noise ratio, reverberant and variable environments. In this work, we have shown that it has overall classification accuracy of 92.0% in noisy environments over an SNR range of clean environment to -10 dB. Compared to conventional methods the improvement in accuracy was as high as 34.8%. Also, we have shown that it has overall classification accuracy of 95.2% in reverberant environments over an RT60 range of clean environment to 1.80 sec. Compared to conventional methods the improvement in accuracy was as high as 13.5%.

- b. **VLER** audio feature is used to classify an audio segment as speech or music. The VLER algorithm looks at the variance of the ratio of the low-band energy (*frequency < 1500 Hz*) to the whole band energy in multiple small frames in a certain audio segment duration. The introduced feature is capable of handling very low signal-to-noise and variable noisy environments. In this work, we have shown that VLER has overall classification accuracy of 85.0% in noisy environments over an SNR range of clean environment to -20 dB. Compared to conventional methods the improvement in accuracy was as high as 31.0%.

This work is published in:

- i. A.R. Abu-El-Quran, and R.A Goubran, "Adaptive Pitch-Based Speech Detection for Hands-Free Applications," in *IEEE Proc. International Conference Acoustics, Speech, and Signal Processing*, Philadelphia, PN, USA, Vol. 3, Mar. 2005, pp. 305-308.
- ii. AR Abu-El-Quran, RA Goubran, and A.D.C. Chan, "Adaptive Feature Selection for Speech/Music Classification," in *IEEE Proc. International Workshop on Multimedia Signal Processing*, Victoria, BC, Canada, Oct. 2006, pp. 212-216.

2. Introduce two methods for robust ASC in variable reverberant and noisy environments using multiple features: Multi-Feature Selection and Multi-Feature Combination.

- a. **Multi-Feature Selection method** selects the best performing feature. Feature selection is conducted using environment information (SNR and RT60). The introduced selection method is capable of handling variable noisy environments. In this work, we have shown that it has an overall classification accuracy of 87.0% in noisy environments over an SNR range of clean environment to -20 dB. Compared to conventional methods the improvement in accuracy was as high as 33.0%.
-

- b. **Multi-Feature Combination method** weights and combines the features using environment information. The introduced combination method is capable of handling variable noisy environments. In this work, we have shown that it has overall classification accuracy of 88% in noisy environments over an SNR range of clean environment to -20 dB. Compared to conventional methods the improvement in accuracy was as high as 34.0%.

This work is published in:

AR Abu-El-Quran, RA Goubran, and A.D.C. Chan, "Adaptive Feature Selection for Speech/Music Classification," in *IEEE Proc. International Workshop on Multimedia Signal Processing*, Victoria, BC, Canada, Oct. 2006, pp. 212-216.

3. **Introduce novel Discrete-Valued SNR Estimator** system that is computationally efficient, with low complexity. This system provides information about the noisy surrounding environment (environment classification), which is used in this work to adapt the ASC to the surrounding environment. In this work, we have shown that it has environment classification accuracy of 98.4%.

This work is published in:

A.R. Abu-El-Quran, R.A. Goubran, and A.D.C. Chan, "Security Instrument using Talker Identification and Microphone Arrays in Variable Noisy Environments," in *IEEE proc. International Instrumentation, and Measurement Technology Conference*, Victoria, BC, Canada, May, 2008, pp. 1046-1050.

4. Introduce two methods for robust ASC in variable reverberant and noisy environments using multiple engines (i.e. multiple classifiers): Multi-Engine Selection and Multi-Engine Weighted Fusion.

- a. **Multi-Engine Selection method** has multiple classifiers. Each classifier is trained in a different SNR or RT60 level environment. Using these classifiers and information about the environment (e.g. estimate of noise level or reverberation level), a selection method is developed to select the classifier which has been trained in a similar environment. ASC accuracy, using a single classifier, is highest when the test data from a particular environment is used with a classifier trained in the same environment. By applying this method on talker identification (TI) system, we have shown that it has overall classification accuracy of 86% in noisy environments over an SNR range of clean environment to 0 dB. Compared to conventional methods the improvement in accuracy was as high as 43.0%. Also, we have shown that it has overall classification accuracy 95.4% in reverberant environments over an RT60 range of clean environment

to 1.80 sec. Compared to conventional methods the improvement in accuracy was as high as 9%.

- b. **Multi-Engine Weighted Fusion method** has multiple classifiers. Each classifier is trained in a different SNR or RT60 level environment. Using these classifiers and information about the environment (e.g. estimate of SNR level or RT60 level), a weighted fusion method is developed to weight and combine the classifiers decisions. The weighting of the Multi-Engine is done by the confidence of the characterization of the surrounding environment. The introduced system is capable of handling variable environments. By applying this method in the talker identification (TI) system, we have shown that it has overall classification accuracy of 91.5% in noisy environments over an SNR range clean environment to 0 dB. Compared to conventional methods the improvement in accuracy was as high as 48.5%. Also, we have shown that it has overall classification accuracy of 96.5% in reverberant environments over an RT60 range of clean environment to 1.80 sec. Compared to conventional methods the improvement in accuracy was as high as 10%.

This work is published in:

- i. A.R. Abu-El-Quran, R.A. Goubran, and A.D.C. Chan, "Security Instrument using Talker Identification and Microphone Arrays in Variable Noisy Environments," in *IEEE proc. International Instrumentation, and Measurement Technology Conference*, Victoria, BC, Canada, May. 2008, pp. 1046-1050.
- ii. A.R. Abu-El-Quran, J.S. Gammal, R.A. Goubran, and A.D.C. Chan, "Talker Identification Using Reverberation Sensing System," in *IEEE Proc., International Conference on Sensors*, Atlanta, Georgia, USA, Oct. 2007, pp. 970-973.
- iii. A.R. Abu-El-Quran, A.D.C. Chan, and R.A. Goubran, "Security Instrument using Multi-Engine Talker Identification and Microphone Arrays in Variable Noisy Environments," *submitted to IEEE Trans. Instrumentation and Measurement*, 2008.

1.4.2 Minor Contributions

1. **Introduce a generalized adaptive feature threshold** for ASC using an interface between the environment variable (e.g. SNR or RT60) and the audio classification threshold. This interface estimates the threshold of the audio feature as a function of the environment variable then adapts the threshold to enhance the classification accuracy performance. This method is implemented on features for speech/non-speech classification and speech/music classification. We have shown that by applying this generalized adaptation on standalone speech/non-speech classification

feature, this feature achieved classification accuracy of 85.0% at SNR = -5 dB. Compared to the non-adaptive feature the improvement in accuracy was as high as 33.0%. Also we have shown that by applying this generalized adaptation on standalone speech/music classification feature, this feature achieved classification accuracy of 90.0% at SNR = 0 dB. Compared to the non-adaptive feature the improvement in accuracy was as high as 40.0%.

This work is published in:

A.R. Abu-El-Quran, R.A. Goubran, and A.D.C. Chan, "Audio Classification Using Microphone Arrays," *Ontario Centers for Excellence*, Toronto, Canada, 2006. Competition WINNER poster from OCE.

2. **Introduce a novel Reverberation Sensing System (RSS)** that determines the approximate reverberation level (RT60) of the surrounding environment. This system is inspired by the work done in [16]. This system provides information about the reverberant surrounding environment. This information is used in this work to adapt and enhance the classification accuracy of the ASC system.

This work is published in:

A.R. Abu-El-Quran, J.S. Gammal, R.A. Goubran, and A.D.C. Chan, "Talker Identification Using Reverberation Sensing System," in *IEEE Proc.*

International Conference on Sensors, Atlanta, Georgia, USA, Oct. 2007, pp. 970-973.

3. **Architecture of a security system** that can detect and classify the location using a microphone array and nature of different sounds within a room using ASC. This system is reliable and robust, even in the presence of reverberation and in low signal-to-noise environments. This system is a hierarchical ASC system. First, it classifies an audio signal as speech or non-speech. Second, it classifies non-speech audio signal into a particular audio class (e.g. footsteps, windows breaking) and performs talker identification for speech audio signal. This system achieved average classification accuracy of 87.0% at SNR = 20. Compared to the same system without the utilization of the environment information the improvement in accuracy was as high as 57.0%.

This work is published in:

- i. A.R. Abu-El-Quran, R.A. Goubran, and A.D.C. Chan, "Security Instrument using Talker Identification and Microphone Arrays in Variable Noisy Environments," in *IEEE proc. International Instrumentation, and Measurement Technology Conference*, Victoria, BC, Canada, May. 2008, pp. 1046-1050.
 - ii. A.R. Abu-El-Quran, R.A. Goubran, and A.D.C. Chan, "Security-Monitoring Using Microphone Arrays and Audio Classification," *IEEE*
-

Trans. Instrumentation and Measurement, vol. 55, no. 4, pp.1025-1032, 2006.

4. **Creation of new database of audio sounds** contains speech and non-speech audio sounds. The significance of this database comes from its wide range of audio classes such as windows breaking, wind, traffic, and cocktail noise, which relate to many ASC applications.

This work is published in:

A.R. Abu-El-Quran, and R.A. Goubran, "Security-Monitoring using Microphone Arrays and Audio Classification," in *IEEE proc. Instrumentation, and Measurement Technology Conference*, Ottawa, ON, Canada, May. 2005, pp.1144-1148.

1.5 Thesis Outline

The thesis is organized into 9 chapters. The chapters are organized as follows:

Chapter 1 presents a general introduction to the ASC and describes the problem that this thesis addresses. This chapter contains a statement of the objectives of this thesis. A summary of the contributions of this thesis is also presented.

Chapter 2 contains an overview of ASC and previous works in this area. A detailed description is given of the main stages of the ASC. A literature review of the ASC categories performance in the variable environments is also presented in this chapter.

Chapter 3 presents the impact of variable environments on ASC. The performance degradation of ASC systems due to the variation of the surrounding environment is discussed. Examples of the ASC categories such as speech/non-speech and talker identification were simulated to show the effect of the reverberant and noisy environments variation on the performance of the ASC.

Chapter 4 presents the experimental setup of the ASC systems. Included in this chapter is a description of the models used to simulate the reverberation and the noise that have been used to simulate the surrounding environments. A detailed description of the audio database is also presented in this chapter. Finally, the performance evaluation criteria are also discussed in this chapter.

Chapter 5 presents a novel framework to solve the performance degradation of ASC in variable environments. This framework involves a system that adapts to its environment. Several methods are introduced within the framework; each method is introduced and discussed. The abstract implementation of these methods is also presented in this chapter.

Chapter 6 presents novel adaptive ASC features; APR and VLER, which have a resilient performance in variable and harsh environments (low SNR and high reverberation). A new generalized adaptation method for the ASC features is also developed. This chapter introduces Multi-Feature Selection Multi-Feature Combination methods. These methods select or combine the weighted decision

of each feature using the environment information to enhance the ASC performance.

Chapter 7 presents methods to estimate environment variables, including SNR and reverberation (RT60). Using the estimated environment variable, a Multi-Engine ASC system is developed that has increased accuracy and robustness in variable environments. Two methods have been developed to perform Multi-Engine ASC, which were evaluated in variable reverberant environments. The first method is the Multi-Engine Selection method, which selects the classifier with the reverberation training condition that best matches the surrounding environment. The second method is the Multi-Engine Weighted Fusion method, which weights the output decisions of the ASC classifiers, using the environment estimator output, and combines these weighted decisions to perform the ASC.

Chapter 8 presents a novel security instrument using ASC. This instrument estimates the environment variable to adapt the speech/non-speech classifier to the surrounding environment. If the detected signal is speech, a talker identification system using the Multi-Engine methods will identify the talker; otherwise, an audio classification system is employed to determine the audio class of the detected signal (e.g. widows breaking, and wind sounds).

Chapter 9 is devoted to conclusions and future work

Chapter 2

Overview of Audio Signal Classification

This chapter contains an overview of audio signal classification (ASC) and previous works in this area. A detailed description is given of the main stages of the ASC. Also a literature review of the ASC categories performance in variable reverberant and noisy environments is presented in this chapter.

2.1 History of ASC

ASC has caught the attention of researchers in the past few years [1], [6], [17]. As the demand grows for multimedia [4], [5], biomedical [1], [2], and security applications [3], the need for ASC with high performance becomes more apparent.

Alexander Graham Bell invented the microphone and, in 1876, his "electrical speech machine," known now as the telephone. By 1878, Bell had finished the first telephone exchange in New Haven. From that time, audio became an important research area [18].

Bell Laboratories introduced one of the ASC categories, automatic speech recognition (ASR), in 1952. In 1959, Bell Laboratories introduced Time Assignment Speech

Interpolation (TASI). The TASI system used another category of the ASC systems, namely voice activity detection (VAD) [18], to pause the transmission of the carrier signal when there is no speech. The TASI system increased the telephone circuit capacity of communication channels.

In these days ASC categories have been deployed in many applications. The revolution of information technology has created the need to perform audio retrieval and indexing of the massive multimedia storage [19], [20]. Modern applications such as biomedical instruments, audio-video conferencing, and security application will have automation capability if it is combined with robust ASC systems. For example, hearing aids can use ASC to optimise their response according to the surrounding environment [1]. Audio-video conferencing may use ASC to automatically steer the camera appropriately to the talker, and avoid steering the camera to an undesired sound source (e.g. fan noise). Security applications can use ASC to identify sounds (e.g. windows breaking, footsteps) and respond to these sounds appropriately (e.g. alerts, enabling video) [3].

2.2 Overview of ASC Process

ASC is defined as the automatic categorization of an audio signal into a certain class (e.g. speech, music, environment sounds).

To develop an ASC system, a categorization for the audio signal needs to be done first. Many categorizations have been introduced to group the similar audio signals into main audio classes and subclasses [21]. Examples of audio signal classes are as follows:

- **Environmental sounds** are defined as sounds that come from nature, such as rain, wind and bird sounds. Also, sounds such as traffic and a door closing are assumed to be environmental sounds [22], [23].
 - **Music** is defined as a form of auditory communication using instrumental or vocal tones. In this thesis, the music produced from man-made instruments is only considered as music. Birdsongs are considered to be environmental sound, not music [24].
 - **Noise** sound is defined as random or pseudorandom sound signals that can be classified using the distribution of the energy in the spectrum of the signal. White noise contains a uniform energy spectrum distribution, while colored noise contains a non-uniform energy spectrum distribution, such as pink and brown noises [25], [26]. Cocktail noise is defined as a number of sound sources with almost the same energy level produced at the same time. These sources may be a speech or environmental sound sources. Cocktail noise is produced by a large number of simultaneous talkers [27].
 - **Speech** is the sound produced from a human mouth due to the movement of the vocal tract as a response to an excitation signal. The speech sound may be periodic at the glottis which is called voiced speech or noise-like due to the deformation in the vocal tract which is called unvoiced speech [25], [28] (Figure 2.2.1).
-

Figure 2.2.1 Speech waveform of voiced (*a*) and unvoiced (*sh*) types

The ASC process is conducted using three main stages: audio capturing, feature extraction and classification.

The first stage is audio capturing; that is, to capture and digitize the input audio signal. The second stage is feature extraction, which performs data reduction techniques to create feature vectors. These features should be chosen such that they are similar for audio from the same class and dissimilar for audio from different classes, enabling discernment of audio classes. Finally, in the third stage, the feature vector is passed to a classifier to determine the most likely class for the input audio signal (Figure 2.2.2). In the following subsections the different stages of the ASC process will be discussed.

Figure 2.2.2 Audio signal classification process

2.2.1 Audio Capturing Devices

Audio capturing devices, known as microphones, are transducers which detect audio signals and produce an electrical signal as a voltage or a current which is proportional to the amplitude of the audio signal [29].

Microphones are described with different directional patterns. The directional pattern is simply the arc which describes the sensitivity of the microphone in different directions as shown in Figure 2.2.3.

Figure 2.2.3 Audio capturing device Arc.

There are two main types of microphones from a spatial discrimination point of view: omnidirectional and unidirectional devices. Omnidirectional microphones do not have spatial discrimination capabilities as they capture sound from all directions with the same gain. Unidirectional microphones have spatial discrimination capability (Figure 2.2.4).

Figure 2.2.4 Unidirectional audio capturing device (Mitel®)

A unidirectional microphone can be implemented using a group of omnidirectional microphones; this grouping is known as microphone array. Combining the signals from these microphones to form a synthetic beam in a certain direction is called beamforming [30]. Beamforming is desirable to create a microphone with the ability to focus on a specific audio source, meanwhile avoiding the interfering audio signals from an undesired location. The beamformer beam can be abruptly steered to different locations electronically. A simple algorithm to create a beamform using microphone array is a delay-and-sum beamformer (Appendix A).

2.2.2 Feature Extraction

The digitized audio signal has a large amount of information which needs large memory and processing power. Feature extraction reduces the amount of information; meanwhile, it retains the information necessary for ASC.

Many audio features are introduced and described in the literature. Each of these features is appropriate for a certain ASC category. Also, these features have different

performance in different reverberant and noisy environments. The main features utilized in the ASC are listed in Table 2.2.1 and are described in the following sections.

Table 2.2.1 Main audio features

Features
Energy
Zero Crossing Rate
Periodicity
Mel Frequency Cepstrum Coefficient
4 Hz Modulation
Entropy
Segmentation count
Spectral Centroid

2.2.2.1 Energy

Short time average energy E_n provides the representation of the amplitude variation over a period of time (Figure 2.2.5) [31].

Figure 2.2.5 Short time average energy

Short time average energy E_n is computed as follows:

$$E_n = \frac{1}{N} \sum_{m=1}^N [x(m)w(n-m)]^2 \quad (2.2.1)$$

where $x(m)$ is the discrete time audio signal, n is the time index and $w(n)$ is a window function (e.g. Hamming, rectangular) with length N [31], [32].

Many categories of the ASC are implemented using the short time energy feature such as VAD; generally voiced speech has a higher E_n than unvoiced speech components. Also, it can be used to detect the silence period at high SNR. In [33], a speech/non-speech classification decision is supported by the fact that the energy of the speech signal should change smoothly, without abrupt changes while segments with large energy variation are assumed to be non-speech. In [34], a footsteps detector is constructed using the energy distribution of the vibration generated by human activities; human activity has Rayleigh distribution [34].

2.2.2.2 Zero Crossing Rate

Zero crossing rate (ZCR) is the number of time-domain zero-crossings within a certain temporal signal frame. ZCR is utilized to measure the dominant frequency in an audio signal [35]. As shown in Figure 2.2.6, ZCR is said to occur if successive samples with different signs are observed [31]. Short time average ZCR is defined as follows:

$$Z_n = \frac{1}{2} \sum_{m=-\infty}^N [\text{sgn}[x(m)] - \text{sgn}[x(m-1)]] w(n-m) \quad (2.2.2)$$

$$\text{sgn}[x(i)] = \begin{cases} 1 & x(i) \geq 0 \\ -1 & x(i) < 0 \end{cases} \quad (2.2.3)$$

where $x(m)$ is the discrete time audio signal, n is the time index and $w(n)$ is a windowing function with length N [31].

Figure 2.2.6 Zero crossing rate of audio signal

ZCR is a time-domain feature with real-time calculation compatibility. ZCR provides information about the spectrum characteristics without calculating the spectrum itself. The major drawback of the ZCR is the sensitivity to the background noise, which can be partially overcome by using a modified ZCR. This modification is done by replacing the zero threshold at equation (2.2.3) with a fractional threshold of the root mean squared energy (RMSE) to avoid change of the small signal sign (relative to the RMSE) in the presence of noise (i.e. hysteresis). This can be shown as follows:

$$\text{sgn}[x(i)] = \begin{cases} 1 & x(i) \geq \alpha \times RMS, & 0 \leq t \leq 1 \\ \text{sgn}[x(i-1)] & -\alpha \times RMS < x(i) < \alpha \times RMS & 0 \leq t \leq 1 \\ -1 & x(i) \leq -\alpha \times RMS, & 0 \leq t \leq 1 \end{cases} \quad (2.2.4)$$

where α is a positive fraction threshold.

ZCR can be utilized as a voiced/unvoiced/silent standalone classifier at high SNR; the unvoiced part has a higher value of ZCR [36]-[38]. In [4], a speech/music classifier, used in the radio broadcast, utilized the variance of ZCR contour to perform speech/music classification, as speech has a higher ZCR variance.

2.2.2.3 Periodicity (Pitch Detection)

Periodicity is the value of the fundamental frequency of periodic or pseudo-periodic signals (Figure 2.2.7).

Figure 2.2.7 Pitch detection of audio signal

In Figure 2.2.7 the initial pitch value is equal to zero because the algorithm requires at least one period to predict the pitch value. The pitch can be determined by detecting similarities in the signal waveform [39]-[44]. Pitch detection is utilized in ASC categories such as VAD [40] and speech/non-speech classification [45].

2.2.2.4 Mel-Frequency Cepstrum Coefficients (MFCC)

Mel-frequency cepstrum coefficient (MFCC) is fundamentally simulating the human auditory model which can fit in many ASC categories. The main steps to calculate the MFCC are shown in Figure 2.2.8 [46].

Figure 2.2.8 MFCC calculation steps

As shown in Figure 2.2.8 a pre-emphasis filter is used in the speech application where the spectrum decays with the frequency. The pre-emphasis filter produces a flat top spectrum [46], [47]. A human auditory model is simulated using a mel-frequency bank which composed of a set of K overlapping triangular filters (Figure 2.2.9). Summation of the power spectrum within the same triangle filter is then carried on to produce $\{X_k, 1 \leq k \leq K\}$.

Figure 2.2.9 Nonlinear Mel-scale filters (10 filters designed for 8 KHz audio signal)

The centers of the triangular filters in the mel filter bank are linear on the mel frequency scale [48]. The mel transformation is done as follows:

$$\text{Mel}(f) = 2595 \log_{10} \left(1 + \frac{f}{700} \right) \quad (2.2.5)$$

where the filters centers are computed as follows:

$$fc_i = 700 * \left[10^{\frac{(mel f_{min} + \frac{(mel f_{max} - mel f_{min})i}{K+1})}{2595}} - 1 \right], i = 1 \dots K \quad (2.2.6)$$

where $mel f_{min}$, $mel f_{max}$ are minimum and maximum frequencies on the mel scale.

DCT compresses and de-correlates the feature vector of the filter bank outputs X_k [48].

$$MFCC_i = \sum_{k=1}^K \log(X_k) \cos \left[i \left(k - \frac{1}{2} \right) \frac{\pi}{K} \right], \quad i = 1, 2, \dots, L \quad (2.2.7)$$

where L is the length of the resulting MFCC vector.

MFCC is a commonly used with speech ASC sub-categories such as talker identification (TI) and automatic speech recognition (ASR) [49].

2.2.2.5 4 Hz Modulation

4 Hz modulation is defined as the modulation energy, which can be detected in the speech signal around 4 Hz; 4 Hz is used because speech has a syllabic rate that is generally near 4 Hz [31]. 4 Hz modulation is calculated as shown in Figure 2.2.10.

Figure 2.2.10 4 Hz Modulation Algorithm

First, the audio segment is cut into smaller frames. Mel frequency spectrum coefficients (MFSC) are calculated. MFSC is calculated using the same procedure as MFCC except that it does not calculate the DCT. The sum of the filtered version of the energy of the perceptual channels (Triangle filters) using 4 Hz band-pass filter is divided by the sum of the energy of the perceptual channels. The variance of the resulting vector is the 4 Hz modulation [50].

The 4 Hz modulation feature is used to isolate the speech from other audio classes; speech has higher 4 Hz modulation value than other audio class. Speech/music classification is one of the ASC categories that can use 4 Hz modulation [50], [5]. The major disadvantage of this feature is that its classification performance is degraded dramatically at low SNR (SNR < 20 dB).

2.2.2.6 Entropy

Entropy is a time domain statistical measure of the randomness of the signal. The value of the entropy increases as the randomness increases. Entropy is utilized in speech/music classification; speech has more randomness than music [50]. Figure 2.2.11 shows the main steps to calculate the entropy. Entropy is calculated as follows:

$$H = \sum_{i=1}^k -p_i \log_2 p_i, \quad p_i = \text{Propability of event } i. \quad (2.2.8)$$

Figure 2.2.11 Entropy calculation

2.2.2.7 Segmentation Count

Segmentation count is defined as the number of the statistical segments in a certain temporal audio signal. In [51], segmentation count is implemented by calculating the statistical distance between a small frame (smaller than the total signal duration) and the audio signal duration. This distance is then compared with a certain threshold. If the distance is larger than this threshold, a new segment index will be set and another frame will be grabbed to detect the next segment.

As shown in Figure 2.2.12 and Figure 2.2.13, a speech signal is composed of alternate periods of transient and steady parts (steady parts are mainly vowels); meanwhile, audio signal such as music, glass breaking, and wind sounds are steadier and have smaller number of segments.

Figure 2.2.12 Music segmentation

Figure 2.2.13 Speech segmentation

2.2.2.8 Spectral Centroid

Spectral centroid is defined as the balancing point of the spectral power distribution. Centroid of a spectral frame is calculated by dividing its average frequency components, weighted by its amplitude, by the sum of its amplitudes as follows [31]:

$$c = \frac{\sum_{i=1}^L f_i A_i}{\sum_{i=1}^L A_i} \quad (2.2.9)$$

where f_i is the frequency and A_i is the amplitude of the i^{th} spectral component, and L is the spectral length. Centroid is utilized in speech/music classification; the music signal has higher centroid value than the speech signal [5].

2.2.3 Classifiers

Several classifiers exist such as linear classifiers such as principle of component analysis (PCA), and non-linear classifier such as Gaussian mixture models (GMM), neural networks (NN), and hidden Markov model (HMM). These classifiers use feature vectors extracted from captured audio signals to create models for audio classes.

2.2.3.1 Linear Classifiers

The discernment function of the linear classifier is a linear combination of the feature vector components [52]. A simplified linear classifier for two dimensional features is shown in Figure 2.2.14. This classifier is commonly used in the coarse ASC categories when there are only two audio classes to be classified. Linear classifier can also be utilized in a multi-class classification. Examples of the linear classifier is PR algorithm [45] and principle component analysis (PCA) [53].

Figure 2.2.14 Linear classifier (two classes and two features)

2.2.3.2 Non-linear Classifiers

GMM is used to model the extracted feature vector. A multimodal Gaussian probability density function (PDF) (Figure 2.2.15).

Figure 2.2.15. Clustering of a single audio class feature vectors

In Figure 2.2.15 a single Gaussian in the mixture model is illustrated by a single cluster (ellipse).

Assume that the number of Gaussians is G and the number of elements in the feature vector is L . Each audio class s is modeled by a mixture of G Gaussians during the training mode of the ASC. Each of the Gaussians g is characterized by its covariance, mean and weight. Each audio class s has a mixture with G mean vectors with length equal to L , G weights which are a single scalar weight value for each mixture; the mixture weights have a sum of 1 for a single audio class, and G covariance which is $L \times L$ matrix [54]. Each audio class C_s is modeled by a GMM λ_s :

$$\lambda_s = \{\vec{\mu}_g, p_g, \Sigma_g\} \quad g = 1, \dots, G \quad (2.2.10)$$

where G is the number of Gaussians, $\vec{\mu}_g$, p_g and Σ_g are mean vector, weight and covariance matrix of Gaussian g in the mixture of audio class C_s respectively. The

probability of an unknown audio class feature vector $\vec{y}$ belonging to class C_s is calculated using all the mixtures in the model as follows [55]:

$$p(\vec{y} | \lambda_s) = \sum_{g=1}^G p_g b_g(\vec{y}) \quad (2.2.11)$$

$$b_g(\vec{y}) = \frac{1}{(2\pi)^{L/2} |\Sigma_g|^{1/2}} \exp\left\{-\frac{1}{2}(\vec{y} - \vec{\mu}_g)' \Sigma_g^{-1} (\vec{y} - \vec{\mu}_g)\right\} \quad (2.2.12)$$

where $|\bullet|$ is the determinant.

The parameters of each audio class are determined using M training vectors $X, \{\vec{x}_i \in X, 1 \leq i \leq M\}$. The clustering algorithm, such as the Linde, Buzo, Gray (LBG), clusters the training vectors [56]. For each cluster the mean and the variance are then calculated. The density of each audio class in each cluster is computed as the ratio of the number of feature vectors of this audio class in this cluster to the total number of feature vectors of this audio class. Using these clusters and an iterative algorithm such as Expectation Maximization (EM) algorithm, a new model is created after each iteration [55], [57]. The iterative algorithm optimizes $\vec{\mu}_g, p_g$ and Σ_g which guarantees that $P(X | \lambda^{r+1}) \geq P(X | \lambda^r)$; λ^r represents the model that produced after iteration r [57].

In testing mode, N feature vectors $Y, \{\vec{y}_j \in Y, 1 < j < N\}$ are extracted from an unknown audio signal. Assuming that the test feature vectors of the audio signals are independent, the probability that the unknown audio signal $h \in C_s$ is calculated as follows [55]:

$$P(Y | \lambda_s) = \prod_{j=1}^N p(\tilde{y}_j | \lambda_s) \quad (2.2.13)$$

The log likelihood of the previous equation is calculated as follows [55]:

$$\log P(Y | \lambda_s) = \frac{1}{N} \sum_{j=1}^N \log p(\tilde{y}_j | \lambda_s) \quad (2.2.14)$$

For S audio classes, the most likely audio class s^* for the unknown audio signal Y is calculated as follows [55]:

$$s^* = \underset{1 \leq s \leq S}{\arg}(\max(\log P(Y | \lambda_s))) \quad (2.2.15)$$

Neural networks (NN) are commonly used for ASC categories such as talker recognition [58] and audio class classification [27]. Many types of NN have been introduced in the literature [58], [59]. Multi layer perceptron (MLP) is a very common NN structure. This structure is constructed from input, hidden and output layers. Each layer contains a number of nodes. The number of nodes in the input layer is equal to the length of the feature vector L . As reported in [60], a good rule of thumb to choose the number of the hidden nodes (the nodes of the hidden layer) is to use $2L-1$. The number of the nodes in the output layer is equal to or less than the number of the audio classes S . During training, the back propagation (BP) update algorithm is used to update the weights of the interconnection between the nodes. During testing, the feed forward (FF) algorithm is utilized. At each node of the hidden layer and the output layer, the weighted input to this node is summed. Figure 2.2.16 is an example of a MLP with 2 inputs, 1

hidden layer of 3 nodes and one output node. A nonlinear activation function is performed on this sum. Examples of non-linear activation functions that can be used in the hidden and the output layer are hyperbolic tangent sigmoid function and log-sigmoid function.

Figure 2.2.16. Multi layer perceptron neural network structure

2.3 ASC Categories

ASC categories are potentially far-reaching and relevant research areas. Examples of ASC categories are audio indexing and retrieval, music genre classification, speech/non-speech classification, and talker identification. ASC categories can be represented as a standalone application such as ASR systems or a combined application such as ASR systems with prosodic¹ detector.

As mentioned in Chapter 1, ASC categories can be classified into two main levels: coarse level and fine level. The coarse level mainly performs binary classification (two audio classes) such as speech/non-speech classification. Fine level performs further audio

¹ Prosody is the changes in pitch and loudness of speech that convey information, like rising pitch at the end of a question.

classification after the main audio class is determined such as performing talker identification after determining that the input audio signal is speech from the coarse level [61].

2.3.1 Coarse ASC

In the following subsections, examples of coarse ASC level categories and their applications are provided; more specifically, speech/non-speech classification and speech/music classification.

2.3.1.1 Speech/Non-Speech Classification

Several approaches have been suggested in the literature to perform speech/non-speech classification [21], [33], [62]. The Silence Ratio approach was investigated in [21]. The silence period is defined as the idle time without any speech activity. In this approach the authors claim that speech has longer silence period duration than other audio classes such as music. The classification accuracy using these Silence Ratio approach was 87.0% [21].

In [33], speech/non-speech classification is achieved using three thresholds. The first threshold is the energy of the audio signal, where at speech signal the energy changes smoothly. The second threshold is speech duration; English words are usually in order of 70 ms [33]. The third threshold is the change of the pitch value of speech which should change slower than other audio class. Using 400 ms audio segments, the classification accuracy using these thresholds was 89.0% [33].

In [62], speech parts have been isolated from other audio classes in the broadcasting stream using variance of spectrum flux. Phoneme transitions (e.g. vowels to consonants) create transition periods between periodic and non-periodic speech parts. This leads to the fact that the variance of the spectrum flux of the speech audio class has larger value than other audio classes. The classification accuracy using the variance of spectrum flux was 91.5% [62].

2.3.1.2 Speech/Music Classification

Speech/music classification becomes progressively more important in many applications [21],[35],[63]-[65]. Content-based audio retrieval utilizes speech/music classification in hierarchical classification systems [21],[35]. Multi-modal coders employ a speech/music classifier to switch between speech and audio coders, increasing the overall codec efficiency [66]. Music genre classification utilizes speech/music classification to classify the audio signal as music prior to music genre classification [63]. Also, audio signals need to be classified as speech prior to automatic speech recognition (ASR), where ASR is disabled if non-speech sound is detected (e.g. news broadcast transcription) [65].

Several approaches have been suggested in the literature to perform speech/music classification [5], [50], [67]. In [5], eight features were used for speech/music classification: 4 Hz modulation, low energy frames, spectral rolloff, spectral centroid, spectral flux, zero crossing rate, cepstral residual, and pulse metric. Using one-second

audio segments, the classification accuracy of these features ranged from 78.0% to 94.0% [5].

In [50], four features have been used with GMM and fusion classifiers: entropy modulation, 4 Hz modulation, number of segments, and segment duration. Using one-second audio segments, the classification accuracy using multiple features is 93.9% for speech and 89.9% for music.

In [67], four features and their delta (first difference) were examined: cepstral coefficients, amplitude, pitch, and zero crossing rate. A GMM was used as the classifier. The library used in [67] contains speech samples from thirteen different languages and different music types. The classification accuracy ranged from 80.0% to 98.0% using a 64 GMM.

2.3.2 Fine ASC

After the audio signal is classified into a main audio class using the coarse ASC level, further classification may be required to bring the classification decision to fine ASC level. For example, ASC category such as talker identification (TI) first requires coarse ASC to classify the audio signal into speech, and then requires fine ASC to identify the talker. In this work, the only fine level we examine is TI.

2.3.2.1 Talker Identification

Talker identification is defined as selecting the talker who uttered a speech signal from a set of known talkers [68]. The majority of (TI) algorithms use MFCC vectors and

the linear predictions cepstral coefficients (LPCC) [11], [68]-[70]. Cepstral transformation, using the discrete cosine transform (DCT), de-correlates the vector elements to enhance the capability of measuring the distance between different vectors; de-correlated elements enhance the classification boundaries [11].

2.3.3 Combined ASC

ASC categories can be implemented as standalone fine or coarse levels. However some of the ASC categories require a combined ASC level such as audio indexing and retrieval. In these combined ASC categories, the information provided from the coarse level is valuable, as well as the fine ASC information (e.g. the index is “Music, Rock”).

2.3.3.1 Audio Indexing and Retrieval

Rapid increase in the amount of digital audio clips characterized by various formats, types, durations, and other parameters enforced the demand of a framework for efficient audio indexing and retrieval techniques. An audio collection needs to be managed to allow an efficient access, search, and browsing capabilities for the desired audio clip. For example, a user may want to find an audio clip that sounds similar to an audio sample in hand. In this case, to retrieve the desired clip, the audio data in the database have to be matched with the sound of interest, and then return the clip with similar features to the user.

Query by example technique is a global approach, which is adopted by several research studies and implementations [6],[20],[71]. A popular audio retrieval system

known as Muscle-Fish [71], [72] retrieves the audio clips using several extracted audio features. In this approach, a feature vector is constructed using a number of audio features such as bandwidth, pitch, and amplitude. These features are used for similarity search for the query sound. In [20], a 13-dimensional feature vector is extracted (12 MFCC and spectral energy) to be used in a tree-based Vector Quantizer (VQ) to create a template for each audio class.

2.4 ASC in Variable Reverberant and Noisy Environments

In the previous Section 2.3 it has been shown that many ASC categories and applications have been developed with high accuracy performance; however very little research has been conducted to study the effect of variable reverberant and noisy environments on the ASC performance. ASC in variable reverberant and noisy environments is essential in many applications. An example of these applications is cellular phones, where users use certain words to initialize certain phone functions. Users of cellular phones suffer from a dysfunction of the word recognition engines. The main reason of this dysfunction is that the engine fails to recognize the word. This failure happens if the test environment is different than the training environment or even if the microphone is at different position than the training position from the reference point (mouth of the talker). The microphone position changes the reverberation and the SNR pattern from the microphone perspective which is effectively a change in the surrounding environment.

2.4.1 ASC in Reverberant Environment

Reverberation has a high degradation impact on the performance of the ASC (impact of the reverberation on the performance of the ASC is discussed in Chapter 4); however very little research has been conducted to study and solve this degradation impact on the performance. ASC in reverberant environments has been addressed in references [11], [13], [16].

In [11], 51 talkers were used to conduct a closed-set text-independent TI task. The ASC process is defined as a closed-set if the set of known audio is limited and the unknown audio (test) is one of the audio classes in the known set (training). The ASC is defined as an open-set if there is as likelihood that the audio signal is not included in the known audio classes set. The talkers in [11] were tested in five different reverberant environments, which were created using the image method [73] (image method is discussed in section 3.2.1.1). In [11], it was shown that the performance can be enhanced if the classification engine is trained using reverberated speech rather than clean speech even if the training and testing room are different. This reference used 16 linear prediction cepstral coefficients (LPCCs) as a feature vector and used 32-GMM as classifier to perform the TI. This result confirmed that the performance of the TI depends on the similarity of the training and testing environment.

In [13], the effect of reverberation on eight talker closed-set text-independent TI task was investigated. NN was utilized as the classifier. The extracted feature vectors were MFCC, line spectrum pairs (LSP), and reflection coefficients. The reverberation was

simulated using the image method [73]. The authors used different environment setups to test the performance degradation of a TI system. The important observations were:

- Performance depends on the similarity of the environments of the training and test speech.
- Distance between the microphone and the talker does not have a major impact on the performance as long as the positions of microphone and talker are constant in training and testing.
- Reverberation time (RT60) is not the only factor that can affect the performance but also the reverberation density (higher number of signal replicas). For example, when comparing two rooms with different volumes having the same RT60, the room with the smaller volume will result in lower performance due to the greater reverberation density.

In [16], room identification using speech signal was introduced (Figure 2.4.1). Six reverberated rooms were simulated using the image method [73]. A room identification engine was implemented using second order autoregressive vector method (AR-AGS). In [16], the training and test environments set used were the same. This room identification was used as reverberation identification. The classification accuracy of the rooms in this paper achieved 96.5%. The introduced algorithm in [16] is not generalizable, as it is limited to a certain number of rooms (training and test environment are the same).

Figure 2.4.1 Room Identification System

2.4.2 ASC in Noisy Environment

ASC in noisy environments has been addressed in references [6], [12], [17], [38], and [14]. In [6], ASC was utilized to select different modes of operations in modern hearing aids (Figure 2.4.2). The selection of these modes depends on the detected audio class (e.g. music, speech and noise). In [6], the classification is conducted in different surrounding environments (e.g. car, office, mixture of babble, printers). The feature used for the classification is the modeling of the auditory system. This feature is applied on a GMM classifier. Using a one second audio segment this algorithm achieved coarse ASC level (speech/music/noise) accuracy of 88.0% [6]. In [6], the authors did not mention the value of the SNR, used in the experiments, which makes an evaluation of the performance of the system impossible. Also, the authors did not discuss the satisfaction of the hearing aid users after applying the algorithm on the hearing aid device.

Figure 2.4.2 Mode selection in hearing aids device

In [12], the effect of five different speech enhancement techniques on GMM based automatic speech recognition (ASR) performance was evaluated. These enhancements techniques are: combined two-pass spectral and perceptual wavelet adaptive denoising (PWAD) (TPS-PWAD), VAD based noise floor estimation (VAD-NFE), minimum mean

square error based two-dimensional spectral enhancement (MMSE-TDSE), PWAD, and two-pass adaptive noise estimation (TPANE). In [12], the speech is enhanced without relying on *a priori* information of the environment. First, the noisy speech was denoised to improve the SNR then the effect of this improvement on the ASR classification accuracy was investigated.

The performance of the speech enhancement techniques was evaluated using Perceptual Evaluation of Speech Quality-Mean Opinion Score (PESQ-MOS) [74]. PESQ-MOS predicts the subjective quality of input speech by returning a score between 0 and 4.5, the higher the score the cleaner the input speech (i.e. higher quality). These techniques were used as preprocessing blocks to the ASR engine. The overall performance was evaluated based on the ASR classification accuracy using each enhancement technique.

In [12], TIMIT speech database was used to evaluate the performance of the ASR using the five enhancement techniques. The performance of the ASR using the five enhancements techniques is shown and compared in Figure 2.4.3 - Figure 2.4.4 and Table 2.4.1. These figures and table are taken from reference [12]. Two noise types are discussed in this section: white and babble noise. For more noise types discussion please refer to [12].

Figure 2.4.3 ASR classification accuracy with White noise

Figure 2.4.4 ASR classification accuracy with Babble noise

Table 2.4.1 ASR classification accuracy improvement with White, and Babble noise

	Algorithm	20 dB	15 dB	10 dB	5 dB	0 dB
White	Two-Pass + Wavelet	9.29%	15.72%	9.29%	12.86%	7.86%
	VAD Noise Estimation	6.72%	12.14%	8.57%	13.57%	5.00%
	MMSE-TDSE	-0.71%	16.43%	7.86%	7.14%	2.86%
	Perceptual Wavelet	8.57%	18.57%	14.29%	10.71%	-0.71%
	Two-Pass Spectral	3.57%	10.00%	9.29%	14.29%	5.71%
Babble	Two-Pass + Wavelet	-7.86%	-15.72%	-14.57%	-5.00%	6.43%
	VAD Noise Estimation	-3.57%	-4.29%	-4.57%	6.43%	13.57%
	MMSE-TDSE	-29.29%	-32.86%	-30.29%	-14.29%	2.14%
	Perceptual Wavelet	0.00%	0.71%	0.43%	3.57%	3.57%
	Two-Pass Spectral	-5.00%	-3.57%	-8.86%	2.14%	11.43%

From Figure 2.4.3, the speech enhancement techniques of the corrupted speech signal with white noise, generally improved the classification accuracy of the ASR. This improvement is inconsistent for different SNR values for a single enhancement technique. As example using the PWAD, as a preprocessing to the ASR, achieves the best ASR classification accuracy for $20 \geq \text{SNR} > 5$ dB, while the TPS-PWAD achieves the best ASR classification accuracy for $5 \geq \text{SNR} \geq 0$ dB.

The author of [12] noticed that MMSE-TDSE has the greatest average improvement of the speech quality PESQ-MOS scores over SNR range from 0 to 20 dB; improvement of 0.7 points [12]. This improvement did not translate into the best ASR classification accuracy. Also MMSE-TDSE shows a negative improvement of -0.71%. This negative improvement means classification accuracy degradation compared to the classification accuracy of ASR without speech enhancement techniques. Also, it was noticed that VAD-NFE achieved the best SNR improvement of 0.5 dB and 3.7 dB at 20 and 15 dB, respectively [12]; however, it did not translate into the best ASR classification accuracy

at these SNR values. These notices prove that the speech quality enhancement and the SNR improvement are uncorrelated with the ASR classification accuracy.

Table 2.4.1 shows that the speech enhancement techniques of a speech corrupted with babble noise degraded the ASR classification accuracy with average of 5.2%; compared to the classification accuracy of ASR without speech enhancement techniques. In [12], it has been proved that using the five speech enhancement techniques improved the PESQ-MOS score and the SNR value for the speech signal corrupted with babble noise. These improvements improved the speech audibility from the human perspective and introduced speech distortion from the ASR perspective. This distortion degraded the ASR classification accuracy for speech signal corrupted with babble noise. These results prove that SNR and PESQ-MOS improvement using audio enhancement techniques may degrade the ASR classification accuracy instead of improving it.

In [17], an adaptive VAD is used to enhance the performance of the ASR in vehicular environment (Figure 2.4.5). Two adaptive VAD thresholds are used to detect the noise and voiced periods of the speech signal. In the vehicular environment noise estimator is required to avoid the detection of the unvoiced part (nasal sounds) as noise periods which will affect the noise suppression and produce false background noise estimation. Meanwhile in the ASR engine it is necessary to avoid the detection of the nasal sound as voiced sounds which will affect the performance of the ASR engine.

In a vehicular environment, car noise with low frequency can mask the high frequency nasal sounds which can lead to false voiced speech detection. To overcome

these false detection and false background noise estimation, two thresholds are calculated. Using these two thresholds an adaptive VAD in the vehicular environment is constructed. By combining the adaptive VAD algorithm with noise estimation and ASR algorithms, isolated word recognition in car environments achieved accuracy of 94.4%; an enhancement of 14.0% over the baseline recognition method. The adaptive algorithm introduced in [17] is suitable for the real time processing because of its low complexity.

Figure 2.4.5 Adaptive VAD with ASR

In [38], high order statistics (HOS) were utilized to create VAD with high immunity against white Gaussian noise (WGN). It was shown that there were distinct HOS characteristics between the WGN and the speech. These characteristics allow this VAD to have a high performance in low SNR environment. The main disadvantage of this VAD is that it assumes that there should be a WGN in the background of the audio signal; otherwise, the VAD will fail its functionality. This disadvantage limits the functionality of this algorithm to low SNR at WGN environments.

In [14], VAD is used with the Global System for Mobile communications (GSM) to enhance the discontinuous transmission (DTX). DTX utilizes the fact that the parties of the phone call conversation are not active all the time. During the idle time DTX reduces

the bit rate which leads to an effective utilization of the bandwidth as well as reduction of the co-channel interference. In [14], two VADs have been introduced to deal with two different noise types: stationary noise and time-varying noise. A scheme was introduced to combine the two VADs according to the detected noise type. This scheme assumes that the initial state is stationary noise. If time-varying noise is detected for 400 ms, then the system changes state to the time-varying noise. This VAD achieved enhancement of 25.0% over the baseline VAD (stationary VAD) at SNR = 5 dB.

Figure 2.4.6 VAD selection algorithm

2.5 Discussion

Different approaches have been introduced to enhance the performance of the ASC in noisy and reverberant environments. These approaches can be summarized as follows:

- **Features developed for certain environments.** In this approach ASC needs *a priori* knowledge about the environment. In this approach the system is designed for a certain environment (e.g. low SNR [13], [16], [77]).
- **Classifier designed for certain environments.** In this approach the system is designed for a certain environment (e.g. reverberant environment with a particular level of reverberation) [11], [16].

- **Algorithm selection according to the detected audio class.** In this approach ASC changes the mode of operation depending on the detected audio class, such as music, speech, or noise [6].
- **Threshold selection according to the detected noise.** In this approach the threshold is selected according to the detected noise type such as stationary or time varying noise [14].
- **Audio enhancement with *a priori* knowledge of the surrounding environment.** In this approach the background noise is detected then an audio enhancement is performed to enhance the speech from the speech audibility perspective [16].
- **Audio Enhancement without *a priori* knowledge of the surrounding environment.** In this approach audio enhancement techniques are utilized to enhance the audibility of audio signal. These techniques are used as a preprocessing for the ASC to estimate and remove the background noise [12].

From the literature we found that the researchers succeed in addressing the problem of the ASC in certain reverberant or noisy environments (with and without *a priori* knowledge of these environments). Meanwhile the problem of the ASC in variable environment needs to be automated in the variable noisy and reverberant environments. To resolve this problem, the behavior of the features and the ASC classifiers with environment variation need to be investigated. Subsequently, an environment sensor needs to be developed to estimate the environment variable then adapt the performance of the ASC according to this variable. As a result of these adaptations the ASC overall performance will be robust with high accuracy in variable reverberant and noisy environments.

Chapter 3

Experimental and Simulation Environments Setup

In this chapter, the experimental and simulation environments of the introduced methods are presented. Section 3.1 describes the audio database that is used in this thesis. Section 3.2 describes the reverberant and noisy environments. Section 3.3 describes the hardware and the software used to conduct the simulations. Section 3.4 describes the performance evaluation criteria.

3.1 Audio Database

Audio database specification is very important to evaluate the performance of the ASC categories. Audio database should include all possible test cases and not to be biased toward a certain test case. A database example is the database used in TI where the database should include all possible talkers with randomly uttered sentence to make sure that the database is not biased toward a certain sentence. This example guarantees the text-independent database.

3.1.1 Speech Database

The performance evaluation of the TI system is conducted using the KING database [75]. The KING database consists of speech from 51 talkers; each talker has 10 utterances with the exception of 2 talkers, which only have 5 utterances. Data are sampled at a rate of 8 kHz, with 16-bit resolution, and utterance durations ranging from 30 to 60 sec. The talkers in this database utter random sentences which guarantee the text-independent database [75].

3.1.2 Speech/Music Database

Speech/music database used in this thesis is the same database used in [5]. This database was collected from an FM tuner using different broadcasting stations. This database consists of 80 speech and 80 music samples. Data are sampled at a rate of 22.05 kHz, with a 16-bit resolution and duration of 15 sec. This database contains different music types such as pop, country, rock, and new age music.

3.1.3 Speech/Non-Speech Database

We collected a total of 1356 speech and non-speech audio samples from different resources such as recording from door opening and closing. Also some samples were collected from movies' soundtracks and the internet. A full list of the audio classes can be found in Table 3.1.1. Data are sampled at a rate of 16 kHz, with a 16-bit resolution and duration of 0.5 sec. This duration is long enough to conduct the speech/non-speech classification.

Table 3.1.1 Speech/non-speech audio database

Audio class		Number	Number of Audio files
Speech	Male		282
	Female		282
Non-speech	Chair		20
	Cocktail noise		20
	Door		61
	Drawer		20
	Fan noise		169
	Foot-steps		41
	Glass breaking		145
	Rain		50
	Thunder		40
	Traffic		20
	Water		20
	Wind		186
	Total		1356

3.2 Environment Models

The acoustic effect of the environments such as reverberation and noise on the audio signal can be generated using simulated or real environments. The effect of the simulated environment on the audio signal can be generated using a software simulation. As example the effect of the reverberant environment can be simulated by convolving the clean audio signal (original signal) with the impulse response of the simulated environment (Figure 3.2.1). The effect of the real environment on the original audio signal can be generated by producing the clean audio signal in this real environment then recording the emanating sound using an audio capturing device (Figure 3.2.2).

Figure 3.2.1 Simulated environment

Figure 3.2.2 Real environment

3.2.1 Reverberant Environments

In reverberant environments the audio signal propagates between the sound source and the audio capturing device in multi-paths. As shown in Figure 3.2.3 these paths are the direct path between the sound source and the audio capturing device as well as the path of the audio signal reflected by the environment boundaries (e.g. wall, ceiling, furniture). The effect of these multi-paths on the non-reverberated signal (audio source) can be modeled by a linear filter [76]. The causal impulse response of this linear filter is known as the room impulse response. The reverberated audio signal y_n is the result of convolving the audio source signal x_n with the room impulse response h_n .

$$y_n = x_n * h_n \quad (3.2.1)$$

Figure 3.2.3 Reverberation

3.2.1.1 Reverberation Models (Image Method)

The image method is a well known method used to simulate the reverberation at rectangular room acoustics [73]. Four sets of values are the inputs of the simulation program of this method. The four sets are the dimensions of the room, the location of the sound source, the location of the audio capturing device, and the reflection coefficients of the surfaces of the room. The reflection coefficient is defined as the ratio of the amplitude of the reflected audio signal to the amplitude of the incident audio signal on a certain surface [76].

The image method considers multiple order of reflection. Multiple order reflection is the reflection of a sound signal off the surface which has been already reflected off another surface before reaching the audio capturing device. Each time sound signal reflects off the surface the amplitude is attenuated according to this surface reflection coefficient.

Assume that we have a rectangular room and we are looking to the plan (horizontal cross section) as shown in Figure 3.2.4. To create a first order reflection, a virtual source is set outside the room. The sound is attenuated according to the distance from the virtual

source to the audio capturing device. The amplitude of the attenuated sound of the virtual source is then multiplied by the reflection coefficient of the walls. The total real path from the source to the audio capturing device is illustrated by the line $a + b$. The virtual path of the simulation software is illustrated by line $c + b$. From Figure 3.2.4 it can be shown that virtual source x is the image of the source X .

Figure 3.2.4 First order reflection

Second order reflection is shown in Figure 3.2.5. This figure illustrates the propagation of the sound from the source to the audio capturing device after two reflections. It is shown that the number of times the sound wave is reflected (solid line) is the number of times that the sound wave penetrating a boundary. In this second order reflection the amplitude of the sound wave is attenuated from the virtual sound source to the audio capturing device and then multiplied by the reflection coefficients of the two walls.

Figure 3.2.5 Second order reflection

The simulation program in this thesis uses the image method with multiple virtual sources (images) as shown in Figure 3.2.6. The bold box represents the real room. The source, the audio capturing device and the virtual sources are represented by X, O, and x's, respectively (Figure 3.2.6).

Figure 3.2.6 Spatial arrangement of image sources used in the simulation program

3.2.1.2 Reverberation Specifications

Reverberation can be characterized by the reverberation time (RT60) [11]. RT60 is defined as the time duration required for the energy decay curve of the impulse response of a certain environment to decay 60dB. Energy decay curve EDC is defined as follows:

$$EDC(t) = \int_t^{\infty} h^2(\tau) d\tau \quad (3.2.2)$$

where $h(\tau)$ is the impulse response of the surrounding environment.

The simulated RT60 value is selected to cover the reverberation range from a 0.25 sec to 1.80 sec (Table 3.2.1). This range is selected based on the literature review [11] and the audibility of the audio signal. The effect of the impulse response with RT60 = 0.25 sec is hardly perceptible and the effect of impulse response with RT60=1.80 sec is very clear (like a large cathedral sound).

Table 3.2.1 RT60 simulation values

RT60 (sec)	0.25	0.50	0.68	0.85	1.00	1.15	1.25	1.35	1.50	1.65	1.80
---------------	------	------	------	------	------	------	------	------	------	------	------

For each RT60 value, four different surrounding environment sets are created. These four environment sets have the same RT60 as shown in Table 3.2.1; however each set has a different room setup. As a result 44 different reverberant environments are generated in addition to the non-reverberant (clean) environment. A common factor for each environment set is the distance d between the sound source and the audio capturing device. These distances d are chosen to be 0.75, 0.87, 1.00 and 1.25 m (Appendix B).

3.2.2 Noisy Environments

The captured signal in the noisy environments is the sum of the original signal and the background noise. This captured signal is known as the noisy signal. The mathematical model of the captured signal y_n is as follows:

$$y_n = x_n + N_n \quad (3.2.3)$$

where x_n is the audio source signal and N_n is the background noise signal.

3.2.2.1 Noise Models

Noise is as random or pseudorandom sound signals. White noise contains a uniform spectrum energy distribution, while colored noise contains a non-uniform spectrum energy distribution, such pink and brown noises [25], [31]. In this thesis additive white Gaussian noise and pink noise are used as a noise source N_n .

3.2.2.2 Noise Specifications

The amount of the noise in the audio signal is evaluated by the signal-to-noise ratio (SNR). SNR is defined as the power ratio between the audio signal and background noise.

The simulated SNR is selected to cover the noise range from a -20 dB to 50 dB. This range is selected based on the literature review [5], [77] and the audibility of the audio signal, where the effect of the noise at $\text{SNR} = 50$ dB is hardly perceptible and the

effect of noise at SNR = -20 dB is very clear (the audio signal is totally buried under the noise level).

3.3 Simulation Environment (Hardware and Software)

The simulation of the introduced algorithm is done using Matlab™ software v 7.x. The operating system is Microsoft Windows™ (Windows XP™), RAM storage capacity of 1 ~2 GB ,and the hardware is the Intel™ P4 processor with speed of 1.5~3.0 GHz. The audio files are saved as wav files (Windows PCM).

3.4 Performance Evaluation Criteria

Performance evaluation of the work done in this thesis and the previous work is described in the following subsections.

3.4.1 Baseline Comparison Method

The performance of the introduced methods in this thesis is compared with baseline methods that are trained in clean environment and tested in reverberant or noisy environments. All ASC systems in this thesis are evaluated using a closed-set testing; known audio classes are limited and the unknown audio (test) is one of the audio classes in the known set (training).

3.4.2 Classification Accuracy

Audio database is divided into training and test data sets. Closed-set ASC performance is evaluated using classification accuracy. ASC classification accuracy is the

number of correct times that the audio signal of the unknown audio class is identified by the ASC system divided by the total number of the trial times:

$$\text{Classification Accuracy} = \frac{\text{number of correct trials}}{\text{total number of trials}} \times 100\% \quad (3.4.1)$$

For binary classification problem:

$$\text{Classification Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \quad (3.4.2)$$

where TP is true positive, TN is true negative, FP is false positive and FN is false negative (Figure 3.4.1 and Figure 3.4.2). For multiple class problem, classification accuracy is the diagonal sum of the confusion matrix divided by the sum of all the elements of the confusion matrix.

Figure 3.4.1 Classification accuracy parameters

		Predicted Class	
		+	-
Actual Class	+	TP	
	-		TN

Figure 3.4.2 Confusion table of classification accuracy

Binary classification classifies the database into two classes. In this classification it is important to keep the balance between the two classification measurements; sensitivity and specificity; $\text{sensitivity} = \text{TP}/(\text{TP}+\text{FN})$ and $\text{specificity} = \text{TN}/(\text{TN}+\text{FP})$. In some cases the sensitivity and specificity are independent where both can achieve 100%. Typically, the relation between the sensitivity and specificity is a trade off relation. This is because the threshold that separates the two classes can be biased toward one of the classes (Figure 3.4.1). As shown in Figure 3.4.1 high sensitivity can be achieved if the threshold is set to a very low value, meanwhile the classification system will have a low specificity. To select the threshold that maximizes both measurements; sensitivity and specificity, a combination between these two measurements to balance the classification is required (i.e. find the threshold that maximize both measurements).

3.4.3 Matthews's Correlation Coefficient

Mathews' Correlation Coefficient (MCC) is used in pattern classification to combine the sensitivity and specificity of the binary classification into one measurement [52]. This combination is used to measure the balancing of the binary classification threshold even if the two classes have different sizes. MCC is calculated as follows [52]:

$$MCC = \frac{TP \times TN - FP \times FN}{[(TP + FP) \times (TN + FN) \times (TP + FN) \times (TN + FP)]^{0.5}}, \quad -1 \leq MCC \leq +1 \quad (3.4.3)$$

MCC value ranges between -1 and +1. At $MCC = +1$, both sensitivity and specificity have 100% prediction accuracy which represent the best possible prediction, and at $MCC = -1$, both sensitivity and specificity have 0% prediction accuracy which represent the worst possible prediction. In this thesis MCC is used to tune the threshold of the binary classification to maximize both measurements; sensitivity and specificity.

Chapter 4

Performance of ASC in Reverberant and Noisy Environments

In this chapter we discuss the impact of reverberant and noisy environments on ASC performance. The performance of ASC systems is degraded when training and test audio are captured in mismatched environments. This performance degradation will be discussed in this chapter using examples of ASC categories such as talker identification, speech/non-speech classification and speech/music classification. Section 4.1 describes the ASC performance in clean environments, section 4.2 describes the ASC performance in reverberant environments and section 4.3 describes the ASC performance in noisy environments.

4.1 Performance of ASC in Clean Environments

To investigate the effect of reverberant and noisy environments on the ASC performance, the performance of the ASC in clean environment should be evaluated. This performance is then compared with the ASC performance in the presence of

reverberation and noise. The following sub-sections provide interpretation and analysis of the accuracy performance of ASC categories examples.

4.1.1 Speech/Non-Speech Classification

Several approaches have been introduced in the literature to classify speech and non-speech audio segments such as LPC-based algorithm [33], the whole segment algorithm [21] and the pitch ratio algorithm which we introduced in [45].

LPC-based algorithms extract three features and compare it to three thresholds. The unknown audio segment is assumed to be speech by analyzing the following features:

- **Energy of the frames of speech signal**, which should change smoothly, without abrupt changes. Audio segments with large energy variation are assumed to be non-speech (Figure 4.1.1).
- **Duration of the speech signal**, which is usually in the order of 70 milliseconds [33].
- **Pitch contour of the speech signal**, which should change smoothly, within 15 Hz limit. The audio segment is classified as nonspeech if the pitch contour has larger variation than 15 Hz within successive frames (Figure 4.1.1) [33].

Figure 4.1.1 Audio segments and frames

The whole segment algorithm calculates the presence of the pitch in the whole audio segment. If the pitch is detected in the audio segment then it is considered as a speech segment; otherwise, it is considered as a non-speech segment.

In [45], we introduced the Pitch Ratio (PR) algorithm to perform speech/non-speech classification in clean environment (Figure 4.1.2). PR algorithm divides the audio segment into frames, estimates the presence of the human pitch in each frame, and calculates a PR parameter. If the PR parameter is above a certain threshold it is considered a speech segment; otherwise, it is considered a non-speech segment. To compute the pitch, a modified autocorrelation pitch detection method is used [43]. Input audio segments are normalized and then low-pass filtered with a cutoff frequency of 900 Hz; voiced part of the speech signal is concentrated at frequency < 1000 Hz [21]. Each segment is subdivided into smaller overlapping frames as shown in Figure 4.1.1. The frame size (FS) can be chosen as follows:

$$20ms \leq FS \leq SD \quad (4.1.1)$$

where SD is the segment duration.

Figure 4.1.2 Pitch Ratio Algorithm

Pitch detection is performed on the frames using the autocorrelation function. When the autocorrelation is computed directly from the speech signal, it is difficult to distinguish the peaks of formant² from those due to the fundamental periodicity (pitch) [44]. The formant can be found in the speech signal as a peak in the frequency response, due to the vocal tract vibrations. Formant should be removed before computing the autocorrelation function of the speech signal and without distorting the periodicity of the speech (pitch period). This can be accomplished by applying center clipping on the speech signal prior to computing the autocorrelation [44]. Center clipping is applied to each frame, which can be calculated as follows:

² A formant is a preferred resonating frequency of any acoustical system.

$$c(n) = \begin{cases} s(n) - c_t & \text{if } s(n) > c_t \\ s(n) + c_t & \text{if } s(n) < -c_t \\ 0 & \text{otherwise} \end{cases} \quad (4.1.2)$$

where $c(n)$ is center clipped sample at time index n , $s(n)$ is the audio sample at time index n , and c_t is center clipping threshold. The center clipping threshold is calculated as:

$$c_t = L \min[c_{t1}, c_{t2}] \quad (4.1.3)$$

where L is equal to 0.64 [43], and c_{t1} and c_{t2} are the maximum temporal amplitude of the first and last third of the frame, respectively.

The pitch value is the index of the maximum peak in the autocorrelation function of the frame, which must exceed a threshold equal to 0.4 of the root mean square energy (RMSE) of the frame; otherwise, the pitch will not be considered present in the frame:

$$\begin{aligned} \text{Pitch} &= \arg(\max_{\tau} \{R_{xx}(\tau) : R_{xx}(\tau) > 0.4RMSE\}) \\ \text{where} \\ R_{xx}(\tau) &= E[c(n)c(n+\tau)] \end{aligned} \quad (4.1.4)$$

Once the pitch value is computed, it is determined whether or not it is considered a Human Pitch by comparing its value to the human pitch range [45]:

$$\text{Human Pitch} = \{\text{Pitch} : 70\text{Hz} < \text{Pitch} < 280\text{Hz}\} \quad (4.1.5)$$

A pitch counter is used to keep track of the number of frames in the audio segment that have human pitch; therefore, this counter is incremented only in the presence of

human pitch. Pitch Ratio (PR) is defined as the ratio of frames with human pitch to the total number of frames in an audio segment:

$$PR = \frac{NP}{NF} \quad 0 \leq PR \leq 1 \quad (4.1.6)$$

where NP is the numbers of frames that have a human pitch and NF is the total number of frames. The total number of the frames can be calculated as follows:

$$NF = \left\lfloor \left(\frac{SD - FS}{1 - OL} \right) / FS \right\rfloor + 1 \quad (4.1.7)$$

where SD is the audio segment duration, FS is the frame size, and OL is the overlap ratio (eg. 0.50). $\lfloor \bullet \rfloor$ is the floor operator.

To discern speech from non-speech audio segments, a threshold is used on the PR parameter. Segments with a PR above this threshold would be classified as speech, while segments with a PR below this threshold would be classified as non-speech.

$$PR \text{ decision} = \begin{cases} \text{Speech} & PR \geq \text{Threshold} \\ \text{Non-speech} & PR < \text{Threshold} \end{cases} \quad (4.1.8)$$

For clean environment ($SNR > 30$ dB), the speech/non-speech PR optimal threshold (i.e. the threshold that maximizes classification accuracy) is a function of frame size, overlap ratio, and segment duration (e.g., with $FS = 50$ ms, $OL = 0.15$, and $SD = 450$ ms the PR threshold = 0.245) (Figure 4.1.3). The threshold is chosen to maximize the classification accuracy.

MCC, as described in section 3.4.3, is used to maximize the classification accuracy of the speech/non-speech classification. The corresponding threshold of the maximum MCC value is the PR threshold. This method guarantees the maximization without biasing the classification threshold toward the speech or non-speech (Figure 4.1.3).

Figure 4.1.3 (a) PR distribution of speech and non-speech audio segment (b)MCC variation with PR threshold. (c) Speech and non-speech classification accuracies with the threshold.

The speech/non-speech algorithm is trained using half of the 1356 speech/non-speech database audio segments and tested with the other half using clean database. As shown in Table 4.1.1 PR algorithm has the highest average classification accuracy in clean environments, outperforming the LPC-based and whole segment algorithms (PR = 97.5%, LPC-based = 86.0%, and whole segment = 90.7%). Speech/non-speech classification algorithms show a high classification performance (classification accuracy > 86.0%) in clean environment.

Table 4.1.1 Comparison of Speech/Non-speech classification accuracy using three algorithms in clean environment (%)

Algorithm Audio class	PR	Whole Segm.	LPC
Speech	96.5	84.0	77.7
Non-speech	98.5	97.4	94.3
Average	97.5	90.7	86.0

4.1.2 Speech/Music Classification

As discussed in section 2.3.1.2 there are several algorithms have been introduced in the literature to perform speech/music classification. Seven features were simulated using speech/music database. This database is described in section 3.1.2. In total, there were 1320 music and 1320 speech one-second audio segments, used to test the speech/music algorithms, and 300 music and 300 speech one-second audio segments used to train the speech/music algorithms. As shown in Table 4.1.2, the classification accuracy of speech/music classification using the seven algorithms ranges from 77.7% to 94.0%. The best performing algorithm is the 4Hz modulation, while the worst performing algorithm is the ZCR.

Table 4.1.2 Comparison of speech/music classification accuracy using seven algorithms in clean environment

Algorithm	Classification accuracy (%)
4Hz	94.0
Entropy	92.0
PR	79.0
Segmentation	82.0
Centroid	80.6
Low Energy Frames	85.7
ZCR	77.7

4.1.3 Talker Identification

GMMs are used commonly for TI [52]. During training, the GMM parameters of each talker are determined from a sequence of M training vectors. First, the clustering algorithm Linde, Buzo, Gray (LBG) groups the vectors into clusters [56]. Second, for each cluster the mean and variance are then computed. During testing, the probability that a given talker uttered a certain test speech is computed from the GMMs. Using extracted N feature vectors from the test utterance, the TI engine computes a log-likelihood probability vector as follows:

$$L = [l(1) \ l(2) \ \dots \ l(S)]^T \quad (4.1.9)$$

where S is the number of elements in this vector equal to the numbers of the talkers.

The engine performs TI by selecting the talker that uttered the test speech as the talker with the highest log-likelihood value.

The baseline TI method is trained using 3 utterances from each of the 51 talkers of the KING database (section 3.1.1). The performance of the TI is tested using the remaining 7 utterances of 49 talkers and 2 utterances of the exceptional 2 talkers (as described in section 3.1.1, these talkers only have a total of 5 utterances). There are a total of 347 utterances in the test set and each one is a 15 second utterance. The utterances in the test set are different from those used in the training set (the test and training set are mutually exclusive). A 32-GMM is used as classifier, with a feature vector of 16 MFCCs. This setting is proven to be the best TI performance in clean environment [11].

Training and testing of the TI engine is conducted using a clean KING database. The TI engine has a high classification accuracy of 97.4%. This classification accuracy is expected to be degraded with the variation of the surrounding environment. This degradation will be addressed later in this chapter.

4.2 Performance of ASC in Reverberant Environments

Performance of ASC systems is severely degraded when training and test audio are captured in mismatched environments [16], [13]. The captured audio can differ because of the different levels of reverberation of the surrounding environment. In the following subsections the reverberant environments are created by convolving the clean audio databases with the simulated impulse responses as described in section 3.2.1.

4.2.1 Speech/Non-Speech Classification

Simulation is done to show and compare the performance of the PR algorithm, LPC-based algorithm, and the whole segment algorithm in variable reverberant environments.

The speech/non-speech database is filtered using 11 impulse responses ($RT60 = 0.25, 0.50, 0.68, 0.85, 1.00, 1.15, 1.25, 1.35, 1.50, 1.65, \text{ and } 1.80 \text{ sec}$) to create the reverberated database; $d = 1.00 \text{ m}$, in addition to the clean database. The training of the algorithms is conducted in a clean environment (clean database). The PR algorithm is trained using half of the 1356 speech/non-speech database audio segments and tested with the other half.

Figure 4.2.1 is a plot of the speech/non-speech classification accuracy versus the reverberation time (RT60) using PR algorithm, LPC-based algorithm, and the whole segment algorithm. Results demonstrate that the reverberation degrades the performance of the three algorithms. PR algorithm has the lowest classification accuracy degradation (at RT60 = 1.80 sec the decrease of the classification accuracy of PR = 3.5 %, whole segment = 4.7%, and LPC-based = 6.0%).

The PR algorithm has the highest average classification accuracy in reverberant environments, outperforming the LPC-based and whole segment algorithms. The LPC-based algorithm is not resilient to the presence of the reverberation [33], with classification accuracies that are more than 13.0% lower than the PR algorithm. LPC-based algorithm has poor performance in the presence of the reverberation, because of the thresholds that its classification decision relies upon. For example in reverberant environments, the pitch contour threshold, which relies on the smooth changes in the pitch value, will fail because the pitch detection algorithm will not detect the presence of the pitch in a significant number of frames of the speech segment. This will produce abrupt change of the pitch value and result in misclassification of speech segments, as non-speech segments.

Figure 4.2.1 Comparison of Speech/Non-Speech classification accuracy using three algorithms in reverberant environment (%) (PR parameter FS = 60 ms, OL = 0.85 and PR Threshold = 0.32)

In the whole segment algorithm, pitch detection is conducted only once using the whole audio segment. This algorithm can fail to detect a pitch due to reverberation.

The PR algorithm divides audio segments into smaller overlapping frames and pitch detection is performed on these frames. The PR algorithm determines the presence of pitch, by comparing it to a PR threshold. To enhance the performance of the PR algorithm an adaptation to this threshold should be conducted. This adaptation will be presented in section 6.1.1.1.

4.2.2 Talker Identification

The TI engine is trained in a clean environment. A 32-GMM is used as classifier, with a feature vector of 16 MFCCs.

Same procedure was used as was done in section 4.1.3. The test utterances are filtered using new 33 impulse responses ($RT60 = 0.25, 0.50, 0.68, 0.85, 1.00, 1.15, 1.25, 1.35, 1.50, 1.65, \text{ and } 1.80 \text{ sec}$; three different room setting for each $RT60$) to create the reverberated speech; $d = 0.75, 0.87 \text{ and } 1.25 \text{ m}$, in addition to the clean speech. Each utterance is a 15 sec.

Figure 4.2.2 plots the results of TI engine which has been trained in clean environment. The maximum performance of TI occurs when the test data from clean environment is used with the TI engine trained in that environment. Results demonstrate that reverberation cause degradation in the performance of the TI engine which has been trained in clean environment (at $RT60 = 1.80 \text{ sec}$, accuracy performance = 76.0 %; a degradation of 21.0%).

Figure 4.2.2 Classification accuracy of TI engine (trained in clean environment) in variable reverent environment

4.3 Performance of ASC in Noisy Environments

Noise can severely degrade the performance of ASC [81], [82]. More specifically, mismatches between the training and testing environments result in decreased accuracy; these mismatches can arise due to a variable noise environment. In the following subsections the performance of different ASC categories in noisy environments will be discussed.

4.3.1 Speech/Non-Speech Classification

Simulation is done to show and compare the performance of the PR algorithm, LPC-based algorithm, and the whole segment in variable noisy environment.

The speech/non-speech algorithms were trained using half of the 1356 speech/non-speech database audio segments, using clean database and were tested using the other half over an SNR range of clean environment to -10 dB, using white and pink noise.

In Figure 4.3.1, results demonstrate that noise degrades the performance of the three algorithms. At SNR = -10 dB all the algorithms fail to perform speech/non-speech classification, have an accuracy of 50%. This outcome is due to the distortion of the pitch periods in the severe noise environment. At SNR = -5 dB the whole segment algorithm has the highest classification accuracy. For SNR > -3 dB, the classification accuracy of the PR algorithm is superior to the other speech/non-speech classification algorithms. For example, at SNR = 5 dB, PR has an accuracy of 95.0%, the whole segment algorithm has an accuracy of 85.0%, and the LPC-based algorithm has an accuracy of 53.0%.

Figure 4.3.1 Comparison of Speech/Non-Speech classification accuracy using three algorithms in noisy environment (%) (PR parameter FS = 60 ms, OL = 0.85 and PR Threshold = 0.32)

4.3.2 Speech/Music Classification

Simulation is done to show and compare the performance of different speech/music classification algorithms (Centroid, ZCR, Entropy, Low Energy Frames, 4 Hz, PR, and Segmentation) in variable noisy environment. The speech/music classification using the seven algorithms is trained with 300 music and 300 speech one-second audio segments, using clean database and tested with 1320 music and 1320 speech one-second audio segments over an SNR range of -20 dB to clean environment, using white noise.

In Figure 4.3.2, results demonstrate that noise degrades the performance of the seven algorithms. At SNR < -10 dB, all the algorithms fail to perform speech/music classification, have an accuracy of 50%. At SNR = 10 dB the Centroid algorithm achieved the highest classification accuracy of 94.0%. At clean environment the 4 Hz algorithm achieved the highest classification accuracy of 94.0%. The simulation shows

that different algorithms have different classification accuracy performance at different SNR value.

Figure 4.3.2 Comparison of Speech/music classification accuracy using seven algorithms in noisy environment

Centroid algorithms exhibit higher classification accuracy at the SNR range 20 to 5dB than the SNR range 30 to clean environment. This emanates from the fact that the variance of the feature is used as the discern value, not the feature itself. The addition of the noise to the speech signal affects the value of the unvoiced part more than the voiced part so the variance will have a higher value; however, music audio class is not significantly affected, so the classification accuracy is increased until the noise dominates the energy of the signal.

In speech/music classification the variance of the ZCR is used as the discern value, not the ZCR itself. Figure 4.3.3 explains the drop of the performance of the variance of the ZCR in Figure 4.3.2 at the SNR range 12.5 to -5dB. The threshold that used in this

algorithm is the same threshold determined from the training of the ZCR algorithm using clean database. At SNR = -5 the clean training threshold has a value close to the optimal threshold value; optimal threshold value = 0.0027 and clean threshold = 0.0033. At SNR = 2dB the optimal threshold value has a value far from the clean training threshold; optimal threshold value = 0.0085 and clean threshold = 0.0033. The classification accuracy can be enhanced if the threshold is adapted with the SNR value. This adaptation will be discussed in section 5.2.1.

Figure 4.3.3 Speech/music classification distributions using ZCR and fixed threshold at different SNR (a) SNR = -5 dB (b) SNR= 2 dB (c) SNR = 40 dB.

4.3.3 Talker Identification

Same procedure was used as was done in section 4.1.3. Test utterances, used ten SNR values (SNR($j = 1$) is clean environment and SNR($j \in \{35, 30, 25, 20, 15, 10, 5, 0, -5, -10\}; j = 2, 3, \dots, 11$), with two different noise types; white and pink noise, to create the

noisy speech. A total of 20 test noisy environments is used, in addition to the clean environment.

Figure 4.3.4 Classification accuracy of TI engine trained in clean environment and tested in variable noisy environment

Results demonstrate that noise degrade the performance of the TI engine which has been trained in clean environment (at SNR = -10 and SNR = 25 dB the TI achieved average classification accuracy of 2% and 71%, respectively; a degradation of 95.4% and 26.4% less than the classification accuracy in a clean environment, respectively). These results reconfirm the findings that the closer the match between the training and testing environments the higher the TI accuracy.

4.4 Discussion

Different scenarios of the impact of the variable environments on the ASC performance have been studied. The concluded results from these scenarios can be summarized as follows:

- The closer the match between the training and testing environments the higher the ASC accuracy.

- Different algorithms have different performance in different environment.
- Classification threshold should be adapted with the environment variable such as the RT60 and SNR.
- The classification accuracy of the ASC can be enhanced if the surrounding environment information is available.

From the study that conducted in this chapter, we found that the training of the ASC categories in clean environment can achieve high classification accuracy if tested in clean environment. However ASC categories have performance degradation if there is a mismatch between the training and test environments. The ASC features and classifiers should be adapted according to the variable environments. As a result of these adaptations the ASC overall performance will be enhanced in variable environments.

Chapter 5

Solutions for ASC in Reverberant and Noisy Environments

In Chapter 4, we introduced the impact of the variable environment on the ASC. It is shown that the accuracy performance of the ASC is severely degraded when the training of the ASC algorithm is conducted in clean environment, while the testing of the ASC algorithms is conducted in variable reverberant and/or noisy environments.

In this chapter, a general framework is introduced to enhance the performance of the ASC in variable environments. To achieve this framework, new feature extraction methods are introduced. Also, adaptation techniques for ASC systems in variable environments are introduced. As a result of these methods and techniques the overall ASC performance will be enhanced in variable environments.

5.1 ASC in Reverberant and Noisy Environments

As mentioned in Chapter 2, conventional methods to improve the ASC performance in reverberant and noisy environments are attempted using audio enhancement techniques as a preprocessing to the ASC. In these methods the audio signal is enhanced without relying on *a priori* knowledge of the surrounding environment. This enhancement

improves the audio signal audibility by removing the effects of the environment (e.g. de-noising or de-reverberation) which improves the audio signal from the human perspective and may introduce audio distortion from the ASC perspective [12].

Other methods have been introduced to enhance the ASC classification accuracy using *a priori* knowledge of the surrounding environment. This *a priori* knowledge is used to develop an ASC system for a specific environment [11], [34], [16], [77]. These methods ignored that there is a possibility that this ASC system may be operated in unknown non-clean surrounding environments which disallows the generalization of the ASC system in variable environments.

To generalize the ASC in reverberant and noisy environments, ASC should be adaptive with these environments. The adaptation of the ASC should consider the main stages of the ASC system (i.e. audio capturing, feature extraction and classifier). In this thesis, two stages are considered: feature extraction (Chapter 6) and classifier (Chapter 7). For audio capturing, a microphone array is recommended due to the fact that the microphone array improves the SNR and can also reduce the effect of reverberation at certain directions [83]-[85].

To automate this generalization, a new environment sensor should be developed (Chapter 7). This sensor should extract information from the captured audio signal at certain environment, and then adapt the ASC system to this environment. As a result of this adaptation the overall performance of the ASC should be enhanced in variable environments (Figure 5.1.1).

Figure 5.1.1 ASC generalization in variable environments

5.2 Feature Extraction

In ASC, feature extraction involves computing a set of values from the audio signal that enables classification. Thus, features from the same audio class should have values that are similar, while features from different audio classes should have values that are dissimilar. The impact of different environments on the classification accuracy of an ASC system depends on which audio features are being utilized [86].

ASC applications with a known surrounding environment can be implemented with a certain set of features known to have high performance in this environment. If the surrounding environment has harsh conditions (i.e. low SNR or high reverberation) a set of resilient features in this environment are required. A problem arises if the surrounding environment changes and the features are sensitive to these changes; ASC performance decreases as a result.

To illustrate the effect of a variable environment on the audio features, assume that we have two audio classes and one feature to be used to classify these two classes. As shown in Figure 5.2.1, the classifier uses a simple decision threshold with the following classification criteria:

$$\text{Classification decision} = \begin{cases} \text{Class A} & \text{Feature} > \text{Threshold} \\ \text{Class B} & \text{Feature} \leq \text{Threshold} \end{cases} \quad (5.2.1)$$

Figure 5.2.1 (a) Features at clean env. (b) First scenario after env. change with clear decision threshold. (c) Second scenario after env. change without clear decision threshold.

Two example scenarios are shown when the surrounding environment is changed. The first scenario shows that the distribution of this feature will be changed; meanwhile, the feature value of the two classes is still distinct from each other. In this scenario, the decision threshold could be adapted to accommodate this surrounding environment and maintain high classification accuracy. In the second scenario, this feature does not have a

clear decision threshold that is capable of separating the two classes. This will lead to a classification failure. In this scenario, another feature with a clear threshold should be selected. To resolve the issues identified in the two scenarios, environment information could be used. In the first scenario, environment information could be used to adapt the threshold of the classifier. In the second scenario, environment information could be used to select a feature that is appropriate for the given environment.

5.2.1 Adaptive Feature Threshold

In the first scenario, illustrated in Figure 5.2.1(b), the feature threshold should be adapted to the environment variable to maximize the classification accuracy. By adapting the threshold from the clean environment value (threshold = 50) to the optimal threshold of the new environment (threshold = -18), the classification accuracy will be 72.0% instead of 50.0%.

To adapt the features, two pieces of information should be available. The first is the value of the environment variable. The second is the mapping between the environment variable and the threshold, which maximizes the classification accuracy (Figure 5.2.2).

Figure 5.2.2 Adaptive Feature Threshold

5.2.2 Multi-Feature Fusion ASC

To implement a generalized ASC with high performance in reverberant and noisy environments, a Multi-Feature ASC system should be implemented to handle different environments (Figure 5.2.3). Using the environment information, the system fuses the decisions of the T features. Two fusion methods are introduced. The first method, the Multi-Feature Selection method, selects the best performing feature out of T features, in a certain environment. The system decision is equal to the decision associated with this single ASC feature. The second method, the Multi-Feature Combination method, intelligently weights and combines the output decisions of the best performing M ($M \leq T$) features in a certain environment.

Figure 5.2.3 Multi-Feature ASC

5.2.2.1 Multi-Feature Selection

Different features have different classification accuracies, which can also change with different environments. Environment information can be utilized to select the best performing feature in a certain environment (Figure 5.2.4).

Figure 5.2.4 Multi-Feature Selection method

As shown in Figure 5.2.4, the environment information selects the best performing feature in certain environment. The best performing feature is determined *a priori* using the training stage in different environments. The Multi-Feature Selection method can

enhance the classification accuracy particularly if it is combined with the Adaptive Feature Threshold method (Figure 5.2.5).

Figure 5.2.5 Multi-Feature Selection with Adaptive Feature Threshold

The combined Adaptive Feature Threshold and Multi-Feature Selection method enhances the classification accuracy; however this combined method uses only one feature to make the classification decision. Another method can be developed by using a combination of features. This Multi-Feature Combination method can be developed using a multi-expert method [86].

5.2.2.2 Multi-Expert Method

Further improvements in ASC performance are obtained by using a multi-expert system. In such a system, there are multiple experts, each performing audio classification on a single feature. Assume that we have S mutually exclusive audio classes C_i ($i \in \{1, 2, \dots, S\}$) and we have T experts Ex_t ($t \in \{1, 2, \dots, T\}$). The number of experts is equal to the number of features in this multi-expert system. An unknown audio event h is

captured and is classified by this multi-expert system. Each expert outputs information on one of three levels, which is used to classify the event h to one of the audio classes C_i [87]. Information levels are categorized and named in [87] as follows:

- Abstract level: Expert Ex_i indicates that the unknown audio event $h \in C_i$ (Figure 5.2.6)
- Rank level: Expert Ex_i indicates that the unknown audio event h has a rank vector $R_i = [r_i(1) \ r_i(2) \ \dots \ r_i(S)]$. This rank vector is sorted so it reflects the degree of the confidence that the expert assumes $h \in C_i$, $i = r_i(s)$. The highest degree of confidence that $h \in C_i$ when $i = r_i(1)$ (Figure 5.2.7).
- Measurement level: Expert Ex_i indicate that the unknown audio event h has a measurement vector $M_i = [m_i(1) \ m_i(2) \ \dots \ m_i(S)]$. The elements of this measurement vector $m_i(s)$ provide information about the expert's confidence that $h \in C_s$ (Figure 5.2.8).

Figure 5.2.6 Abstract information level (example)

Figure 5.2.7 Rank information level (example).

Figure 5.2.8 Measurement information level (example).

As shown in Figure 5.2.6 - Figure 5.2.8, the measurement level is the highest level of the information. The rank vector can be generated from the measurement vector as follows:

$$R_t = \arg \underset{s}{\text{sort}}(M_t(s)) \quad (5.2.2)$$

Abstract level information can be generated from the rank vector as follows:

$$h \in C_i \text{ where } i = r_t(1) \quad (5.2.3)$$

Experts' opinions can be combined on different levels of information. Abstract level is the simplest combination level of the opinions; however performing the combination on a higher level can enhance the ASC performance. In the following subsections different methods of combining the opinions of experts on different information levels will be discussed.

5.2.2.2.1 Multi-Experts on Abstract Level

Majority Vote is a simple method of combining the multi-expert opinion on the abstract level. At the abstract level each expert Ex_t produce a decision d_t for each class C_i using the unknown audio input h as follow:

$$d_t(h \in C_i) = \begin{cases} 1 & \text{if } Ex_t(h) = i \\ 0 & \text{otherwise} \end{cases} \quad (5.2.4)$$

Majority Vote rule is to decide $E(x) = i$ only if the largest group (major) number of the experts agreed upon class i . The largest group is determined by the number of experts that exceed a certain threshold as follows:

$$E(h) = \arg \underset{i}{\max} \left[\max \left[\sum_{t=1}^T d_t(h \in C_i) \right] \right] \text{ if } \max \left[\sum_{t=1}^T d_t(h \in C_i) \right] \geq \delta T \quad (5.2.5)$$

where δ ($0 < \delta \leq 1$) is constant, that determines the number of experts needed to agree upon an unknown audio event to be considered as one of the known audio classes. The most conservative case when $\delta = 1$ where all the experts need to agree on the same class to be considered as a valid decision. A typical value is $\delta \geq 0.5$ where half or more of the experts need to agree upon certain audio class to be valid decision.

The Majority Vote method is simple and does not need training to produce the output. Meanwhile this method has disadvantages that the degree of confidence of each expert in certain audio class is not considered and all the experts are treated as they have the same level of knowledge (i.e. the classification accuracy of each expert) of a certain audio class.

5.2.2.2.2 Multi-Experts on Rank Level

Unlike the abstract level where each expert Ex_t has binary output decision (i.e. 0 or 1) for each class, at the rank level each expert Ex_t has a rank vector R_t . This vector reflects the expert's confidence level that the unknown audio event h is related to each audio class (Figure 5.2.6b). Borda count is a method of combining the multi-expert opinion on the rank level. Based on the rank vector R_t with S audio classes, Borda count, of class s for expert t , $q_t(s)$ is calculated as follows:

$$q_t(s) = S - \arg \underset{n}{[r_t(n) = s]} \quad (5.2.6)$$

where $\arg \underset{n}{[r_t(n) = s]}$ is the ranking of class s for expert t . The combined Borda count from the T experts is calculated as follows:

$$Q(s) = \sum_{t=1}^T q_t(s) \quad (5.2.7)$$

The combined decision $E(h)$ is calculated as follows:

$$E(h) = \arg \max_s [Q(s)] \quad (5.2.8)$$

The disadvantage of the rank level Borda count method is that all experts are treated as they have the same level of knowledge (expected classification accuracy) of a certain audio class.

5.2.2.3 Multi-Feature Combination

Unlike the abstract level and rank level, a Multi-Feature Combination method that weights the experts decision according to their level of knowledge is developed. This Multi-Feature Combination method is a modified version of Borda count with weighted experts which can enhance the performance of the ASC. Two modifications have been introduced. The first modification is adding a dynamic weight $w_{Env,t}$ for each expert t , where the behavior of the experts changes as a function of the environment Env variable as follows:

$$Q^w(s) = \sum_{t=1}^T w_{Env,t} q_t(s) \quad (5.2.9)$$

The second modification is to create a pool of experts T that uses the best M weighted experts (i.e. best-performing M features), where $M \leq T$. In this modification the weight

vector is modified to zero out the contribution of all experts except the best-performing M features. Modified summation of Borda counts from individual experts becomes as follows:

$$Q_M^w(s) = \sum_{t=1}^T w_{Env,t,M} q_t(s) \quad (5.2.10)$$

As a consequence the Multi-Feature Combination method decision is as follows:

$$E(h) = \arg \max_s \left[Q_M^w(s) \right] \quad (5.2.11)$$

The block diagram of the Multi-Feature Combination method is shown in Figure 5.2.9.

In Section 5.2 it has been shown that Adaptive Feature Threshold and Multi-Feature Fusion can enhance the performance of the ASC. To achieve this enhancement the environment information should be extracted (estimated) from the captured audio signal.

Figure 5.2.9 Multi-Feature Combination method

5.3 Environment Information Estimator

Environment information estimator is desirable to enable the fusion of the multiple features as indicated in the previous section as well as the multiple ASC engines which will be discussed in the following section. Output of this estimator should be a discrete value that matches one of the training environments' variable. This estimator provides information about the environment variable in the surrounding environment by extracting features from the captured audio sounds emanating within this environment.

To implement the environment estimator we desire a feature vector with high sensitivity to the environment variable but independent of the audio class (i.e. provides information about the environments without being affected by the audio class). Also, the output of this estimator should represent distance between the E training environments

and the surrounding environment. Let distance $d(e)$ represents the similarity between training environment e and the test environment; the smaller the distance the more the similarity between the training and test environment. The distance vector is as follows:

$$D = [d(1) \ d(2) \ \dots \ d(E)]^T, \quad d(e) \geq 0 \quad (5.3.1)$$

A detailed discussion of the environment estimators for the reverberant and noisy environments will be presented in Section 7.1 and Section 7.4 respectively.

5.4 Classifier Engines

Optimum performance of a classifier occurs when the test data from a particular environment are used with a classifier trained in this environment [11]. Also, the performance of a classifier trained in similar variable environments is better than those trained in environments that greatly differ.

5.4.1 Multi-Engine Classifier in Variable Environments

To implement a generalized robust ASC with high classification accuracy in variable environment, a Multi-Engine ASC classifier should be implemented to handle different environments (Figure 5.4.1). Using the environment information value from the environment estimator, the system fuses the outputs of the E ASC engines. Two fusions methods are introduced. The first method, the Multi-Engine Selection method, selects the ASC engine with training environments that best matches the surrounding environment and the system decision is equal to the decision associated with this single ASC engine. The second method, the Multi-Engine Weighted Fusion method, intelligently weights and

combines the output decisions of the E ASC engines; the higher the similarity of the testing environment to the engine's training environment, the higher the weighting that engine receives.

Figure 5.4.1 Solution for ASC in variable environment

5.4.1.1 Multi-Engine ASC

Classification algorithms such as GMM or HMM can be used to implement the individual ASC engines. During training, the parameters of each audio class model are determined from a sequence of M training vectors. During testing, the probability that a given audio class is captured (detected) is computed using the classification algorithms. Using extracted N feature vectors from the test audio signal, the e^{th} ASC engine computes a likelihood probability vector.

$$L_e = [l_e(1) \quad l_e(2) \quad \dots \quad l_e(S)]^T \quad (5.4.1)$$

where S is the number of elements in this vector which is equal to the numbers of the audio classes. The engine performs ASC by selecting the audio class as the audio class

with the highest likelihood value. Using a set of E engines, E likelihood probability vectors are produced. Using the environment information from the environment information estimator Multi-Engine fusion is performed.

5.4.1.2 Multi-Engine Selection

The distance $d(e)$ is computed for each training environment using the environment information estimator. The engine selection is calculated as follows:

$$\hat{e} = \arg \min_{1 \leq e \leq E} D \quad (5.4.2)$$

Figure 5.4.2 Multi-Engine Selection method

The ASC engine associated with the minimum distance is selected and the audio class determined by this engine is used as the ASC system output (Figure 5.4.2).

5.4.1.3 Multi-Engine Weighted Fusion

Performance of ASC engines varies in different environments. This performance variation depends on the similarity between the test and the training environment. Using this fact, ASC system can utilize the output decision of each engine and combine the decisions of these engines to achieve higher ASC performance (Figure 5.4.3).

Figure 5.4.3 Multi-Engine Weighted Fusion method

Assume that we have S audio classes C_i ($i \in \{1, 2, \dots, S\}$) and E ASC engines trained in E environments. The conditional probability for each audio class C_i generated from engine e is denoted as $p(C_i | env_e)$ and is computed using the following steps. Each audio class engine that is trained in environment e generates a maximum likelihood vector L_e as follows:

$$L_e = [l_e(1) \quad l_e(2) \quad \dots \quad l_e(S)]^T \quad (5.4.3)$$

where $l_e(i)$ is likelihood of the i^{th} audio class from the e^{th} ASC engine. These likelihoods are normalized as:

$$p(C_i | env_e) = \frac{l_e(i) - \min_j [l_e(j)]}{\sum_k (l_e(k) - \min_j [l_e(j)])} \quad (5.4.4)$$

This normalization ensures that the values of $p(C_i | env_e)$ have properties consistent with probabilities; that is:

$$\sum_{i=1}^S p(C_i | env_e) = 1, \quad p(C_i | env_e) \geq 0 \quad (5.4.5)$$

The probability that the surrounding environment is environment e is denoted as $p(env_e)$ and is computed as follows, first, the distance between the surrounding environment and the training environment is measured using the environment information estimator. Second, the distance vector is generated as in (Eq. 5.3.1). The training environment e with the smallest distance $d(e)$ value has the highest similarity with the surrounding environment. The distances are normalized to the probability of the surrounding environments as follows:

$$p(env_e) = \frac{d(e) - \max_j [d(j)]}{\sum_k (d(k) - \max_j [d(j)])} \quad (5.4.6)$$

This normalization ensures the values of $p(env_e)$ have properties consistent with probabilities; that is:

$$\sum_{e=1}^E p(env_e) = 1, \quad p(env_e) \geq 0 \quad (5.4.7)$$

The probability that audio class C_i is computed for the ASC system as follows:

$$p(C_i) = \sum_{e=1}^E p(C_i | env_e) p(env_e) \quad (5.4.8)$$

Finally, the decision of the Multi-Engine Weighted Fusion engines is the audio class with the maximum probability. The Multi-Engine Weighted Fusion method generalizes the solution for the ASC systems in a variable environment.

5.5 Discussion

In this chapter, a general framework was introduced to enhance the performance of the ASC in variable reverberant and noisy environments. The introduced framework adapts the feature extraction and the classification stages of the ASC system to the environment variables (Figure 1.3.1). As a result of this adaptation the performance of the ASC is enhanced.

Chapter 6

Feature Extraction in Variable Environments

In Chapter 5, we introduced the general frameworks to enhance the performance of the ASC in variable environments. New feature extraction and classification fusion methods were introduced, which have high performance in variable environments. Adaptation techniques for ASC systems in variable environments were also introduced.

In this chapter, we introduce two novel adaptive ASC features: Adaptive Pitch Ratio (APR) and Variance of Low-band Energy Ratio VLER, which have a resilient accuracy performance in variable and harsh environments (high reverberation and noise). This chapter utilizes Multi-Feature Selection and Multi-Feature Combination methods which were introduced in Chapter 5. These methods fuse the multi-feature decision by selecting or combining the weighted decision of each feature using the environment information to enhance the overall classification accuracy and robustness. In this chapter an ideal environment information estimator was assumed to avoid the confounding effect of an estimator.

6.1 Resilient and Adaptive Features

As discussed in Chapter 4 the performance of the ASC features classification accuracy degrades in variable environments. In this section, we introduce novel robust features with adaptive thresholds which have high classification accuracy in variable, low SNR, and reverberant environments.

6.1.1 Adaptive Pitch Ratio algorithm (APR)

The PR speech/non-speech classification algorithm (see Chapter 4) was originally introduced by the author [27]. In this section, we introduce further enhancements to this algorithm, which is capable of functioning in a low SNR and reverberant environment without significant degradation in performance. As a result of these refinements, the ASC system reliability and robustness are improved.

To discern speech from non-speech audio segments, the PR parameter is compared with certain threshold. Audio segments with a PR above this threshold would be classified as speech, while segments with a PR below this threshold would be classified as non-speech.

6.1.1.1 APR in Variable Reverberant Environment

For a reverberant environment, the pitch ratio threshold should be adapted to the reverberation level (RT60), to maximize the speech/non-speech classification accuracy. If a fixed PR threshold is used, the performance of the speech/non-speech classification will degrade because the pitch detection algorithm will fail to detect the presence of the

human pitch in a significant number of frames. To avoid failed detection of human pitch, the adaptive pitch ratio (APR) algorithm, appropriately changes the PR threshold based upon a reverberation estimate.

Figure 6.1.1 is a plot of the classification accuracy versus RT60, which illustrates the difference in performance, with and without an adaptive PR threshold. The APR algorithm is trained using half of the 1356 speech/non-speech database audio segments and tested with the other half using: $d = 1.00$ m (see Chapter 3). This result shows that the APR algorithm has slightly higher classification accuracy than the fixed threshold algorithm (PR) with in RT60 range of clean to 1.80 sec; average accuracy of 94.0% and 95.2% of fixed PR and APR algorithms, respectively. For $RT60 > 0.50$ sec the adaptive PR threshold has slight changes; ranges from 0.20 to 0.25. These results confirm that the fixed PR algorithm is resilient in the presence of the reverberation.

Figure 6.1.1 Speech/non-speech classification accuracy vs. RT60, using adaptive and fixed PR. ($FS = 60$ ms, $OL = 0.85$, and $SD = 450$ ms)

The performance of the APR algorithm was compared to the PR algorithm, LPC-based algorithm and the whole segment algorithm, with in RT60 range clean to 1.80 sec. The algorithms are trained using half of the 1356 speech/non-speech database audio segments and tested with the other half. As shown in Figure 6.1.2. the APR algorithm has the highest average classification accuracy in these variable reverberant environments, outperforming the fixed PR, LPC-based and whole segment algorithms. At RT60 = 0.68 sec APR exhibit accuracy performance of 96.0 %; higher accuracy of 2.5%, 9.5%, and 14.0% than fixed PR, whole segment, and LPC-based algorithms, respectively.

Figure 6.1.2 Comparison of Speech/Non-Speech classification accuracy using four algorithms in reverberant environment (%) (PR parameter $FS = 60$ ms, $OL = 0.85$)

6.1.1.2 APR in Variable Noisy Environment

At SNR greater than 12 dB, the speech/non-speech optimal PR threshold is a function of frame size, overlap ratio, and segment duration (e.g., with $FS = 60$ ms, $OL = 0.85$, and

$SD = 450$ ms the PR threshold = 0.32) (Figure 6.1.3). For SNR less than 12 dB, the PR threshold is also a function of the SNR and therefore should be adapted to the SNR in order to maintain the highest speech/non-speech classification accuracy.

Figure 6.1.3 Threshold and corresponding speech / non-speech classification accuracy vs. frame size and overlap (SNR = 30 dB, and $SD = 450$ ms)

At SNR less than 12 dB, the performance of the speech/non-speech classification is degraded because the pitch detection algorithm fails to detect the presence of the human pitch in a significant number of frames. To avoid failed detection of human pitch, the adaptive pitch ratio (APR) algorithm, appropriately reduces the PR threshold based upon an SNR estimate.

The speech/non-speech APR threshold is function of frame size, overlap ratio, segment duration, and SNR value. The value of the threshold is determined using lookup tables. These lookup tables are created using training audio segment library, with thresholds adjusted to maximize the speech/non-speech classification accuracy measured using the MCC. Figure 6.1.4 shows the threshold behavior at $OL = 0.85$, and $SD = 450$ ms is function of SNR and FS and the corresponding speech/non-speech classification accuracy.

Figure 6.1.4 Adaptive threshold and corresponding speech / non-speech classification accuracy vs. frame size and SNR ($OL = 0.85$, and $SD = 450$ ms)

Figure 6.1.5 Speech/non-speech classification accuracy vs. SNR, using adaptive and fixed PR. (FS = 60 ms, OL = 0.85, and SD = 450 ms)

For SNR values less than 12 dB, the adaptive nature of the APR algorithm enhances the performance of the fixed PR algorithm. The speech/non-speech using the APR algorithm is trained using half of the 1356 speech/non-speech database audio segments, using white noise, and tested with the other half over an SNR range of clean to -10 dB environments, using white and pink noise. The average performance plot using the white and pink noise of APR algorithm is shown in Figure 6.1.5. This plot illustrates the difference in performance, with and without an adaptive PR threshold. This plot shows that the APR algorithm has higher classification accuracy than the fixed threshold algorithm (PR), with a noticeably large enhancement for SNR values below 0 dB. There is an enhancement of 10% in the speech/non-speech classification accuracy for the -10 dB SNR level (fixed PR accuracy 50.0%; APR accuracy 60.0%), a 30% enhancement at the -5 dB SNR level (fixed PR accuracy 52%; APR accuracy 82.0%), and an 11.0% enhancement at the 0 dB SNR level (fixed PR accuracy 85%; APR accuracy 96.0%).

The performance of the APR algorithm was compared to the PR algorithm, LPC-based algorithm and the whole segment algorithm, with SNR values within range clean to -10 dB, using white and pink noise. As shown in Figure 6.1.6. APR algorithm has the highest classification accuracy in the whole SNR range, outperforming the PR, LPC-based, and whole segment algorithms.

Figure 6.1.6 Comparison of Speech/Non-Speech classification accuracy using four algorithms in noisy environment (%)

At SNR = -5 dB APR exhibit accuracy performance of 83.0%; higher accuracy of 29.0%, 31.0%, and 33.0% than whole segment, fixed PR, and LPC-based, respectively.

6.1.2 Variance of Low-Band Energy Ratio Algorithm (VLER)

A new energy-based feature, termed the variance of low-band energy ratio (VLER), is introduced to perform speech/music classification. This VLER feature produces high classification accuracy, relative to conventional features for audio classification, particularly at low SNR.

Variance of Low-band energy ratio (VLER) is the variance of the ratio of the low-band energy to the whole band energy vector, as shown in Figure 6.1.7

Figure 6.1.7 Variance of low-band energy ratio algorithm (VLER)

An input audio segment is segmented into N smaller frames of length s . The energy is calculated as follows:

$$Ein_i = \sum_{j=1}^s x_j^2 \quad (6.1.1)$$

where Ein_i is the input energy of frame i and x is the input audio sample.

Output energy $Eout_i$ is the energy of the filtered frame using a fourth order Butterworth low-pass filter with a cutoff frequency of 1500 Hz. Low-band energy ratio (LER) of the i^{th} frame is calculated as follows:

$$LER_i = \frac{Eout_i}{Ein_i} \quad 0 \leq LER_i \leq 1 \quad (6.1.2)$$

Finally, the VLER value of the N frames within the audio segment is calculated as follows:

$$VLER = \sum_{i=1}^N (LER_i - \mu)^2 \quad (6.1.3)$$

where μ is the mean of LER vector.

Speech contains two distinct signal types; voiced and unvoiced types. The frequency distributions of these types have different configurations. The voiced part has higher LER value than the unvoiced part. The different configuration is because the energy content at low-band energy (frequency < 1500 Hz) of the voiced type is relatively higher than the unvoiced type as shown in Figure 6.1.8. Typically for voiced speech ($\bar{a}$) LER = 0.92 and for unvoiced speech (sh) LER = 0.017. For music, the value of the LER remains relatively constant. Therefore, it is expected that speech would have higher VLER than music. The input segment is considered to be speech if the value of the VLER exceeds the threshold value; otherwise, it is considered music. VLER threshold is a function of the SNR and is adjusted to maximize the classification accuracy of speech/music classification.

Adaptation of the decision thresholds is conducted by estimating the threshold values that maximize the MCC of speech/music classification corresponding to different SNR values. Fixed threshold is calculated to maximize the MCC at high SNR value (SNR > 30 dB). Figure 6.1.9 shows that using adaptive threshold can enhance the classification accuracy of the algorithms, especially for SNR < 20 dB using speech/music.

Figure 6.1.8 Power spectral density of voiced (a) and unvoiced types (sh) (sampling rate 22.05 kHz)

Figure 6.1.9 Speech/music classification accuracy vs. SNR, using adaptive and fixed VLER

In Figure 6.1.9 the VLER is trained with 300 music and 300 speech one-second audio segments over SNR range of clean to -20 dB environments to determine the adaptive threshold. VLER is tested with 1320 music and 1320 speech one-second audio segments

over SNR range of clean to -20 dB environment, using white noise. VLER algorithm exhibits higher classification accuracy at the SNR range 30 to 0 dB than the SNR range clean to 30 dB. This emanates from the fact that the variance of the feature is used as the discerning value, not the feature itself. The addition of the noise to the speech signal affects the value of the unvoiced part more than the voiced part so the variance has a higher value; however, music audio class is not significantly affected, so the classification accuracy is increased until the noise dominates the energy of the signal.

Figure 6.1.10 is a plot of the speech/music classification accuracy versus the SNR using nine algorithms: Centroid, ZCR, Entropy, Low Energy Frames, 4Hz, PR, Segmentation count, Fixed VLER, and adaptive VLER. The speech/music classification algorithms are trained with 300 music and 300 speech one-second audio segments, using clean database except the adaptive VLER which has been trained over SNR range of clean to -20 dB environments to determine the adaptive threshold. The nine algorithms are tested with 1320 music and 1320 speech one-second audio segments over an SNR range of clean to -20 dB environment, using white noise.

Figure 6.1.10 shows that Adaptive VLER has the highest classification accuracy at $\text{SNR} < 20$ dB. Adaptive Feature Threshold can be generalized to accommodate the rest of the classification features. This threshold adaptation is conducted by estimating the threshold values that maximize the MCC of the speech/music classification accuracy at different SNR values.

Figure 6.1.10 Comparison of Speech/music classification accuracy using nine algorithms in noisy environment

6.1.3 Adaptive Feature Threshold in Noisy Environments

Noise affects the distribution of the audio classification features. By observing this distribution the decision threshold can be adapted to the SNR value of the noisy environment to maximize the classification accuracy as well as the robustness (Figure 6.1.11- Figure 6.1.15). From these plots it is shown that the thresholds should be adapted to the SNR values to achieve high classification accuracy. Also, it is shown that the best performing algorithm depends on the SNR environment; VLER has higher classification accuracy at SNR range 30 to -20 dB and 4Hz has higher classification accuracy at SNR > 30 dB.

Figure 6.1.11 (a) 4Hz (b) VLER algorithms distribution at SNR = -20

Figure 6.1.12 (a) 4Hz (b) VLER algorithms distribution at SNR = -10 dB

Figure 6.1.13 (a) 4Hz (b) VLER algorithms distribution at SNR = 0 dB

Figure 6.1.14 (a) 4Hz (b) VLER algorithms distribution at SNR = 10 dB

Figure 6.1.15 (a) 4Hz (b) VLER algorithms distribution at clean environment

Assuming that we have a perfect SNR estimator, using the same adaptation method of the decision threshold, applied to other features, the classification accuracy should be enhanced (Figure 6.1.16 to Figure 6.1.18).

Figure 6.1.16 Speech/music classification accuracy vs. SNR, using adaptive and fixed Entropy (left) and Segmentation (right)

Figure 6.1.17 Speech/music classification accuracy vs. SNR, using adaptive and fixed APR (left) and Centroid (right)

Figure 6.1.18 Speech/music classification accuracy vs. SNR, using adaptive and fixed Low Energy Frames (left) and ZCR (right)

The speech/music classification algorithms are trained with 300 music and 300 speech one-second audio segments over SNR range of clean to -20 dB environments to determine the adaptive threshold. These algorithms are tested with 1320 music and 1320 speech one-second audio segments over an SNR range of clean to -20 dB environment, using white noise. Figure 6.1.19 reconfirms that using the threshold adaptation method enhances the classification accuracy of the speech/music classification algorithms.

Figure 6.1.19 also shows the classification accuracy gain, which is the difference between the classification accuracy using adaptive and fixed threshold. As shown in Figure 6.1.19, at SNR = -15 dB VLER has accuracy gain of 22.0%. Centroid achieved accuracy gain of 40.0% at SNR = 0 dB. From the simulation it is shown that VLER, Centroid, PR, and entropy have high gains at SNR < 10 dB, also 4Hz and segmentation have high gains at SNR > 10 dB. Using these accuracies of different features, Multi-

Feature Selection or Multi-Feature Combination system can be constructed with further improvements in classification accuracy (section 6.2).

Figure 6.1.19 Classification accuracy of adaptive threshold and classification accuracy gain of adaptive threshold over fixed threshold classification vs. SNR

Classification accuracy is also affected by the segment duration. The segment duration used in the previous simulations is 1 second. In Figure 6.1.20 the effect of the segment duration on the classification accuracy shows that the classification accuracy increases with the segment duration.

Figure 6.1.20 Adaptive threshold classification accuracy vs. segment duration

6.2 Multi-Feature Methods

The previous section has shown that by adapting the feature threshold to the environment variable an enhancement of the classification accuracy can be achieved. Also it has been shown that different algorithms have different classification accuracy in different environments. Using these new findings Multi-Feature methods have been developed: Multi-Feature Selection and Multi-Feature Combinations.

6.2.1 Multi-Feature Selection Method

A block diagram of the Multi-Feature Selection method is discussed and shown in Figure 5.2.5. In this chapter it is already mentioned that a perfect SNR estimator was assumed to avoid the confounding effect of an estimator. From training, the best performing feature is selected according to the SNR value.

In Figure 6.2.1, the speech/music classification algorithms are trained with 300 music and 300 speech one-second audio segments over SNR range of clean to -20 dB environments to determine the adaptive threshold, using white noise. Also the speech/music algorithms are tested with 1320 music and 1320 speech one-second audio segments over an SNR range of clean to -20 dB environment, using white noise.

Multi-Feature Selection method improves the overall classification accuracy over the single feature classification. Classification accuracy matches the highest single feature classification. The classification accuracy of this selection method depends on the assumption of the perfect SNR estimator. In reality the SNR estimator has certain tolerance which affects the performance of the selection system. A robust Multi-Feature system can further improve classification accuracy by combining the different features decisions.

Figure 6.2.1 Classification accuracy using best-performing feature selection algorithm vs. SNR (dB)

6.2.2 Multi-Feature Combination Method

Further improvements in classification accuracy are obtained using a Multi-Feature Combination method (Section 5.2.2.3). In such a system, there are multiple features, where each feature is used to perform speech/music classification. As shown in section 5.2.2.3 the Multi-Feature Combination method decision is as follows:

$$E(h) = \arg \left[\max_s \left[Q_M^w(s) \right] \right] \quad (6.2.1)$$

where

$$Q_M^w(s) = \sum_{t=1}^T w_{SNR,t,M} q_t(s) \quad (6.2.2)$$

where q_t is Borda count, $w_{SNR,t,M}$ is the weight of the feature t at certain SNR , using the best-performing M features out of T features. q_t can be calculated using Eq. 5.2.6.

To create the weight vector $W_{SNR} = [w_{SNR,1} \ w_{SNR,2} \ w_{SNR,3} \ \dots \ w_{SNR,T}]$, first the classification accuracy vector CL_{SNR} is calculated as follows:

$$CL_{SNR} = [cl_{SNR}(1) \ cl_{SNR}(2) \ \dots \ cl_{SNR}(T)]^T \quad (6.2.3)$$

where $cl_{SNR}(i)$ is the classification accuracy of the i^{th} feature at certain SNR . These accuracies are normalized as follow:

$$w_{SNR,t} = \frac{cl_{SNR}(t) - \min_j [cl_{SNR}(j)]}{\sum_k (cl_{SNR}(k) - \min_j [cl_{SNR}(j)])} \quad (6.2.4)$$

This normalization ensures the values of $w_{SNR,t}$ have properties consistent with probabilities; that is:

$$\sum_{t=1}^T w_{SNR,t} = 1, \quad w_{SNR,t} \geq 0 \quad (6.2.5)$$

The modified weight vector $w_{SNR,t,M}$ is equal to the weight vector $w_{SNR,t}$ with zeroing out the contribution of all experts except the best-performing M features, $M \leq T$.

Figure 6.2.2 shows the speech/music classification accuracy using Multi-Feature Combination method using $M = 3$ and $M = 8$. Classification algorithms are trained with 300 music and 300 speech one-second audio segments over SNR range of clean to -20 dB

environments to determine the adaptive threshold and the weight vector. These algorithms are tested with 1320 music and 1320 speech one-second audio segments over an SNR range of clean to -20 dB environment, using white noise. The Classification accuracy matches or below the highest single feature classification by 2.0% at $-20 \text{ dB} < \text{SNR} < 10 \text{ dB}$. For $\text{SNR} > 10 \text{ dB}$, the classification accuracy is superior to the single feature classification accuracy (at $\text{SNR} = 30$, the highest single classification accuracy = 91.0%; Multi-Feature Combination = 94.0%). Multi-Feature Selection can be assumed to be as a special case of the Multi-Feature Combination method with $M = 1$.

Figure 6.2.2 Classification accuracy using Multi-Feature Combination algorithm vs. SNR (dB)

6.3 Discussion

Two novel adaptive ASC features: Adaptive Pitch Ratio (APR) for speech/non-speech classification and Variance of Low-band Energy Ratio (VLER) for speech music

classification, which have a resilient accuracy performance in variable and harsh environments (low SNR and high reverberation) have been introduced.

Two methods were introduced to perform Multi-Feature ASC; Multi-Feature Selection method and Multi-Feature Combination method. Multi-Feature Combination method has classification accuracy matches or below the highest single feature classification by 2.0% at $-20 \text{ dB} < \text{SNR} < 10 \text{ dB}$. For $\text{SNR} > 10 \text{ dB}$, the classification accuracy of the Multi-Feature Combinations is superior to the individual feature classification accuracy. Naturally, the Multi-Feature Selection method cannot perform better than all individual ASC features. The Multi-Feature Selection is simpler than the Multi-Feature Combination. Meanwhile compared to the Multi-feature Selection the Multi-Feature Combination is more generalizable solution at unknown test environment. This generalizable solution is achieved because the Multi-Feature Combination method does not just select one feature associated with a certain environment; rather, it weights and combines all output decisions of the ASC features using environment information. The Multi-Feature Selection method is better solution, when the expected test environments are similar to one of the training environments, as it has lower complexity.

In general it has been shown that using multi-feature algorithms; Multi-Feature Selection and Multi-Future Combination, can enhance the performance of the ASC.

Chapter 7

Environment Information Estimators with ASC

In Chapter 6, we introduced novel adaptive ASC features which have a resilient accuracy performance in variable and harsh environments (low SNR and high reverberation). A new generalized adaptation method for the ASC features is also developed. These methods select or combine the weighted decision of each feature using the environment information to enhance the overall classification performance. A perfect environment information estimator was assumed to avoid the confounding effect of an estimator.

In this chapter, we introduce environment information estimators' methods: Reverberation Sensing System and Discrete-Valued SNR Estimator to estimate reverberation (RT60) and noise (SNR), respectively. Using this estimated RT60 and SNR, a Multi-Engine ASC system is implemented. As described in Chapter 5, two methods are implemented to perform Multi-Engine ASC: Multi-Engine Selection method and Multi-Engine Weighted Fusion method. These methods are evaluated in simulated reverberant, noisy, hybrid noisy and reverberant, and real environments.

7.1 Reverberation Sensing System (RSS)

The reverberation sensing system (RSS) is an environment information estimator that provides information about the reverberation level in the surrounding environment. This information is provided by extracting features from the captured audio sounds emanating within this environment.

A conceptually simple and computationally efficient method is introduced to function as the RSS. The RSS is implemented using covariance matrix modeling [88]-[89]. The feature vector of the reverberant environment is comprised MFCC and delta MFCC, computed using the filter bank output without the use of the discrete cosine transform (DCT); this feature vector is known as Mel frequency spectrum coefficient (MFSC). The DCT of the MFCC vector de-correlates the feature vectors' elements, which leads to a feature vector with low sensitivity to the change in the environment (in this section it is the reverberation) and high sensitivity to the audio class (i.e. dependent on the audio class) [11]. In our case we desire a feature vector with high sensitivity to the reverberation but independent of the audio class (i.e. provides information about the environments without being affected by audio class source).

7.1.1 RSS Implementation

To implement the RSS, let X_e represent the covariance matrix of the training feature vectors for environment e $\{\vec{x}_i, 1 \leq i \leq M\}$ and let Y represent the covariance matrix of the test feature vectors $\{\vec{y}_j, 1 < j < N\}$ from an unknown environment. Feature vectors for

both training and testing covariance matrices, have a length L ; therefore, X_e and Y are matrices of size $L \times L$ and are computed as follows:

$$X_e = \frac{1}{M} \sum_{i=1}^M (\vec{x}_i - \bar{x})^T (\vec{x}_i - \bar{x}) \quad (7.1.1)$$

$$Y = \frac{1}{N} \sum_{j=1}^N (\vec{y}_j - \bar{y})^T (\vec{y}_j - \bar{y}) \quad (7.1.2)$$

where vector $\bar{x}$ is the mean of the M training $\vec{x}_i$ vectors and vector $\bar{y}$ is the mean of the N test $\vec{y}_j$ vectors. Using the following property [89]:

$$\frac{1}{N} \sum_{j=1}^N (\vec{y}_j - \bar{y})^T X_e^{-1} (\vec{y}_j - \bar{y}) = \text{tr}(Y X_e^{-1}) \quad (7.1.3)$$

The distance between training environment e and test environment is calculated using the arithmetic harmonic sphericity measure as follows [89]:

$$d(e) = \log[\text{tr}(Y X_e^{-1}) \text{tr}(X_e Y^{-1})] - 2 \log(L), \quad 1 \leq e \leq E \quad (7.1.4)$$

where L is the feature vector length. The distance $d(e)$ represents the similarity between training environment e and the test environment; the smaller the distance the more the similarity between the training and test environment. The distances between the E training environments and the test environment are calculated to generate the distance vector as follows:

$$D = [d(1) \quad d(2) \quad \dots \quad d(E)]^T, \quad d(e) \geq 0 \quad (7.1.5)$$

The test environment is matched with one of the E training environments based on the minimum distance:

$$\hat{e} = \arg \min_{1 \leq e \leq E} D \quad (7.1.6)$$

The KING database was utilized to evaluate the RSS accuracy performance. The RSS was trained using a randomly selected talker. The 3 utterances in the training set, associated with this talker, were filtered with 5 impulse responses, simulating 5 reverberant environments in addition to a non-reverberant environment. The impulse responses have RT60 values of 0.50, 0.85, 1.15, 1.35, and 1.65 sec and the distance between the audio source and audio capturing device $d = 1.00$ m. RSS training and testing are conducted using a feature vector of 19 MFSCs and delta MFSCs. The minimum distance is used to classify the test audio to one of the training environments.

Using the same testing database configuration as in Section 4.1.3, test utterances are unfiltered (RT60($j = 1$) is clean environment) or filtered using 33 new impulse responses associated with reverberant environments. Eleven RT60 values were used (RT60(j) $\in$ {0.25, 0.50, 0.68, 0.85, 1.00, 1.15, 1.25, 1.35, 1.50, 1.65, and 1.80 sec}; $j = 2, 3, \dots, 12$), with three different d ($d = 0.75, 0.87$ and 1.25 m) setting for each RT60 value to create the reverberated speech. The test environments have different setup than the training environments; even if the RT60 value is the same for the test and training environments (i.e. each room has different four sets values of the image method) (Table B.1.1-

Table B.1.16).

Table 7.1.1 summarizes the results of the RSS classification (reverberant environment classification). The total number of utterances for j^{th} RT60 value in the test set is equal to U_j , where $U_j = 347$ for $j = 1$ (clean environment) and $U_j = 3 \times 347 = 1041$ for $j \neq 1$ (reverberant environments). The elements of Table 7.1.1, Q_{ji} are the percentage of the total number of utterances U_j that were classified to the i^{th} training RT60 value.

Table 7.1.1 RSS classification results (%)

Test Env. \ Training Env.	Clean	0.50	0.85	1.15	1.35	1.65
Clean	93.4	5.5	1.1	0.0	0.0	0.0
0.25	2.3	69.0	22.0	3.7	0.0	3.0
0.50	0.0	55.7	32.8	11.0	0.5	0.0
0.68	0.0	12.6	71.0	3.2	0.8	12.4
0.85	0.0	0.3	90.0	3.0	3.7	3.0
1.00	0.0	0.3	43.0	35.0	7.5	14.2
1.15	0.0	0.0	20.5	44.7	14.8	20.0
1.25	0.0	0.3	8.2	38.7	27.2	25.6
1.35	0.0	0.3	0.6	6.3	43.5	49.3
1.50	0.0	0.0	0.3	0.3	30.3	69.1
1.65	0.0	0.0	0.6	4.5	17.5	77.4
1.80	0.0	0.0	0.0	3.0	23.0	74.0

The RSS output decision is considered to be correct if the RT60 of the selected training environment has the closest value to the RT60 of the surrounding test environment or one of the adjacent environments in terms of RT60 (these are highlighted in grey in Table 7.1.1). For example, if the surrounding environment has an RT60 = 0.68, then the output decision is considered to be correct if the selected RT60 = 0.50 or 0.85 sec (accuracy = 12.6% + 71.0% = 83.6%). In addition, if the surrounding environment has RT60 = 1.15 then the output decision is considered to be correct if the selected RT60 = 0.85, 1.15 or 1.35 sec (accuracy = 20.5% + 44.7% + 14.8 = 80.0%).

As noted in [13], the RT60 is not the only factor that describes the reverberation (e.g. the reverberation density also influences the overall reverberation). As such, the RT60 classification is not expected to be precise, which justifies the inclusion of the adjacent RT60 values in the definition of a correct decision. Overall, the average accuracy of the RSS is 87.5%.

7.2 Multi-Engine Selection Method in Reverberant Environment

Gaussian mixed models (GMM) are used to implement the individual ASC engines. During training, the parameters of each audio class GMM are determined from a sequence of M training vectors. Using extracted N feature vectors from the test utterance, the ASC engine which trained at environment e computes a log-likelihood probability vector.

$$L_e = [l_e(1) \quad l_e(2) \quad \dots \quad l_e(S)]^T \quad (7.2.1)$$

where S is the number of elements in this vector equal to the number of the audio classes. In the following simulations TI is implemented as a category of the ASC. The engine performs TI by selecting the talker that uttered the test speech as the talker with the highest log-likelihood value.

Six TI engines ($E=6$) are trained using training set filtered with the impulse response of the training environments used in Section 7.1. A 32-GMM is used as TI engine, with a feature vector of 16 MFCCs.

Using the same testing database configuration as in Section 7.1, Figure 7.2.1 plots the classification accuracy results of the 6 individual TI engines. The maximum performance of a single TI occurs when the test data from a particular reverberant environment is used with the TI engine trained in that environment. Also, the performance of TI engines trained in similar reverberant environments is better than those trained in environments that greatly differ.

Figure 7.2.1 Talker identification accuracy using Multi-Engine Selection method vs. RT60

The RSS is utilized to select the best performing TI engine for given environments using the RT60 value for Multi-Engine Selection method. Figure 7.2.1 shows that this Multi-Engine Selection method achieves an average identification accuracy of 95.4%, with the lowest accuracy being 93.0%; whereas the identification accuracy using the TI engine trained in a non-reverberant environment only achieves an average accuracy of 86.5%, with the lowest accuracy being 76.1%.

RSS at reverberant environment $RT60 = 1.15$ has an environment classification accuracy of 80.0%. This indicates a similarity between this environment and other environments with different $RT60$ s. Incorrect environment classification by the RSS, however, does not necessarily translate into a severe degradation of the overall TI system accuracy. Indeed, for the reverberant environment $RT60 = 1.15$, the incorrectly selected TI engine with training $RT60 = 1.65$ (selected 20.0% of the time) still achieved an identification accuracy of 94.0%.

7.3 Multi-Engine Weighted Fusion Method in Reverberant Environment

The 6 TI engines and database have the same configuration as Section 7.1. Figure 7.3.1 plots the Multi-Engine Weighted Fusion TI engines' performance. RSS is utilized to weight and combine the TI decision of the 6 TI engines. Figure 7.3.1 shows that this Multi-Engine Weighted Fusion method achieves an average identification accuracy of 96.5%, with the lowest accuracy being 95.0%. The identification accuracy of the Multi-Engine Weighted Fusion method outperforms Multi-Engine Selection method (95.4%) and the identification accuracy using the TI engine trained in a non-reverberant (86.5%).

Figure 7.3.1 Talker identification accuracy using Multi-Engine Weighted Fusion method vs. RT60

For $RT60 > 0.50$ sec, the classification accuracy of the Multi-Engine Weighted Fusion TI engines is superior to all individual TI engines. For example, at $RT60 = 1.35$, the engine trained at $RT60 = 1.35$ has an accuracy of 94.0%, the Multi-Engine Selection classification accuracy also has an accuracy of 94.0%, and the Multi-Engine Weighted Fusion has an accuracy of 96.5%. Naturally, the Multi-Engine Selection method cannot perform better than all individual TI engines.

The Multi-Engine Weighted Fusion method has higher complexity than the Multi-Engine Selection method; albeit, only slightly higher complexity associated with Eq. (5.4.4), (5.4.6), and (5.4.8). Meanwhile, the TI Multi-Engine Weighted Fusion method has higher classification accuracy and reliability (lowest classification accuracy of Multi-Engine Weighted Fusion = 95.0% and Multi-Engine Selection = 93%). Also, the Multi-Engine Weighted Fusion is a more generalized solution for ASC systems in variable reverberant environments. This includes TI for environments with different RT60 values

than the training environments; in these tests these are $RT60 = 0.25, 0.68, 1.00, 1.25, 1.50$, and 1.80 sec, where the average accuracy of the Multi-Engine Weighted Fusion method is 96.6%; an enhancement of 2.0% over the Multi-Engine Selection method.

Variable environment with a known $RT60$ set (e.g. TI application at environments with $RT60$ similar to the training environments; $RT60 = \text{clean}$, and $0.50, 0.85, 1.15, 1.35$, and 1.65 sec) Multi-Engine Selection is better solution as it has lower complexity. The Multi-Engine Weighted Fusion method at environments with similar $RT60$ to the training environments is 96.4%, which is an enhancement of only 0.4% over the Multi-Engine Selection method.

For low reverberation environments ($RT60 \leq 0.50$ sec), it seems there is no benefit for a Multi-Engine system. Indeed the engine trained at $RT60 = 0.50$ outperformed the Multi-Engine systems (Figure 7.3.1); however, it is clearly not robust as there is severe degradation in performance for larger level of reverberation (at $RT60 = 1.80$ sec classification accuracy of engine trained at $RT60 = 0.50$ and Multi-Engine Weighted Fusion are 87.0% and 95.0%, respectively).

7.4 Discrete-Valued SNR Estimator

In this section, a discrete-valued SNR estimator is introduced. This estimator provides information about the SNR in a discrete manner according to six SNR training environments (Clean environment and $SNR = 30, 20, 10, 0$ and -10 dB).

As in the RSS the discrete-valued SNR estimator is implemented using covariance matrix modeling. Training and testing are conducted using a feature vector of 19 MFSCs and delta MFSCs using the exact equations set (eq. 7.1.1 to eq. 7.1.6). The estimated $\widehat{SNR}$ value is the SNR of the selected training environment $\{\widehat{SNR} = SNR_{\hat{e}}, 1 \leq \hat{e} \leq E\}$

The KING database was utilized to evaluate the Discrete-Valued SNR Estimator accuracy performance. The Discrete-Valued SNR Estimator was trained using a randomly selected talker. Six different SNR values were associated with the 3 utterances in the training set, associated with this talker, simulating 5 noisy environments; using White noise, in addition to a clean environment. The noisy environments have SNR values (-10, 0, 10, 20, and 30 dB and a clean environment).

Performance of the Discrete-Valued SNR Estimator is evaluated on the test set; each utterance is 15 second in duration. Test utterances, used ten SNR values ($SNR(j=1)$ is clean environment and $SNR(j) \in \{35, 30, 25, 20, 15, 10, 5, 0, -15, -10\}; j=2, 3, \dots, 11$), with two different noise types; white and pink noise, to create the noisy speech. A total of 20 test noisy environments is used, in addition to the clean environment.

Table 7.4.1 summarizes the results of the Discrete-valued SNR classification (environment classification). The total number of utterances for j^{th} SNR value in the test set is equal to U_j , where $U_j = 347$ for $j = 1$ (clean environment) and $U_j = 2 \times 347 = 694$ for $j \neq 1$ (noisy environments). The elements of Table 7.4.1 Q_{ji} are the percentage of the total number of utterances U_j that are classified to the i^{th} training SNR value.

Table 7.4.1 Discrete-valued SNR estimator classification results (%)

Test Env. \ Training Env.	Clean	30	20	10	0	-10
clean	100.0	0.0	0.0	0.0	0.0	0.0
35	67.1	32.9	0.0	0.0	0.0	0.0
30	34.0	64.5	1.5	0.0	0.0	0.0
25	10.1	70.0	19.9	0.0	0.0	0.0
20	0.0	19.5	78	2.5	0.0	0.0
15	0.0	7.2	62.1	30.7	0.0	0.0
10	0.0	0.0	18.6	78.5	2.9	0.0
5	0.0	0.0	0.0	29.8	70.2	0.0
0	0.0	0.0	0.0	0.0	97.1	2.9
-5	0.0	0.0	0.0	0.0	2.9	97.1
-10	0.0	0.0	0.0	0.0	0.0	100

The Discrete-Valued SNR Estimator output decision is assumed to be correct if the SNR of the selected training environment has the closest value to the SNR of the surrounding test environment or one of the adjacent environments in terms of SNR (these are highlighted in grey in Table 7.4.1. For example, if the surrounding environment has an SNR = 15 dB, then the output decision is assumed to be correct if the selected SNR = 10 or 20 dB (accuracy = 30.7% + 62.1% = 92.8%).

The SNR classification is not expected to be precise, which justifies the inclusion of the adjacent SNR values in the definition of a correct decision. Overall, the average accuracy of the Discrete-Valued SNR Estimator is 98.4% overall accuracy.

7.5 Multi-Engine Selection Method in Noisy Environment

Six TI engines are trained using the same training set used in the section 7.4. A 32-GMM is used as classifier, with a feature vector of 16 MFCCs.

Figure 7.5.1 plots the results of the 6 individual TI engines and Multi-Engine Selection. The maximum performance of a single TI engine occurs when the test data from a particular noisy environment is used with the TI engine trained in that environment. Also, the performance of TI engines trained in similar noise environments is better than those trained in environments that greatly differ.

Figure 7.5.1 Talker identification accuracy using Multi-Engine Selection method vs. SNR

The Discrete-Valued SNR Estimator is utilized to select the best performing TI engine for given environments using the estimated SNR value. Figure 7.5.1 shows that the Multi-Engine Selection method has average classification accuracy of 88.7 % in noisy environments over an SNR range of clean environment to 10 dB. Compared to the identification accuracy using TI engine trained in a clean environment the improvement in accuracy was as high as 30.6%. Also the Engine Selection achieved an improvement in accuracy as high as 17.0% over an SNR range of 10 to -10 dB.

The Multi-Engine Selection maintains accuracy above 75.0% for SNR above 10 dB, whereas the clean TI engine only does so for SNR above 30 dB.

7.6 Multi-Engine Weighted Fusion Method in Noisy Environment

Six TI engines are trained using the same training set used in the section 7.4. A 32-GMM is used as classifier, with a feature vector of 16 MFCCs.

Figure 7.6.1 plots the Multi-Engine Weighted Fusion performance. Discrete-Valued SNR Estimator is utilized to weight and combine the TI decision of the 6 TI engines. Figure 7.6.1 shows that the Multi-Engine Weighted Fusion method has average classification accuracy of 91.3 % in noisy environments over an SNR range of clean to 10 dB. The identification accuracy of the Multi-Engine Weighted Fusion method outperforms the Multi-Engine Selection method (88.7%) and the identification accuracy using the TI engine trained in a clean environment (58.1%).

Figure 7.6.1 Talker identification accuracy using Multi-Engine Weighted Fusion method vs. SNR

For $\text{SNR} > 5$ dB, the classification accuracy of the Multi-Engine Weighted Fusion is superior to all individual TI engines. For example, at $\text{SNR} = 15$ dB, the engine trained at $\text{SNR} = 10$ and 20 dB have an accuracy of 68.0% and 87.0%, respectively, the Multi-Engine Selection has an accuracy of 81.0%, and the Multi-Engine Weighted Fusion has an accuracy of 89.0%. Naturally, the Multi-Engine Selection method cannot perform better than all individual TI engines.

The Multi-Engine Weighted Fusion is a more generalized solution for TI systems in variable noisy environments. This includes TI for environments with different SNR values than the training environments; $\text{SNR} = 35, 25, 15, 5$, and -5 dB, where the classification performance enhancement is 7.2% over the Multi-Engine Selection method.

For variable environments with a known SNR set (e.g. TI application at environments with SNR similar to the training environments; $\text{SNR} = 30, 20, 10, 0$, and -10 dB, in addition to clean environment) the Multi-Engine Selection method has almost similar accuracy as the Multi-Engine Weighted Fusion method. This makes the Multi-Engine Selection method a better solution, when the expected testing environments are similar to one of the training environments, as it has lower complexity.

For high SNR environments ($\text{SNR} \geq 30$ dB), it seems there is no benefit for a multi-engine system. Indeed the engine trained at $\text{SNR} = 30$ dB has almost similar accuracy as the Multi-Engine systems (Figure 7.6.1); however, it is clearly not robust as there is severe degradation in performance for lower level of SNR.

7.7 Multi-Engine Selection and Weighted Fusion Methods in Real Environments

The KING database was utilized to evaluate the classification accuracy of the Multi-Engine Selection and the Multi-Engine Weighted Fusion methods at real environments. The RSS is trained using a randomly selected talker. The 3 utterances in the training set, associated with this talker, were filtered with 5 impulse responses, simulating 5 reverberant environments in addition to a clean environment. The impulse responses have RT60 values of 0.50, 0.85, 1.15, 1.35, and 1.65 sec and $d = 1.00$ m. RSS training and testing are conducted using a feature vector of 19 MFSCs and delta MFSCs.

Six TI engines ($E=6$) are trained using the same training set used in RSS training. A 32-GMM is used as classifier, with a feature vector of 16 MFCCs.

Real environment experiments are placed at two rooms at Carleton University, Mackenzie Building, Rooms ME 4356 and ME 4359. A Dell model A-215 speaker is used to play the KING database speech files. A Mitel prototype circular microphone array is used to capture the emanating speech within the real environment. The captured signal is acquired and pre-amplified using the PreSonus Audio Electronics recording system, FirePodTM. The pre-amplified signal is recorded using a Toshiba laptop (Figure 7.7.1). The speech database was played and recorded in both rooms: ME 4356 and ME 4359. The rooms' specifications are illustrated in Figure 7.7.2.

Figure 7.7.1 Real Environment setup (ME4356)

Figure 7.7.2: Illustration of the real environment setups (a) ME 4359 (b) ME 4356.

ME 4359 has a noisy central air duct; SNR = 12 dB, which increases the noise floor than ME 4356; SNR = 21 dB. The RT60 of ME 4359 is 0.78 sec and ME 4356 is 0.47 sec. ME 4359 has higher RT60 than ME 4356 due to its larger volume; the volume of ME 4359 = 230.8 m^3 and the volume of ME 4356 = 116.8 m^3 , and its higher reflection coefficients; the vinyl tile has higher reflection coefficient than the carpet.

The RSS is trained using the same configuration as in Section 7.1. The TI engines are trained using the same configuration as in Section 7.2.

Using the clean KING database, three experiments are conducted in three different real environments. Two of these experiments are conducted in ME 4359: first experiment is conducted with setup similar to Figure 7.7.2.a. and second experiment is conducted with the cooling fan of the laptop 10cm apart from the microphone array; SNR = 8 dB. One experiment is conducted in ME 4356 with setup similar to Figure 7.7.2.b. The RSS selection of the individual engines and the classification accuracy of these engines are shown in Table 7.7.1.

Table 7.7.1 Real Environment RSS selection and classification accuracy of the individual TI engine

Test Env. \ Training Env. (RT60)		Clean	0.50	0.85	1.15	1.35	1.65
ME 4356 (RT60 = 0.47 sec, SNR = 21 dB)	RSS Selection	0	47.5%	36.6%	13.7%	1.4%	0.8%
	Accuracy	36.6%	75.4%	79.1%	74.5%	60.3%	61.6%
ME 4359 (RT60 = 0.78 sec, SNR = 12 dB)	RSS Selection	0	3.4%	54.0%	36.2%	5.8%	0.6%
	Accuracy	19.6%	67.2%	75.7%	72.6%	69.8%	68.5%
ME 4359 With Laptop Fan (RT60 = 0.78 sec, SNR = 8 dB)	RSS Selection	0	0	4.8%	93.8%	1.4%	0
	Accuracy	14.2%	50.1%	60.2%	55.7%	57.5%	55.1%

Table 7.7.1 summarizes the results of the RSS classification of the real test environments (reverberant environment classification) and the corresponding classification accuracy of the individual TI engines. For the RSS selection, the total number of utterances for j^{th} RT60 value is equal to U_j , where $U_j = 347$. The elements of Table 7.7.1, Q_{ji} are the percentage of the total number of utterances U_j that were classified to the i^{th} training RT60 value. The accuracy in Table 7.7.1 is the classification

accuracy of the individual TI engine for each training environment in the three experimental environments.

RSS classified the ME 4356 as environment with $RT60 = 0.5$ sec for 47.5% of the time; the measured $RT60 = 0.47$ sec. RSS classified the ME 4359 as environment with $RT60 = 0.85$ sec for 54.0% of the time; the measured $RT60 = 0.78$ sec. These results prove that the RSS selects most of the time the closest $RT60$ to the real environment. Also the classification accuracies of selected individual TI engines have high classification accuracy, compared to the other individual TI engines.

RSS classified the ME 4359 with the laptop fan, as environment with $RT60 = 1.15$ sec for 93.8% of the time. This result show that the selected environment is the adjacent environment of the nearest environment to the measured $RT60$; $RT60 = 0.78$ sec and the nearest environment $RT60 = 0.85$ sec. This skew of the selected $RT60$ is due to the added noise from the laptop fan. The selected $RT60$ shows that the RSS consider the reverberant environment with added noise as a reverberant environment with higher $RT60$; the selected is $RT60 = 1.15$ sec and measured $RT60 = 0.78$ sec.

Table 7.7.2 shows that, in ME 4356 the Multi-Engine Weighted Fusion has classification accuracy of 79.2%. Compared to clean engine method the improvement in accuracy was 42.6%. In ME 4359 the Multi-Engine Weighted Fusion maintained a good degree of robustness with classification accuracy of 75.6 %. Compared to clean engine method the improvement in accuracy was as high as 56.0%. These results prove that the Multi-Engine Weighted Fusion has robust high classification accuracy.

Table 7.7.2 Talker identification accuracy using different methods in real environments

	ASC Algorithm		
	Clean	Multi-Engine Selection	Multi-Engine Weighted Fusion
ME 4356	36.6%	71.5%	79.2%
ME 4359	19.6%	68.2%	75.6%
ME 4359 With Laptop fan	14.2%	55.1%	61.3%
Average	23.4%	65.0%	72.0%

In ME 4359 with laptop fan, the surrounding environment is reverberant environment with high noise level; SNR = 8 dB. Due to this hybrid noisy and reverberant environment the performance of the Multi-Engine Weighted fusion is degraded (without the fan 75.6% and with the fan 61.3%). Compared to clean engine method the improvement in accuracy was 47.1%.

The Multi-Engine Weighted Fusion method achieved average classification accuracy of 72.0%, outperforms Multi-Engine Selection method (65.0%) and the identification accuracy using the TI engine trained in a clean environment (23.4%).

These results reconfirm that Multi-Engine Weighted Fusion method has higher classification accuracy than Multi-Engine Selection if the surrounding environment is not one of the training environments.

7.8 Multi-Engine Fusion Methods at Hybrid Reverberant and Noisy Environments

The KING database was utilized to evaluate the classification accuracy of Multi-Engine Selection and Multi-Engine Weighted Fusion methods at hybrid reverberant and noisy environments.

The Discrete-Valued SNR Estimator was trained using a randomly selected talker. Six different SNR values were associated with the 3 utterances in the training set, associated with this talker, simulating 5 noisy environments in addition to a clean environment. The noisy environments have SNR values (30, 20, 10, 0, -10 dB and a clean environment).

Performance of the Multi-Engine Selection and the Multi-Engine Weighted Fusion using Discrete-Valued SNR Estimator is evaluated on the test set; each utterance is 15 sec in duration. Four sets of test utterances are used. These sets, are unfiltered ($RT60(j) = 1$) is non-reverberant) or filtered using 3 impulse responses associated with reverberant environments. Three RT60 values were used ($RT60(j) \in \{0.50, 1.15, \text{ and } 1.80 \text{ sec}\}; j = 2, 3, 4$), with $d = 1.00$ m. White noise is added to the four reverberant sets to create hybrid reverberant and noisy test database. Ten white noise are added to the unfiltered and filtered database to create different SNR values ($SNR(i = 1)$ is clean environment and $SNR(i) \in \{35, 30, 25, 20, 15, 10, 5, 0, -15, -10\}; i = 2, 3, \dots, 11$). A total of 44 hybrid reverberant and noisy environments are used.

Six TI engines ($E=6$) are trained using training set similar to the training set used in the Discrete-Valued SNR Estimator training. A 32-GMM is used as classifier, with a feature vector of 16 MFCCs.

The Discrete-Valued SNR Estimator is utilized to fuse the decision of the six TI engines using Multi-Engine Selection and Multi-Engine Weighted Fusion using the estimated SNR value. Compared to the classification accuracy using clean engine, the

Multi-Engine Weighted Fusion method has higher classification performance than Multi-Engine Selection method (Figure 7.8.1). This higher classification accuracy have values of 5.4%, 5.3% and 4.7%, at $RT60 = 0.50$, 1.15 and 1.80 sec, respectively over an SNR range of clean to 5 dB. For $SNR < 5$ dB both methods exhibit similar classification accuracy. Compared to the classification accuracy using clean engine, the improvements in accuracy using Multi-Engine Weighted Fusion were 26.5%, 21.7% and 18.1 % at $RT60 = 0.50$, 1.15 and 1.80 sec, respectively.

Figure 7.8.1 Talker identification accuracy using Multi-Engine Weighted Fusion and Multi-Engine Selection methods at different reverberant environments vs. SNR using white noise

At certain SNR value, as the reverberation level is increased the performance of the Multi-Engine Weighted Fusion is decreased. Meanwhile in reverberant environment the performance of the Multi-Engine Weighted Fusion still has higher performance than the engine trained in clean environment (at $RT60 = 0.50$ sec and $SNR = 20$ dB the performance of Multi-Engine Weighted Fusion = 75.0% and clean engine = 20.0%).

For low noise; $SNR > 30$ dB, the multi-engine systems do not improve the classification accuracy, in fact it does worse, except in the non-reverberant environment. At $SNR = 35$ dB and $RT60 = 1.8$ sec the Multi-Engine Selection, the Multi-Engine Weighted Fusion, and the clean engine methods have accuracies of 48%, 59% and 63%, respectively. These results mean that the clean engine has more information about the surrounding environment than the other methods. At $SNR < 30$ dB the Multi-Engine Selection and the Multi-Engine Weighted Fusion favored the TI engines which have SNR near the surrounding environments. This TI engines have higher information about the surrounding environment than the clean TI engine which increase the robustness of the ASC system.

The Multi-Engine Weighted Fusion is a more generalized solution than the Multi-Engine Selection and clean engine for ASC systems in hybrid reverberant and noisy environments. Multi-Engine Weighted Fusion method enhanced the TI performance even if the surrounding environment is not one of the training environments.

7.9 Discussion

RSS and Discrete-Valued noise estimator are introduced to calculate the environment variables, RT60 and SNR respectively. Using the estimated environment variable, a Multi-Engine ASC system for TI is introduced that has high performance in variable environments.

Two methods were introduced to perform Multi-Engine ASC; Multi-Engine Selection method and Multi-Engine Weighted Fusion method. Multi-Engine Weighted Fusion method enhanced the TI performance even if the surrounding environment is not one of the training environments. This enhancement is achieved because the Multi-Engine Weighted Fusion method does not just select one TI engine associated with a certain environment; rather, it weights and combines all output decisions of the ASC engines using the environment information estimator.

Compared to the baseline method, the introduced ASC methods achieved robust performance with high classification accuracy in reverberant, noisy, real, and hybrid reverberant and noisy environments.

Chapter 8

Application of ASC Systems in Variable Environments

In the previous two chapters, we introduced novel adaptive ASC features and classifiers methods which have a robust and high accuracy performance in variable environments. These methods select or combine the weighted decision of each feature and classifier using the environment information to enhance the overall classification accuracy and robustness.

In this chapter, we introduce combined ASC systems in variable environments. A novel security application using this combined ASC is also introduced. This instrument estimates the environment variable to adapt the speech/non-speech classifier to the surrounding environment. If the detected signal is speech, a talker identification (TI) system using the Multi-Engine methods will identify the talker; otherwise, an audio classification system is employed to determine the audio class of the detected signal (e.g. widows breaking, and wind sounds).

8.1 Security System Overview

This system integrates an environment information estimator (in this system it is Discrete-Valued SNR estimator) with a Multi-Engine TI system, Non-Speech audio classifier, Speech/non-speech classifier and Microphone array. A block diagram of the introduced instrument is shown in Figure 8.1.1. The microphone array captures the audio signal then sends the digitized audio signal to the localization block to determine the location of the sound source in addition to steering the microphone array to eliminate undesired acoustic noise [90].

Figure 8.1.1 Introduced security system

An APR speech/non-speech classifier is used as a front-end to the TI system and the audio classification system. If the detected signal is non-speech the audio classification system will determine the audio class of the detected signal (e.g. footsteps, windows breaking, and door opening sounds); otherwise, the speech signal will be sent to both the Discrete-Valued SNR Estimator and the Multi-Engine TI system, consisting of E TI engines to identify the talker. Each TI engine is trained in an environment with a specific noise condition. Using the SNR value from the Discrete-Valued SNR estimator, the system fuses the outputs of the E TI engines. The resultant data (i.e. location, talker identity, and audio class) are sent to the data processing system for application specific tasks.

The introduced system can be utilized in hands-free audio conferencing systems to perform voice-based security authentication for the conference users [91]. Another application of the introduced system is in modern video conferencing systems that use a microphone array and a steerable camera [90]. The introduced instrument can determine the identity and location of the talker of interest then steer the camera toward that talker; this system would also avoid steering the camera inappropriately to noise sources (e.g. fan noise). The system can serve a dual function, as a security surveillance system, when the conferencing function is not in session. Also, the introduced instrument can be utilized in smart homes and health monitoring for older adults that will support independent living and allow them to continue to age in their own homes [92]. Smart homes can benefit from the introduced system by monitoring their occupants through analysis of the audio signals present in the home and recognition of certain sound

sources. For instance, respiration, falls to the ground, and water leak sounds could be detected. The introduced system could perform further analyses and make decisions according to the captured audio class, such as extracting the respiration rate, calling emergency help, or alerting the occupant that a plumber may be needed. The analysis of the audio data preserves the privacy of the users better than video monitoring. Previously mentioned applications require high performance to have successful functionality. In addition, it needs to be robust even in variable conditions, such as environments with different levels of noise. Poor system robustness would limit utility of these applications.

8.2 Overall Performance of the Combined ASC System

The speech/non-speech using the APR algorithm is trained using half of the 1356 speech/non-speech database audio segments, using white noise, and tested with the other half over an SNR range of clean to -10 dB environments, using white and pink noise. The average performance using the pink and white noise of APR algorithm using the Discrete-Valued SNR Estimator is shown in Table 8.2.1. The average classification performance of the pink noise database is similar to the white noise database. Also, the classification performance of KING database is similar to the speech audio database. Due to the deterioration of the pitch period, APR has a poor speech classification performance of 32.5% and 28.2% for White noise and Pink noise at low SNR = -10 dB, respectively. APR maintains an overall accuracy above 93.0% for SNR above 0 dB.

Table 8.2.1 Speech/non-speech classification results (%)

Audio class \ SNR (dB)		-10	0	10	20	30	Clean
White Noise	Nonspeech	84.0	95.2	98.1	98.3	98.6	98.6
	Speech	32.0	93.2	96.8	97.3	97.3	97.3
	KING	33.0	93.7	98.0	98.6	98.6	98.6
	Average	58.2	94.3	97.7	98.1	98.3	98.3
Pink Noise	Nonspeech	87.0	94.0	97.5	97.8	98.4	98.6
	Speech	27.1	90.0	96.0	95.9	95.9	96.7
	KING	29.2	89.1	97.5	98.2	97.4	97.5
	Average	57.5	91.7	97.1	97.8	97.9	97.9
Speech/non-speech Average		57.8	93.0	97.4	98	98.1	98.1

Figure 8.2.1 Overall accuracy of the introduced system

The overall adaptive system accuracy (with SNR estimator and APR) is compared with the overall fixed instrument accuracy (clean TI engine and PR threshold = 0.32) (Figure 8.2.1). The main contributor of the accuracy enhancement of the introduced instrument at SNR above 10 dB is the Multi-Engine TI system (from Table 8.2.1 the performance of the APR is nearly perfect for SNR above 10 dB). The performance curve

of the Multi-Engine Selection, shown in Figure 8.2.1, has a large dip at $\text{SNR} = 5$ dB. For SNR below 10 dB the performance curve could be smoothed by decreasing the SNR spacing between TI engines.

The classification accuracy performance using a perfect SNR estimator with the APR and Multi-Engine Selection is shown in Figure 8.2.1. Using this SNR value, the best-performing TI engine and the APR threshold corresponding to the highest MCC, are selected. Compared to APR with Multi-Engine Weighted Fusion using the Discrete-Valued SNR Estimator the improvement in accuracy is only 1.6% over SNR range clean to -10 dB. This result justifies the utility of the rough estimation of the environment variable value.

8.3 Discussion

This chapter introduced combined ASC instrument that can detect location and identity of talker in noisy environment. To achieve this instrument adaptive talker identification system and Discrete-Valued SNR Estimator are utilized. SNR estimator is also utilized to adapt the speech/non-speech classifier. The introduced instrument is reliable and robust even in variable noise environment.

The utility of the rough environment information estimator has been justified by comparing the performance of the introduced system performance using a perfect environment estimator and the introduced environment estimator. This comparison shows that a slightly performance improvement has been achieved by using the perfect environment estimator.

Chapter 9

Conclusion and Future Work

9.1 Conclusion

The performance of most audio processing systems is degraded severely when the training and testing audio are captured in different acoustic environments. More specifically, mismatches between the training and testing environments result in decreased accuracy; these mismatches can arise due to a variable noise or reverberant environments.

A novel framework to solve the accuracy performance degradation of ASC in variable environments was presented. This framework involves a system that adapts to its environment. Several methods were introduced within the framework, and each method was introduced and discussed. The abstract implementation of these methods was also presented in this thesis.

New adaptive ASC features (APR and VLER), which have a resilient accuracy performance in variable and harsh environments (low SNR and high reverberation) were introduced. APR achieved classification accuracy of 92.0% in noisy environments over an SNR range of clean environment to -10 dB. Compared to conventional features the

improvement in accuracy was as high as 34.8%. Also, achieved overall classification accuracy of 95.2% in reverberant environments over an RT60 range of clean environment to 1.80 sec. Compared to conventional features the improvement in accuracy was as high as 13.5%. VLER achieved overall classification accuracy of 85.0% in noisy environments over an SNR range of clean environment to -20 dB. Compared to conventional methods the improvement in accuracy was as high as 31.0%.

A new generalized adaptation method for the ASC features was also developed. Also Multi-Feature Selection Multi-Feature Combination methods were introduced. These methods select or combine the weighted decision of each feature using the environment information to enhance the overall classification accuracy and robustness. Multi-Feature Selection achieved overall classification accuracy of 87.0% in noisy environments over an SNR range of clean environment to -20 dB. Compared to conventional features the improvement in accuracy was as high as 33.0%. Multi-Feature Combination achieved overall classification accuracy of 88.0% in noisy environments over an SNR range of clean environment to -20 dB. Compared to conventional methods the improvement in accuracy was as high as 34.0%.

Methods to estimate environment variables; Discrete-valued SNR estimator and RSS, were introduced. We have shown that the Discrete-valued SNR estimator and RSS achieved classification accuracy of 98.4% and 87.5%, respectively. Using the estimated environment variable, two methods have been developed to perform Multi-Engine ASC, which were evaluated in variable reverberant and noisy environments.

The first method is the Multi-Engine Selection method, which selects the classifier with the reverberation training condition that best matches the surrounding environment achieved classification accuracy of 86.0% in noisy environments over an SNR range of clean environment to 0 dB. Compared to conventional methods the improvement in accuracy was as high as 43.0%. Also, we have shown that it has overall classification accuracy 95.4% in reverberant environments over an RT60 range of clean environment to 1.8 sec. Compared to conventional method the improvement in accuracy was as high as 9.0%. The second method is the Multi-Engine Weighted Fusion method, which weights the output decisions of the ASC classifiers, using the environment estimator output, and combines these weighted decisions to perform the ASC. Multi-Engine Weighted Fusion achieved overall classification accuracy of 91.5% in noisy environments over an SNR range of 0 to 40 dB. Compared to conventional methods the improvement in accuracy was as high as 48.5%. Also, we have shown that it has overall classification accuracy of 96.5% in reverberant environments over an RT60 range of clean environment to 1.80 sec. Compared to conventional methods the improvement in accuracy was as high as 10.0%.

A complete ASC system that is capable of adapting in a variable environment was presented. A novel security application using this combined ASC was also discussed

9.2 Future Work

1. Introduce ASC using Multi-Engine Weighted Fusion and Multi-Engine Selection trained in hybrid reverberant and noisy environment. In these Multi-Engine Methods the engines should be trained in hybrid

environments. These methods will better suit the real environment; in real environment there are reverberation and noise in the same time. Using these hybrid engines will increase the performance of the ASC.

2. Introduce Hybrid environment variables estimator. This estimator should estimate the values of the reverberation and the noise in the hybrid environment. Using these values the Multi-Engine Selection and Multi-Engine Weighted Fusion can be conducted. This estimator will provide information about the surrounding hybrid environment. Using this information the ASC will fuse the output decision of the multi-engine.
3. Deployment of my introduced research in new areas such as biomedical applications. An example of a biomedical application area that can benefit from my introduced research is hearing aid devices. Hearing aids could utilize the introduced research to customize their parameters to achieve an optimal hearing performance in variable environments. Modern hearing aids allow the user to set their internal equalizer or beam former variables to predefined values by selecting the surrounding environment (e.g. small room, theater, or street noise). However, the major users of hearing aids are older adults, for whom the task of continually updating the environment selection may be difficult or tedious. The deployment of the introduced research in hearing aids can automate the selection of the predefined settings according to the surrounding environment.

4. Deployment of the research in smart homes and health monitoring for older adults. This deployment will allow them to continue to age in their own homes, keeping as much independence as possible. Smart homes can benefit from the introduced research by monitoring their occupants through analysis of the audio signals present in the home and recognition of certain sound sources. For instance, respiration, falls to the ground, and water leak sounds could be detected. The introduced system could make further analysis and make decisions according to the captured audio class, such as extracting the respiration rate, calling emergency help, or alerting the occupant that a plumber may be needed. The analysis of the audio data preserves the privacy of the users better than video monitoring.

Appendix A

Microphone Array

A.1 Linear Delay-and-Sum Beamformer

A simple audio beamforming algorithm can be implemented by adding the captured audio signals at each microphone of the microphone array. This addition should have a delay at each microphone to match the propagation delay of the desired wavefront (direction). The output of the beamformer is given as:

$$A_{out}(t) = \sum_{n=0}^{N-1} I_n(t - \tau_n) \quad (A.1)$$

where $I_n(t)$ and τ_n are the received signal and the delay at the n^{th} microphone respectively, and N is the total number of the microphones.

Linear delay-and-sum beamformer is an example of simple structure beamformers as shown in Figure A.1.1 [93]. Assuming an acoustic sinusoidal plane wave of frequency f is the source signal. Then the received signal at microphone n is:

$$I_n(t) = e^{j2\pi f(t - \tau_n)} \quad (A.2)$$

where τ_n' is the propagation delay of the incident wave detected at microphone n from the reference microphone. As shown in Figure A.1.2. τ_n' can be calculated as follows:

$$\tau_n' = h_n \cos(\theta) / (\text{propagation speed} = 340 \text{ m/s}) \quad (\text{A.3})$$

Figure A.1.1 Delay-and-sum beamformer

From (2.4.1) and (2.4.2) the output of the beamformer is calculated as follows:

$$A_{out}(t) = e^{jt2\pi f} \sum_{n=0}^{N-1} e^{-j2\pi f(\tau_n - \tau_n')} \quad (\text{A.4})$$

From the previous equation to maximize the output of the beamformer τ_n' has to be equal to τ_n . So to steer the microphone array to a certain direction, its delays should be set to be equal to the propagation delays of the incident wave from the same angle. As shown in Figure A.1.3 the beampattern for linear array, assuming single sinusoidal source, has sinc shape [93].

Figure A.1.2 Incident of plan wave on a linear microphone array

Different geometrical shape arrays generate different beampatterns such as a circular microphone array which have a symmetrical beampattern at 0° to 360° [90].

Figure A.1.3 Beampattern of six element linear array steered to 90° from the array axis. Radial scale indicates the normalized directive gain

Appendix B

Reverberant Environments Setup

The specifications of the reverberant environments are shown in (Table B.1.1- Table B.1.16). Samples of the impulse response and energy decay curves are illustrated in Figure B.1.4 - Figure B.1.6; simulation is conducted at sampling frequency = 8 Hz.

Table B.1.1: Reverberation specifications for $d = 0.75$ m

Specification	RT60 = 0.25 sec	RT60 = 0.5 sec	RT60 = 0.68 sec
Room Dimensions	3.00 x 3.00 x 2.80	3.80 x 3.90 x 2.80	4.80 x 4.70 x 2.80
Room volume	25.2	41.5	63.2
Source position	1.00 x 2.15 x 1.30	1.10 x 2.15 x 1.40	1.20 x 2.15 x 1.40
Receiver position	1.30 x 2.40 x 1.03	1.75 x 2.40 x 1.13	1.85 x 2.40 x 1.13
Wall reflection coeffs.	0.85, 0.85, 0.85, 0.85	0.90, 0.90, 0.90, 0.90	0.91, 0.91, 0.91, 0.91
Ceiling & floor reflection coeffs.	0.70, 0.70	0.71, 0.71	0.73, 0.73
RT60	0.25	0.50	0.68

Table B.1.2: Reverberation specifications for $d = 0.75$ m continued (1)

Specification	RT60 = 0.85 sec	RT60 = 1.00 sec	RT60 = 1.15 sec
Room Dimensions	5.40 x 5.20 x 2.80	5.50 x 5.30 x 3.00	6.50 x 5.4 x 3.00
Room volume	78.6	87.5	105.3
Source position	1.40 x 2.15 x 1.40	1.40 x 2.15 x 1.50	1.40 x 2.35 x 2.50
Receiver position	2.05 x 2.40 x 1.13	2.05 x 2.40 x 1.23	2.05 x 2.60 x 2.23
Wall reflection coeffs.	0.92, 0.92, 0.92, 0.92	0.93, 0.93, 0.93, 0.93	0.93, 0.93, 0.93, 0.93
Ceiling & floor reflection coeffs.	0.75, 0.75	0.8, 0.8	0.85, 0.85
RT60	0.85	1.00	1.15

Table B.1.3: Reverberation specifications for $d = 0.75$ m continued (2)

Specification	RT60 = 1.25 sec	RT60 = 1.35 sec	RT60 = 1.50 sec
Room Dimensions	6.50 x 6.80 x 3.00	7.40 x 6.8 x 3.00	7.00 x 7.40 x 3.00
Room volume	132.6	151.0	155.04
Source position	3.40 x 3.35 x 2.50	3.40 x 3.35 x 1.70	3.40 x 3.35 x 1.70
Receiver position	4.05 x 3.60 x 2.23	4.05 x 3.60 x 1.43	4.05 x 3.60 x 1.43
Wall reflection coeffs.	0.93, 0.93, 0.93, 0.93	0.93, 0.93, 0.93, 0.93	0.937, 0.937, 0.937, 0.937
Ceiling & floor reflection coeffs.	0.85, 0.85	0.85, 0.85	0.85, 0.85
RT60	1.25	1.35	1.50

Table B.1.4: Reverberation specifications for $d = 0.75$ m continued (3)

Specification	RT60 = 1.65 sec	RT60 = 1.80 sec
Room Dimensions	7.00 x 8.00 x 3.00	7.10 x 8.10 x 3.00
Room volume	168.0	172.5
Source position	3.40 x 4.35 x 1.7	3.40 x 4.35 x 1.70
Receiver position	4.05 x 4.05 x 4.60	4.05 x 4.60 x 1.43
Wall reflection coeffs.	0.94, 0.94, 0.94, 0.94	0.945, 0.945, 0.945, 0.945
Ceiling & floor reflection coeffs.	0.85, 0.85	0.85, 0.85
RT60	1.65	1.80

Table B.1.5: Reverberation specifications for $d = m 0.87$

Specification	RT60 = 0.25 sec	RT60 = 0.5 sec	RT60 = 0.68 sec
Room Dimensions	3.00 x 3.00 x 2.80	3.80 x 3.90 x 2.80	4.80 x 4.70 x 2.80
Room volume	25.2	41.5	63.2
Source position	2.20 x 1.50 x 1.70	3.20 x 2.50 x 1.70	3.20 x 3.50 x 1.70
Receiver position	2.60 x 2.00 x 1.10	3.60 x 3.00 x 1.10	3.60 x 4.00 x 1.10
Wall reflection coeffs.	0.85, 0.85, 0.85, 0.85	0.90, 0.90, 0.90, 0.90	0.915, 0.915, 0.915, 0.91
Ceiling & floor reflection coeffs.	0.70, 0.70	0.71, 0.71	0.73, 0.73
RT60	0.25	0.50	0.68

Table B.1.6: Reverberation specifications for $d = 0.87$ m continued (1)

Specification	RT60 = 0.85 sec	RT60 = 1.00 sec	RT60 = 1.15 sec
Room Dimensions	5.40 x 5.20 x 2.80	5.50 x 5.30 x 3.00	6.50 x 5.4 x 3.00
Room volume	78.6	87.5	105.3
Source position	3.20 x 3.50 x 1.70	3.20 x 3.50 x 1.70	3.20 x 3.50 x 1.70
Receiver position	3.60 x 4.00 x 1.10	3.60 x 4.00 x 1.10	3.60 x 4.00 x 1.10
Wall reflection coeffs.	0.92, 0.92, 0.92, 0.92	0.93, 0.93, 0.93, 0.93	0.93, 0.93, 0.93, 0.93
Ceiling & floor reflection coeffs.	0.75, 0.75	0.8, 0.8	0.85, 0.85
RT60	0.85	1.00	1.15

Table B.1.7: Reverberation specifications for $d = 0.87$ m continued (2)

Specification	RT60 = 1.25 sec	RT60 = 1.35 sec	RT60 = 1.50 sec
Room Dimensions	6.50 x 6.80 x 3.00	7.40 x 6.8 x 3.00	7.00 x 7.40 x 3.00
Room volume	132.6	151.0	155.04
Source position	3.20 x 3.50 x 1.70	3.20 x 3.50 x 1.70	3.20 x 3.50 x 1.70
Receiver position	3.60 x 4.00 x 1.10	3.60 x 4.00 x 1.10	3.60 x 4.00 x 1.10
Wall reflection coeffs.	0.93, 0.93, 0.93, 0.93	0.931, 0.931, 0.931, 0.931	0.936, 0.936, 0.936, 0.936
Ceiling & floor reflection coeffs.	0.85, 0.85	0.85, 0.85	0.85, 0.85
RT60	1.25	1.35	1.50

Table B.1.8: Reverberation specifications for $d = 0.87$ m continued (3)

Specification	RT60 = 1.65 sec	RT60 = 1.80 sec
Room Dimensions	7.00 x 8.00 x 3.00	6.50 x 7.70 x 3.40
Room volume	168.0	170.20
Source position	3.20 x 3.50 x 1.70	3.20 x 3.50 x 1.70
Receiver position	3.60 x 4.00 x 1.10	3.60 x 4.00 x 1.10
Wall reflection coeffs.	0.94, 0.94, 0.94, 0.94	0.94, 0.94, 0.94, 0.94
Ceiling & floor reflection coeffs.	0.85, 0.85	0.85, 0.85
RT60	1.65	1.80

Table B.1.9: Reverberation specifications for $d = 1.00$ m

Specification	RT60 = 0.25 sec	RT60 = 0.5 sec	RT60 = 0.68 sec
Room Dimensions	2.90 x 2.60 x 2.50	4.00 x 3.50 x 2.70	4.70 x 4.50 x 2.80
Room volume	18.9	37.8	59.2
Source position	1.20 x 2.10 x 1.75	1.20 x 2.10 x 1.75	1.20 x 2.10 x 1.75
Receiver position	1.70 x 2.40 x 0.94	1.70 x 2.40 x 0.94	1.70 x 2.40 x 0.94
Wall reflection coeffs.	0.85, 0.85, 0.85, 0.85	0.90, 0.90, 0.90, 0.90	0.90, 0.90, 0.90, 0.90
Ceiling & floor reflection coeffs.	0.70, 0.70	0.70, 0.70	0.70, 0.70
RT60	0.25	0.50	0.68

Table B.1.10: Reverberation specifications for $d = 1.00$ m continued (1)

Specification	RT60 = 0.85 sec	RT60 = 1.00 sec	RT60 = 1.15 sec
Room Dimensions	5.20 x 5.10 x 2.80	5.50 x 5.20 x 3.00	6.50 x 4.90 x 3.00
Room volume	74.3	85.8	93.6
Source position	1.20 x 2.10 x 1.75	1.20 x 2.10 x 1.75	1.20 x 2.10 x 1.75
Receiver position	1.70 x 2.40 x 0.94	1.70 x 2.40 x 0.94	1.70 x 2.40 x 0.94
Wall reflection coeffs.	0.92, 0.92, 0.92, 0.92	0.93, 0.93, 0.93, 0.93	0.93, 0.93, 0.93, 0.93
Ceiling & floor reflection coeffs.	0.75, 0.75	0.80, 0.80	0.85, 0.85
RT60	0.85	1.00	1.15

Table B.1.11: Reverberation specifications for $d = 1.00$ m continued (2)

Specification	RT60 = 1.25 sec	RT60 = 1.35 sec	RT60 = 1.50 sec
Room Dimensions	6.30 x 6.80 x 3.00	6.40 x 6.80 x 3.00	6.80 x 6.70 x 3.00
Room volume	128.5	130.56	136.7
Source position	1.20 x 2.10 x 1.75	1.20 x 2.10 x 1.75	1.20 x 2.10 x 1.75
Receiver position	1.70 x 2.40 x 0.94	1.70 x 2.40 x 0.94	1.70 x 2.40 x 0.94
Wall reflection coeffs.	0.93, 0.93, 0.93, 0.93	0.935, 0.935, 0.935, 0.935	0.94, 0.94, 0.94, 0.94
Ceiling & floor reflection coeffs.	0.85, 0.85	0.85, 0.85	0.85, 0.85
RT60	1.25	1.35	1.50

Table B.1.12: Reverberation specifications for $d = 1.00$ m continued (3)

Specification	RT60 = 1.65 sec	RT60 = 1.80 sec
Room Dimensions	6.80 x 6.80 x 3.00	6.80 x 6.80 x 3.00
Room volume	138.7	138.7
Source position	1.20 x 2.10 x 1.75	1.20 x 2.10 x 1.75
Receiver position	1.70 x 2.40 x 0.94	1.70 x 2.40 x 0.94
Wall reflection coeffs.	0.945, 0.945, 0.945, 0.945	0.95, 0.95, 0.95, 0.95
Ceiling & floor reflection coeffs.	0.85, 0.85	0.85, 0.85
RT60	1.65	1.80

Table B.1.13: Reverberation specifications for $d = 1.25$ m

Specification	RT60 = 0.25 sec	RT60 = 0.5 sec	RT60 = 0.68 sec
Room Dimensions	3.00 x 3.10 x 2.50	4.50 x 3.50 x 2.70	5.50 x 3.70 x 2.70
Room volume	23.2	42.5	55.0
Source position	1.00 x 1.50 x 1.50	1.00 x 1.50 x 1.50	3.00 x 1.50 x 1.50
Receiver position	1.55 x 2.50 x 1.00	1.55 x 2.50 x 1.00	3.55 x 2.50 x 1.00
Wall reflection coeffs.	0.84, 0.84, 0.84, 0.84	0.89, 0.89, 0.89, 0.89	0.90, 0.90, 0.90, 0.90
Ceiling & floor reflection coeffs.	0.70, 0.70	0.72, 0.72	0.74, 0.74
RT60	0.25	0.50	0.68

Table B.1.14: Reverberation specifications for $d = 1.25$ m continued (1)

Specification	RT60 = 0.85 sec	RT60 = 1.00 sec	RT60 = 1.15 sec
Room Dimensions	5.80 x 4.20 x 2.70	6.00 x 4.50 x 2.80	6.30 x 4.80 x 3.00
Room volume	65.8	75.6	90.7
Source position	3.00 x 2.50 x 1.50	3.00 x 2.50 x 1.50	3.00 x 2.00 x 1.70
Receiver position	3.55 x 3.50 x 1.00	3.55 x 3.50 x 1.00	3.55 x 3.00 x 1.20
Wall reflection coeffs.	0.915, 0.915, 0.915, 0.915	0.925, 0.925, 0.925, 0.925	0.93, 0.93, 0.93, 0.93
Ceiling & floor reflection coeffs.	0.76, 0.76	0.79, 0.79	0.85, 0.85
RT60	0.85	1.00	1.15

Table B.1.15: Reverberation specifications for $d = 1.25$ m continued (2)

Specification	RT60 = 1.25 sec	RT60 = 1.35 sec	RT60 = 1.50 sec
Room Dimensions	6.40 x 4.80 x 3.00	6.50 x 6.30 x 3.00	6.50 x 7.30 x 3.00
Room volume	92.2	122.9	142.4
Source position	3.00 x 1.50 x 1.70	3.00 x 1.50 x 1.70	3.00 x 2.50 x 1.70
Receiver position	3.55 x 2.50 x 1.20	3.55 x 2.50 x 1.20	3.55 x 3.50 x 1.20
Wall reflection coeffs.	0.935, 0.935, 0.935, 0.935	0.935, 0.935, 0.935, 0.935	0.936, 0.936, 0.936, 0.936
Ceiling & floor reflection coeffs.	0.85, 0.85	0.85, 0.85	0.85, 0.85
RT60	1.25	1.35	1.50

Table B.1.16: Reverberation specifications for $d = 1.25$ m continued (3)

Specification	RT60 = 1.65 sec	RT60 = 1.80 sec
Room Dimensions	6.50 x 7.60 x 3.00	6.50 x 7.70 x 3.00
Room volume	148.2	150.2
Source position	3.00 x 3.50 x 1.70	3.20 x 3.50 x 1.70
Receiver position	3.55 x 4.50 x 1.20	3.75 x 4.50 x 1.20
Wall reflection coeffs.	0.94, 0.94, 0.94, 0.94	0.945, 0.945, 0.945, 0.945
Ceiling & floor reflection coeffs.	0.85, 0.85	0.85, 0.85
RT60	1.65	1.80

Figure B.1.4 Impulse response and energy decay curve at $d=1.00$ m and RT60 = 0.25 sec

Figure B.1.5 Impulse response and energy decay curve at $d=1.00$ m and $RT60 = 1.00$ sec

Figure B.1.6 Impulse response and energy decay curve at $d=1.00$ m and $RT60 = 1.80$ sec

References

- [1] S. Ravindran and D.V.Anderson, "Audio classification and scene recognition and for hearing aids," in *Proc. IEEE Int. Symp. On Circuits and Systems*, vol. 2, 2005, pp. 860 – 863.
- [2] A.D.C. Chan, K.B. Englehart, B. Hudgins, D.F. Lovely, "Multi-expert automatic speech recognition using acoustic and myoelectric signals," *IEEE Trans. Biomedical Engineering*, vol. 53, no. 4, pp. 676-685, 2006.
- [3] A.R. Abu-El-Quran, R.A. Goubran, A.D.C. Chan, "Security-monitoring using microphone arrays and audio classification," *IEEE Trans. Instrumentation and Measurement*, vol. 55, no. 4, pp.1025-1032, 2006
- [4] J. Saunders, "Real-time discrimination of broadcast speech/music," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 1996, pp. 993-996.
- [5] E. Scheirer and M. Slaney, "Construction and evaluation of a robust multifeature speech/music discriminator," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 2, 1997, pp. 1331–1334.
- [6] S. Ravindran, and D.V. Anderson, "Audio classification and scene recognition and for hearing aids," in *proc. IEEE Int. Symp. Circuits and Systems*, vol. 2, 2005, pp. 860 – 863.

- [7] S. Srinivasan, K. Janse, "Spatial audio activity detection for hearing aids," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2008, pp. 4021-4024.
 - [8] A. R. Abu-El-Quran and R. A Goubran, "Adaptive pitch-based speech detection for hands-free applications," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 3, 2005, pp. 305-308.
 - [9] T. Zhong, and C.C. Jay Kuo, "Hierarchical classification of audio data for archiving and retrieving," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 6, 1999, pp. 3001-3004.
 - [10] T. Kawahara, Y. Nemoto and Y. Akita, "Automatic lecture transcription by exploiting presentation slide information for language model adaptation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2008, pp. 4929-4932.
 - [11] J. Gammal, *Speaker Recognition in Reverberant Environments*, Master Thesis, Carleton University, 2004.
 - [12] A. El-Solh, *Evaluation of speech enhancement techniques for speaker recognition in noisy environments*, Master Thesis, Carleton University, 2006.
 - [13] P. Castellano, S. Sridharan and D. Cole, "Speaker recognition in reverberant enclosures," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, pp. 117-120, 1996.
 - [14] K. Srinivasan, and A. Gersho, "Voice activity detection for cellular networks," in *Proc. IEEE Speech Coding for Telecom. Workshop*, 1993, pp. 85-86.
 - [15] D. Yu, L. Deng, J. Droppo, J. Wu, Y. Gong, A. Acero, "A minimum-mean-square-error noise reduction algorithm on Mel-frequency cepstra for robust
-

- speech recognition,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, NV, USA, 2008, pp. 4041-4044.
- [16] J.S. Gammal, and R.A. Goubran, “Combating reverberation in speaker verification,” in *proc. IEEE Instrumentation and Measurement Technology Conference.*, 2005, pp. 687- 690.
- [17] B. Clarkson, and A. Pentland, “Extracting context from environment audio,” in *Proc. IEEE Int. Sym. Wearable Computers*, 1998, pp. 154–155.
- [18] <http://www.bellsystemmemorial.com/belllabs.html>.
- [19] K. Kashino, G. Smith, and H. Murase, “Quick audio retrieval based on histogram feature sequences,” in *Journal of Acoust. Society of Japan (E)*, vol. 21, pp. 217–219, 2000.
- [20] J. Foote, “A similarity measure for automatic audio classification,” In *Proc. AAAI Sym. on Intelligent Integration and Use of Text, Image, Video*, 1997.
- [21] G. Lu, and T. Hankinson, “An investigation of automatic audio classification and segmentation,” in *Proc. IEEE Int. Conf. Signal Process.*, vol. 2, 2000, pp. 776–781.
- [22] A.J. Eronen, V.T. Peltonen, J.T.Tuomi, A.P. Klapuri, S. Fagerlund, T. Sorsa, G. Lorho, and J. Huopaniemi, “Audio-based context recognition,” *IEEE Trans. Speech, Lang., Process.*, vol. 14, pp. 321-329, 2006.
- [23] A. Rabaoui, Z.Lachiri, N. Ellouze, “Automatic environment noise recognition,” in *Proc. IEEE Int. Conf. on Industrial Tech.*, vol. 3, 2004, pp. 1670- 1675.
-

- [24] S. Essid, G. Richard, and B. David, "Instrument recognition in polyphonic music based on automatic taxonomies," *IEEE Trans. Speech, Lang., Process.*, vol. 14, pp. 68-80, 2006.
 - [25] M. Schroeder, *Fractals, chaos, power laws*, W. H. Freeman and Company, 1991.
 - [26] V. Mitra and C. Wang, "A neural network based audio content classification," in *Proc. IEEE TENCON*, 2007, pp. 1-6.
 - [27] A.R. Abu-El-Quran, *Feature extraction for audio classification*, Carleton University, Master Thesis, 2003.
 - [28] B.S. Atal, and L.R. Rabiner, "A pattern recognition approach to speech to voiced-unvoiced-silence classification with application to speech recognition," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 24, pp. 201-212, 1976.
 - [29] D. Huber, and R. Runstein, *Modern recording techniques*, Focal Press, 4th Ed., 1997.
 - [30] D. Johnson, and D. Dudgeon, *Array signal processing-concepts and techniques*, Prentice Hall, 1993
 - [31] T.F. Quatieri, *Discrete-time speech signal processing principles and practice*, Prentice Hall signal processing series, 2002.
 - [32] J. Proakis, *Digital Signal Processing*, McGraw Hill, 1996.
 - [33] Q. Zou, X. Zou, M. Zhang, and Z. Lin, "A robust speech detection algorithm in a microphone array teleconferencing system," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 5, 2001, pp. 3025-3028.
-

- [34] G.P. Mazarakis, and J.N. Avaritsiotis, "A prototype sensor node for footstep detection," in *Proc. IEEE European workshop, Wireless Sensor Networks*, 2005, pp. 415 – 418.
 - [35] C. Panagiotakis, G. Tziritas, "A speech/music discriminator based on RMS and zero-crossings," *IEEE Trans., Multimedia*, vol. 7, Issue: 1, pp. 155-166, 2005.
 - [36] R. Ahn, and W.H. Holmes, "Harmonic-plus-noise decomposition and its application in voiced/unvoiced classification," in *Proc. IEEE Conf. Computer and Telecommunication., Control and Power Engineering*, vol. 2 , 1997, pp. 587–590.
 - [37] R. Ahn, and W.H. Holmes, "Voiced/unvoiced/silence classification of speech using 2-stage neural networks with delayed decision input," in *Proc. IEEE Int. Sym. Signal Process., Applications*, vol. 1, 1996, pp. 389–390.
 - [38] E. Nemer, R. Goubran, and S. Mahmoud, "Robust voice activity detection using higher-order statistics in the LPC residual domain," *IEEE Trans. Speech and Audio Process.*, pp. 217–231, 2001.
 - [39] H. Kobatake, "Optimization of voiced/unvoiced decisions in nonstationary noise environments," *IEEE Trans., Acoust., Speech Signal Process.*, vol. 35, pp. 9–18, 1987.
 - [40] S.G. Tanyer, and H. Ozer, "Voice activity detection in nonstationary noise," *IEEE Trans. Speech and Audio Process.*, vol. 8, issue 4, pp. 478–482, 2000.
 - [41] S.G. Tanyer, and H. Ozer, "Voice activity detection in nonstationary Gaussian noise," in *Proc. IEEE Int. Conf. Signal Process.*, vol. 2, 1998, pp. 1620–1623.
-

- [42] R. Tucker, "Voice activity detection using a periodicity measure," in *Proc. IEEE Communications, Speech and Vision*, vol. 139, issue: 4, 1992, pp. 377–380.
 - [43] L.R. Rabiner, M.J. Cheng, A.E. Rosenberg, and C.A. McGonegal, "A comprehensive performance study of several pitch detection algorithms," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 24, pp. 399–417, 1976.
 - [44] M.M. Sondhi, "New methods of pitch extraction," *IEEE Trans. Audio Electroacoust.*, pp. 262–266, 1968.
 - [45] A.R. Abu-El-Quran, and R.A. Goubran, "Pitch-based feature extraction for audio classification," in *Proc. IEEE Haptic, Audio, Visual Environments and their Applications*, 2003, pp. 43–47.
 - [46] S. Davis, P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Trans. on Signal Process.*, vol. 28, Issue: 4, pp. 357–366, 1980.
 - [47] N. Tsourakis, *Speech Recognition*,
http://www.telecom.tuc.gr/~ntsourak/tutorial_sr.htm, 2002,
 - [48] M. Hunt, "Spectral signal processing for ASR," in *proc. IEEE Int. Workshop on Automatic Speech Recognition and Understanding*, 1999, pp. 17–25.
 - [49] G. Doddington, M.A. Prybocki, A.F. Martin, D.A. Reynolds, "The NIST speaker recognition evaluation-overview, method, systems, results, perspective," *Speech Communication*, vol. 31, no. 2–3, pp. 225–254, 2000.
-

- [50] J. Piquier, J.L. Rouas, and R. Andre-Obrecht, "A fusion study in speech/music classification," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Hong Kong, vol. 2, 2003, pp. 17-20.
 - [51] R. Andre-Obrecht, "A new statistical approach for automatic segmentation of continuous speech signals," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, pp. 29-40, 1988.
 - [52] Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*, 2nd Edition, Wiley, 2000.
 - [53] Jolliffe I.T., *Principal Component Analysis*, 2nd Edition, Springer, 2002.
 - [54] D. Y. Zhao, W. B. Kleijn, A. Ypma, and B. de Vries, "Online Noise Estimation Using Stochastic-Gain HMM for Speech Enhancement," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 16, no. 4, pp. 835-846, 2008
 - [55] D. Reynolds, R. Rose, "Robust text-independent speaker identification using Gaussian mixture speaker models," *IEEE Trans. on Speech and Audio Processing*, vol. 3, Issue: 1, pp. 72-83, 1995.
 - [56] Y. Linde, A. Buzo, R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. on Communications*, vol. 1, pp. 84-95, 1980.
 - [57] S. Furui, "Recent advances in speaker recognition," in *proc. IEEE Audio, Video-based Biometric Person Authentication*, 1997, pp. 237-252
 - [58] Bennani, Y. Gallinari, P. "On the use of TDNN-extracted features information in talker identification," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 1, 1991, pp. 385-388.
-

- [59] A. Bendiksen, and K. Steiglitz, "Neural networks for voiced/unvoiced speech classification," in *Proc. IEEE. Int. Conf. Acoust., Speech, Signal Process.*, vol.1, 1990, pp. 521–524.
 - [60] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, pp. 359–366, 1989.
 - [61] G. Lu, and T. Hankinson, "A technique towards automatic audio classification and retrieval," in *Proc. IEEE Int. Conf. Signal Process.*, vol. 2, Oct. 1998, pp. 1142–1145.
 - [62] R. Huang, and J.H.L. Hansen, "Advances in unsupervised audio classification and segmentation for the broadcast news and NGSW corpora," *IEEE Trans. Audio, Speech and Language Process.*, vol. 14, no. 3, pp. 907- 919, 2006
 - [63] G. Tzanetakis, and P. Cook, "Musical genre classification of audio signals," *IEEE Trans.s Audio, Speech and Language Process.*, vol. 10, no. 5, pp. 293-302, 2002
 - [64] D. Gerhard, "Audio Signal Classification: An Overview," in *proc. Canadian Artificial Intelligence*, 2000, pp. 4-6.
 - [65] C. Senac and E. Ambikairajah, "Audio indexing using feature warping and fusion techniques," in *proc. Int. Workshop on Multimedia Signal Process.*, 2004, pp. 359 – 362
 - [66] K. El-Maleh, M. Klein, G. Petrucci, and P. Kabal, "Speech/Music Discrimination For Multimedia Applications," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 6, 2000, pp. 2445–2448.
-

-
- [67] M.J. Carey, E.S. Parris, and H. Lloyd-Thomas, "A comparison of features for speech, music discrimination," in *proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 1, 1999, pp. 149–152.
- [68] D. Reynolds, "An overview of automatic speaker recognition technology", in *proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 4, 2002, pp. 13-17.
- [69] D. Reynolds, T. Quatieri, and R. Dunn, *speaker verification using adapted Gaussian mixture models*, in *Digital Signal Processing*, Academic Press, vol. 10, pp. 19-41, 2000
- [70] L.P. Cordella, P. Foggia, C. Sansone, M. Vento, "A real-time text-independent speaker identification system," in *Proc. IEEE Int. Conf. Image Analysis, Process.*, 2003, pp. 632-637.
- [71] E. Wold, T. Blum, D. Keislar, and J. Wheaton, "Content-based classification, search and retrieval of audio," *IEEE Multimedia Magazine*, vol. 3, pp. 27–36, 1996.
- [72] Muscle Fish LLC. <http://www.musclefish.com/>, 2001
- [73] J. B. Allen, D. A. Berkley, "Image method for efficiently simulating small-room acoustics", *Journal of the Acoustical Society of America*, vol. 65. No. 4. pp. 943-950, 1979.
- [74] International Telecommunications Union (ITU), *Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs*, Telecom Standardization Section, ITU-T, p.862, 2001.
-

- [75] National Institute of Standards and Technology (NIST), *Brief Description of the KING Speech Data Base*, NIST, 1992.
 - [76] H. Kuttruff, *Room Acoustics*, 4th edition, Elsevier, 2000.
 - [77] N. Mesgarani, M. Slaney, and S.A. Shamma, "Discrimination of speech from non-speech based on multiscale spectro-temporal Modulations," *IEEE Trans. Audio, Speech and Language Process.*, vol. 14, no. 3, pp. 920- 930, 2006.
 - [78] M. Wu and D. Wang, "A pitch-based method for the estimation of short reverberation time," *Acta Acustica United with Acustica*, vol. 92, pp. 337-9, 2006.
 - [79] Xi Shao, Changsheng Xu, M.S. Kankanhalli, "Unsupervised classification of music genre using hidden Markov model," in *Proc., Int. Conf. on Multimedia, Expo*, vol. 3, 2004, pp. 2023–2026.
 - [80] C. Senac and E. Ambikairajah, "Audio indexing using feature warping and fusion techniques," in *Proc., Int. Workshop on Multimedia Signal Process.*, 2004, pp. 359 - 362.
 - [81] Gonzalez Rodriquez, J. Ortega-Garcia, "Increasing robustness in GMM speaker recognition systems for noisy and reverberant speech with low complexity microphone arrays," in *Proc. IEEE Int. Conf. on Spoken Lang. Process.*, vol. 3, 1996, pp. 1333-1336.
 - [82] P.J. Barger and S. Sridharan, "On the performance and use of speaker recognition systems for surveillance," in *Proc. IEEE Video, Signal Based Surveillance*, 2006, pp. 109 – 114.
-

- [83] Q. Lin, E. Jan, J. Flanagan, "Microphone arrays and speaker identification", *IEEE Trans. on Speech and Audio Process.*, vol. 2, Issue: 4, pp. 622-629, 1994
 - [84] I. McCowan, J. Pelecanos, S. Sridharan, "Robust speaker recognition using microphone arrays," in *Proc. Of A speaker Odyssey*, 2001, pp. 101-106.
 - [85] E. Jan, J. Flanagan, "Microphone arrays for speech processing," in *proc. International Symposium Signals, Systems, and Electronics*, 1995, pp. 373-376.
 - [86] A.R. Abu-El-Quran, R.A. Goubran, and A.D.C. Chan, "Adaptive feature selection for speech/music classification," in *IEEE Workshop Multimedia Signal Process.*, 2006, pp. 121-216.
 - [87] Brunelli R, Falavigna D, "Person identification using multiple cues," *IEEE Trans. Pattern Analysis and Machine Intelligence*, pp. 955-966, 1995.
 - [88] R. Zilca, "Text-independent speaker verification using covariance modeling," *IEEE Signal Processing Letters*, vol. 8, No. 4, pp. 97-99, 2001
 - [89] F. Bimbot, I. Magrin-Chagnolleau, L. Mathan, "Second-order statistical measures for text-independent speaker identification", *Speech Communication*, vol. 17, no. 1-2, pp. 177-192, 1995.
 - [90] D. Lo, R.A. Goubran, R Dansereau, G. Thompson, and D. Schulz, "Robust Audio-video localization in video conferencing using reliability information," *IEEE Trans., Instrumentation and Measurement*, vol. 53, pp. 1132-1139, 2004.
 - [91] Y. Jeong, Y. Yun, B. Jung, K Kim, "Protection Profile for Security Enhancement of Embedded Operating System for Mobile Terminals," in *IEEE Proc. Advanced Communication Technology*, vol. 3, 2008, pp. 1908-1911.
-

-
- [92] M. Vainio, M. Valtonen, J. Vanhala, "Proactive Fuzzy Control and Adaptation Methods for Smart Homes," in *IEEE Proc. Intelligent Systems*, vol. 23, no. 2, 2008, pp. 42-49.
- [93] J. Kraus, *Electromagnetics*, McGraw Hill, 1992.
-