Incorporating Microphone Arrays into Automatic Speech Recognition

Filippos Ineglis

Thesis submitted in partial fulfillment of the requirements for the

Master of Science degree in Computer Science

University of Crete
School of Sciences and Engineering
Computer Science Department
Voutes Campus, Heraklion, GR-70013, Greece

Thesis Advisor: Assoc. Prof. *Athanasios*, *Mouchtaris*

Heraklion, December 2016

This work was partially supported by **Institute of Computer Science, Foundation of Research and Technology Hellas**

UNIVERSITY OF CRETE
COMPUTER SCIENCE DEPARTMENT

Incorporating Microphone Arrays into Automatic Speech Recognition

Thesis submitted by
Filippos Ineglis
in partial fulfillment of the requirements for the
Master of Science degree in Computer Science

THESIS APPROVAL

Author: ______________________________
Filippos Ineglis

Committee approvals: ______________________________
Athanasios Mouchtaris
Associate Professor, Thesis Supervisor

____
Panagiotis Tsakalides
Professor, Committee Member

____
Xenofontas Dimitropoulos
Assistant Professor, Committee Member

Departmental approval: ______________________________
Antonios Argyros
Professor, Director of Graduate Studies

Heraklion, December 2016

Abstract

Automatic Speech Recognition (ASR) was initially introduced in the 1950s. Since then, a lot of effort has been made to improve speech recognition in single channel recordings. In the last few years, many researchers have shown interest in the combination of speech recognition and multichannel recordings, as many every day devices incorporate multiple microphones. These microphones are usually placed in specific topologies allowing us to take advantage of the directivity of the input signal and achieve more robust speech enhancement. Some examples of devices and applications are mobile phones, tablets, home automation services such as Amazon Echo and Google Home, digital personal assistants like Google Now, Siri, Cortana etc.

In the course of this thesis, we aim to create a robust ASR system combined with a front-end to improve speech recognition in challenging environments such as reverberant rooms with or without background noise. The experiments we examined included scenarios with stationary and moving speakers as well as overlapping speakers. To approach this problem, we divided it into three phases. The first phase was the experimentation on the training data for the acoustic model. Three acoustic models were trained to define the best acoustic model, one with clean speech signals, one with processed speech signals and one with the combination of the previous two training sets. During the second phase, we tested several front-ends, i.e. array processing techniques, and evaluated them in the context of their speech recognition performance. Each array processing technique consists of two main modules, a beamformer and a postfilter. In addition to that, we proposed a new front-end framework based on the binary masks and a Wiener postfilter which achieved better recognition results. The recognition results showed that the combination of a Superdirective beamformer followed by a Wiener postfilter performs better on single speaker experiments while the same beamformer combined with Binary Masks performs better on overlapping speaker experiments. The last phase was to use the outcome of the first and the second phase in order to create a robust combination of a front-end and an acoustic model.

In order to evaluate the performance of each acoustic model and each front-end, we used a common speech recognition metric known as Word Error Rate (WER). The final proposed acoustic model combined with the proposed front-end led to a significant improvement in WER in all experiments, i.e. stationary speaker, moving speaker and overlapping speakers. The relative improvement in terms of WER of the processed speech signals over the unprocessed speech signals for the three experiments is 62.4% for stationary speaker, 57.9% for moving speaker and 49.6 % for overlapping speakers. In particular, the modification we proposed for the binary masks used in the front-end for the scenarios with overlapping speakers, that is a spectral floor and a more strict criterion on the application of the binary masks, led to a relative improvement of 9.9% in WER results.

Περίληψη

Η Αυτόματη Αναγνώριση Ομιλίας πρωτοεμφανίστηκε το 1950. Έκτοτε έχουν γίνει πολλές προσπάθειες για την βελτίωσή της σε μονοφωνικές ηχογραφήσεις. Τα τελευταία χρόνια, πολλοί ερευνητές έχουν δείξει ενδιαφέρον στην Αυτόματη Αναγνώριση Ομιλίας και σε πολυκάναλες ηχογραφήσεις, καθώς όλο και περισσότερες συσκευές της καθημερινότητάς μας ενσωματώνουν όλο και περισσότερα μικρόφωνα. Τα μικρόφωνα αυτά τοποθετούνται σε καθορισμένες διατάξεις δίνοντάς μας την δυνατότητα να εκμεταλλευτούμε την κατευθυντικότητα του σήματος εισόδου και να επιτύχουμε καλύτερη ενίσχυση σήματος. Μερικά παραδείγματα τέτοιων συσκευών και εφαρμογών αποτελούν τα κινητά τηλέφωνα, τα tablets, οι συσκευές οικιακού αυτοματισμού όπως Amazon Echo, Google Home, οι ψηφιακοί προσωπικοί βοηθοί όπως Siri, Google Now, Cortana κ.α.

Στα πλαίσια αυτής της εργασίας, στόχος μας είναι να δημιουργήσουμε ένα σύστημα Αυτόματης Αναγνώρισης Ομιλίας συνδυασμένο με ένα σύστημα ενίσχυσης σήματος (front-end) για να επιτύχουμε τα βέλτιστα αποτελέσματα αναγνώρισης ομιλίας σε μη ευνοϊκές συνθήκες, όπως δωμάτια με έντονη αντήχηση ή/και θόρυβο. Τα πειράματα που εκτελέσαμε περιλαμβάνουν σενάρια με στάσιμους ομιλητές, κινούμενους ομιλητές καθώς και επικαλυπτόμενους ομιλητές. Για την καλύτερη προσέγγιση του προβλήματος, χωρίσαμε τη διαδικασία σε τρεις φάσεις. Η πρώτη φάση ήταν ο πειραματισμός πάνω στα δεδομένα που χρησιμοποιήσαμε για την εκπαίδευση του ακουστικού μοντέλου. Τα ακουστικά μοντέλα που εκπαιδεύσαμε ήταν τρία. Το πρώτο ακουστικό μοντέλο εκπαιδεύτηκε με σήματα καθαρής ομιλίας, το δεύτερο με επεξεργασμένα σήματα ομιλίας και το τρίτο με τον συνδυασμό των δύο παραπάνω. Κατά τη δεύτερη φάση, δοκιμάσαμε ποικίλα συστήματα ενίσχυσης σήματος (front-ends), δηλαδή τεχνικές επεξεργασίας πολυκάναλων αρχείων φωνής και τα αξιολογήσαμε με βάση τα αποτελέσματα της αναγνώρισης. Καθεμιά από τις μεθόδους επεξεργασίας πολυκάναλου σήματος που χρησιμοποιήσαμε, απαρτίζεται από δύο κύρια στοιχεία, το διαμορφωτή λοβού και το πολυκάναλο φίλτρο. Επιπλέον, προτείναμε μια μέθοδο βασισμένη στις δυαδικές μάσκες και το φίλτρο Wiener η οποία μας οδήγησε σε καλύτερα αποτελέσματα αναγνώρισης. Τα αποτελέσματα της αναγνώρισης ομιλίας έδειξαν ότι ο συνδυασμός του υπερκατευθντικού διαμορφωτή λοβού με το πολυκάναλο φίλτρο Wiener αποδίδει καλύτερα στην περίπτωση ενός ομιλητή ενώ ο ίδιος διαμορφωτής λοβού συνδυασμένος με δυαδικές μάσκες αποδίδει καλύτερα σε επικαλυπτόμενους ομιλητές. Κατά την τελευταία φάση, δημιουργήσαμε ένα ακουστικό μοντέλο το οποίο εκπαιδεύτηκε με καθαρά και επεξεργασμένα σήματα φωνής χρησιμοποιώντας ως μέθοδο ενίσχυσης (front-end) τις τεχνικές που αναφέραμε παραπάνω ως βέλτιστες.

Για την αξιολόγηση της απόδοσης κάθε ακουστικού μοντέλου και κάθε μεθόδου ενίσχυσης του σήματος (front-end), χρησιμοποιήσαμε μια μετρική η οποία είναι ευρέως γνωστή σε πειράματα αναγνώρισης ομιλίας, το ποσοστό των λάθος αναγνωρισμένων λέξεων. Η προτεινόμενη μέθοδος οδήγησε

σε σημαντική βελτίωση των αποτελεσμάτων. Σημειώθηκε σχετική μείωση στο ποσοστό των λάθος αναγνωρισμένων λέξεων κατά 62,4 % για στάσιμο ομιλητή, 57,9 % για κινούμενο ομιλητή και 49,6 % για επικαλυπτόμενους ομιλητές σε σχέση με τα αποτελέσματα αναγνώρισης των μην επεξεργασμένων σημάτων. Συγκεκριμένα, οι τροποποιήσεις που εφαρμόσαμε στο σύστημα ενίσχυσης σήματος και στις δυαδικές μάσκες στην περίπτωση των επικαλυπτόμενων ομιλητών, δηλαδή η εισαγωγή ενός συχνωτικού κατωφλιού και ένα πιο αυστηρό κριτήριο για την εφαρμογή των μασκών αυτών, οδήγησε σε σχετική βελτίωση κατά 9.9% σε σχέση με τις προεπιλεγμένες παραμέτρους.

Acknowledgements

First of all I would like to thank my supervisor, Professor Athanasios Mouchtaris for giving me the opportunity to work with him, for his patience, his support and for giving the opportunity to discover the world of research.

I would also like to thank the members of my dissertation committee, Professors Panagiotis Tsakalides and Xenofontas Dimitropoulos for their helpful comments and questions.

I am really very grateful to Dr Nikolaos Stefanakis for the continuous encouragement, constructive meetings, ideas and inputs. I would also like to thank Elena Karamichali for her continuous support during my experiments and my thesis writing.

I would like to acknowledge the Institute of Computer Science (FORTH–ICS) for the financial support and the necessary equipment during this work.

I would like to thank the secretaries of the Department of Computer Science (University of Crete) and especially Gelly Kosma and Rena Kalaitzakh.

I should also thank all my colleagues at the Signal Processing Lab. The warmest thanks to Despoina, Tasos, Giannis and Stavros for the friendly atmosphere and the useful discussions and feedback. Thank you all SPLers.

A dedicated thank you, to my closest friends Anna, Sotiria, Giannis, Vassilis, Aggelos, Panos, Giorgos, Dimitris A. and Dimitris S. for their support and believe in me.

Last, but not least, I would like to thank my family for their encouragement all these years, for giving me the opportunities to pursue my life goals and help me to endure through the darkest of times.

Contents

List of Figures

List of Tables

Abbreviations and Acronyms

AM	Acoustic Model
ASR	Automatic Speech Recognition
BLSTM	Bidirectional Long Short-Term Memory
CICS	Circular Integrated Cross Spectrum
CSD	Cross Spectral Density
CTM	Close Talking Microphone
DAS	Delay and Sum
DF	Directivity Factor
DI	Directivity Index
DNN	Deep Neural Networks
DSR	Distant Speech Recognition
LM	Language Model
MC-WSJ-AV	Multichannel Wall Street Journal Audio-Video
MVDR	Minimum Variance Distortionless
NIST	National Institute of Standards and Technology
NN	Neural Networks
POI	Point of Interest
PSD	Power Spectral Density
RIR	Room Impulse Response
SIR	Signal to Interference Ratio
SDR	Signal to Distortion Ratio
SNR	Signal to Noise Ratio
STFT	Short Time Fourier Transform
ULA	Uniform Linear Array
UCA	Uniform Circular Array
WCR	Word Correct Rate
WER	Word Error Rate
WNG	White Noise Gain
WSJ	Wall Street Journal

Chapter 1

Introduction

In recent years, the technological growth has led to a significant decrease of the price of hardware parts. Thus, a lot of popular devices used in everyday life include more advanced hardware, such as more powerful processors, more memory and multiple microphones.

The additional computational power and memory resulted in the development and establishment of demanding applications that could not be used before. One of these applications is Automatic Speech Recognition. ASR was incorporated in a number of existing services like web browsing, text dictation, GPS navigation etc. Despite of the availability of multiple microphones, most ASR systems like Google Now, Siri, Cortana, Navigon etc, utilize only monophonic recordings until recently.

Only in the last two years, ASR was combined with the research results on multichannel signal processing of the last decade and this led to two successful products, Amazon Echo and Google Home. This kind of technology can be used in smart home applications like home control automation but also in assisting elderly or disable people with their everyday tasks or even for triggering alerts in emergencies.

1.1 Problem Statement

Nowadays, in many applications, the human-computer interaction has changed. The keyboard and the mouse has been replaced by voice commands for convenience. Users are capable to interact with devices, even if they are not close to a microphone, just with their voice and without any physical contact. However, speech recognition in such occasions is a difficult task, especially in real life environments with reverberation and background noise. This introduces an new area of research known as Distant Speech Recognition (DSR). To overcome these obstacles, multiple microphones have been utilized due to the fact that their geometry allows us to robustly enhance the recorded signal.

Speech recognition in real life environments faces multiple difficulties. The first and most common problem in speech recognition is reverberation. In room and office environments, the speech signal propagates in every direction reflecting on multiple surfaces before being captured by the microphone. This results to a distorted speech signal and an inaccurate recognition.

Another problem that affects speech recognition is the background noise. In indoor and outdoor environments, there are externals factors that produce undesired sounds such as PC fans, air conditions, TVs, hi-fi, cars, buses, weather conditions etc.In order to achieve the most accurate speech recognition we have to separate the useful signal from the background noise.

Another challenging situation in DSR is overlapping speakers. In several applications, we need to capture, separate and enhance the signal from multiple speakers in order to recognize speech from each one of them. To overcome all these difficulties, microphone arrays are introduced and array processing techniques are applied.

1.2 Motivation

Although the research community trying to deal with the above problem is very active, there is still room for improvement since we are far from the performance of the close-talking microphone. We propose a front-end mechanism which reduces the reverberation effect and the background noise level in the recorded speech signals achieving better speech recognition results compared to other state-of-the-art methods.

1.3 Related Work

Distant speech recognition (DSR) has been a field of research for a lot of research groups in the last decade. Most of the effort has been made on the back-end of the speech recognition process. Parada et al. [1] trained several acoustic models based on the reverberation level of the training data, and they selected the most appropriate one using the C_{50} estimation which is added in the feature vectors as an additional parameter. Matassoni et al. [2] trained five Deep Neural Networks (DNN) with different architectures and combined the multiple transcriptions using ROVER. Nevertheless, yielding improvement in the performance of the back-ends that now represent the state-of-art is a difficult task. Thus, a lot of front-end techniques have been proposed that significantly contribute in the improvement of ASR. Iain McCowan et al. [3] proposed a front-end using a superdirective beamformer and binary masks for speech enhancement for speech recognition. They used visual information to estimate the Direction Of Arrival (DOA) of the incoming signal. Delcroix et al. [4] used a more complex scheme to address this problem. They applied the Minimum Variance Distortionless (MVDR) beamformer followed by the Weighted Prediction Error

(WPE) algorithm to suppress reverberation and ambient noise. In [5], Weninger proposed a different DOA estimation method based on cross spectrum phase analysis which is then used by a Delay and Sum beamformer. Robust features are extracted from the enhanced signal using Deep Recurrent Neural Networks (DRNN). Heymann et al. [6] proposed applying the Generalized Eigenvalue beamformer in order to enhance the signal. The weights of the beamformer were computed using a Bidirectional Long Short-Term Memory (BLSTM) neural network.

1.4 Contribution of this work

In this thesis, we present an extensive study on array processing techniques to achieve robust speech recognition. We experiment with multiple front-ends such as combinations of beamformers and postfilters and evaluate their performance in terms of Word Error Rate (WER) which is the percentage of words that are recognized incorrectly relative to the total amount of words. In addition, we propose a modification for the binary masks when used as a postfilter following the superdirective beamformer which leads to a better WER performance. More specifically, the front-end consists of a tuned superdirective beamformer and the binary masks with a more strict criterion of 1.4 and 20% spectral floor which resulted in better speech recognition for the experiments with single stationary speaker, single moving speaker and multiple overlapping speakers. We, also, apply a DOA estimation algorithm, based only on audio information and not on visual information, to define the position of the speaker. Lastly, the proposed front-end is a lightweight and a near real-time (NRT) process.

1.5 Organization of the thesis

The organization of this thesis is as follows. In Chapter 2, we discuss the background theory on signal processing, microphone arrays and array processing. We also provide the basics on beamforming and postfiltering including the beamformers such as the Delay and Sum beamformer, the superdirective beamformer and the MVDR beamformer and postfilters like the binary masks and the Wiener postfilter. In addition, we present the basics components of an ASR system including the acoustic model, the language model and the decoding procedure. In Chapter 3, we present the clean, multi-condition and extended acoustic models we trained based on the guidelines of the REVERB Challenge. We also present our experimentation on multiple front-ends and derive our proposed front-end for stationary, moving and overlapping speakers scenarios. In Chapter 4, we include the Word Error Rate (WER) results from all the trained and adapted acoustic models including the WER results of our proposed acoustic model. Finally, Chapter 5 concludes the results of this thesis and proposes future work for ASR and array processing.

Chapter 2

Background

In this chapter we present the background theory on array signal processing and ASR. The array signal processing techniques we use include beamforming and post-filtering. Also, we discuss about the tools for Automatic Speech Recognition, i.e. acoustic model training, acoustic model adaptation and decoding.

2.1 Basics on Multichannel Signal Processing

A microphone array is a group of sensors (microphones) placed in different locations usually in predefined arrangements. Two of the most common array topologies are Uniform Linear Arrays (ULA) and Uniform Circular Arrays (UCA) (see Figure 2.1). Microphone arrays are usually deployed in situations where a signal of interest has to be captured in challenging situations. In the context of this thesis, a UCA is used to capture speech signals from single or multiple speakers in reverberant rooms for better speech recognition performance. In the next section we discuss about the basics of signal processing.

Let's assume a microphone array consisted of *M* microphones, a captured multi-channel input signal **x**(t), a signal of interest **s**(t) and a multi-channel noise signal **n**(t) in the time domain. To perform array processing we have to transfer the time-varying signals from time domain to frequency domain as $\mathbf{x}(\omega, t)$, $\mathbf{s}(\omega, t)$ and $\mathbf{n}(\omega, t)$ respectively. We do that by using Short-Time Fourier Transform (STFT). Frequency domain coefficients can be represented as shown below

$$\mathbf{x}(\omega, t) \triangleq \begin{bmatrix} x_1(\omega, t) & x_2(\omega, t) & \dots & x_M(\omega, t) \end{bmatrix}^T \tag{2.1}$$

$$\mathbf{n}(\omega, t) \triangleq \begin{bmatrix} n_1(\omega, t) & n_2(\omega, t) & \dots & n_M(\omega, t) \end{bmatrix}^T \tag{2.2}$$

Supposing that the diffuse noise is additive noise, the input signal $\mathbf{x}(\omega, t)$ can be written as

$$\mathbf{x}(\omega,t) = \begin{bmatrix} x_1(\omega,t) \\ x_2(\omega,t) \\ \vdots \\ x_M(\omega,t) \end{bmatrix} = \begin{bmatrix} a_1(\omega,t)s(\omega,t) & + & n_1(\omega,t) \\ a_2(\omega,t)s(\omega,t) & + & n_2(\omega,t) \\ & \vdots & \\ a_M(\omega,t)s(\omega,t) & + & n_M(\omega,t) \end{bmatrix} \tag{2.3}$$

Before presenting the array techniques we use, it is necessary to define several useful processing basic components such as *steering vector*, *power spectral density matrix* (PSD) and *coherence matrix*.

- With the far-field assumption where an audio source is far from the array relatively to the size of the array [7] (also known as Fraunhofer diffraction), a plane wave will reach each microphone in T_i seconds after emission. Then, the useful signal $s(\omega,t)$ can be extracted from each microphone as

$$s(\omega,t) \; = \; d_i(\omega,\theta_s)x_i(\omega,t) \; = \; e^{-j\omega T_i}x_i(\omega,t), \qquad n = 1,2,\dots M \tag{2.4}$$

  From Figure 2.1, it is obvious that T_i is different for each microphone and depends on the angle of the incoming wave, θ_s, and the array topology (i.e. linear, circular or elliptical). The vector $\mathbf{d}(\omega,\theta_s)$ containing $e^{j\omega T_i}$ coefficients is known as *steering vector.*

- Next is *power spectral density matrix* (PSD). Let assume a signal $\mathbf{x}(\omega,t)$ in frequency domain in the form of equation 2.1 then the PSD is given by [8]

$$\begin{aligned} \mathbf{\Phi_{xx}}(\omega,t) &\triangleq \mathbf{E}[\mathbf{x}(\omega,t)\mathbf{x}^H(\omega,t)] \\ &= \frac{1}{N}\sum_{t_k\,=\,t-\frac{N-1}{2}}^{t+\frac{N-1}{2}} \mathbf{x}(\omega,t_k)\mathbf{x}^H(\omega,t_k) \end{aligned} \tag{2.5}$$

  where $\mathbf{E}[\cdot]$ denotes the expectation and *N* the number of frames. In the course of this thesis and because of the small window size of STFT analysis, PSD matrix can not be accurately calculated by using only one frame. So, we calculate the product of $\mathbf{x}(\omega,t)$ with its hermitian as an average of *N* consecutive frames including the frame of interest [7, 9]. For odd number of frames we use $\frac{N-1}{2}$ previous frames and $\frac{N-1}{2}$ next frames in addition to the current frame while for even number of frames, we use $\frac{N}{2}$ previous and $\frac{N}{2}-1$ next frames. As a result, the equation is formed as shown in equation 2.5 (for simplicity we assume the number of frames to be even)

- Lastly, it is necessary to define the *coherence function* [10, 11] which will be used in the following beamforming techniques. This coherence matrix $\mathbf{\Gamma}$ acts as a metric to indicate how consistent is the

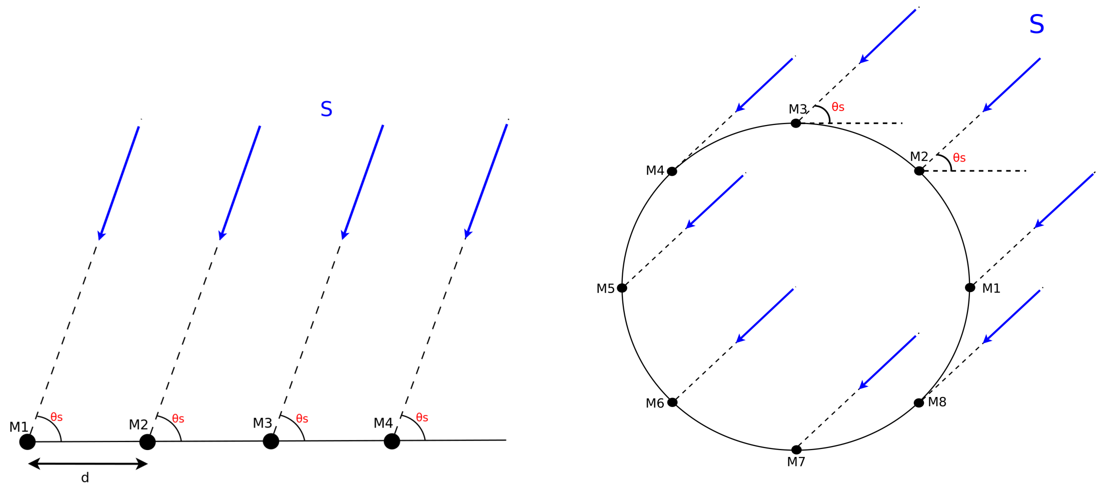


Figure 2.1: A Uniform Linear Array (ULA) with 4 microphones (on the left) and a Uniform Circular Array (UCA) with 8 microphones (on the right) in a far-field model with incoming sound from angle θ_s.

signal between every pair of microphones. In our case we use it to measure the coherence of the noise signal. More specifically, the position *(m,n)* of the matrix contains the coherence value between the noise signal acquired by the *m-th* microphone and the noise signal acquired by the *n-th* microphone. The magnitude of each array element lies within the range [0,1] with 1 indicating strong coherence between the two signals and 0 indicating no coherence. The diagonal elements $\Gamma_{x_k x_k}$ present strong coherence as the signal of each microphone is compared to itself and for these elements we have $\Gamma_{x_k x_k} = 1, \forall k \in [1, 2 \dots M]$. The coherence matrix $\mathbf{\Gamma}_{xx}$ is defined as [12, 13]

$$\mathbf{\Gamma}_{xx} = \begin{bmatrix} 1 & \Gamma_{x_1x_2} & \Gamma_{x_1x_3} & \dots & \Gamma_{x_1x_M} \\ \Gamma_{x_2x_1} & 1 & \Gamma_{x_2x_3} & \dots & \Gamma_{x_2x_M} \\ \Gamma_{x_3x_1} & \Gamma_{x_3x_2} & 1 & \dots & \Gamma_{x_3x_M} \\ \vdots & \vdots & & \ddots & \vdots \\ \Gamma_{x_Mx_1} & \Gamma_{x_Mx_2} & \Gamma_{x_Mx_3} & \dots & 1 \end{bmatrix} \tag{2.6}$$

where

$$\begin{aligned} \Gamma_{x_m x_n} &= \frac{\phi_{mn}}{\sqrt{\phi_{mm}\phi_{nn}}} \\ \phi_{mn} &= E[x_m(\omega,t)x_n^*(\omega,t)] \end{aligned} \tag{2.7}$$

ϕ_{mn} is the cross-spectral density between the signal of the *m-th* and *n-th* microphone and ϕ_{mm} the power spectral density of the signal from the *m-th* microphone. The equations in 2.7 can also be used, without the normalization factor in the denominator, to calculate each element of the PSD matrix.

In real life environments, such as small reverberant rooms and cars, the noise field can be modeled as either a spherical isotropic [14] or diffuse noise field [15]. The first noise field is created by the combination of a large number of noise sources spread in all directions producing noise signals from every angle with equal probability while in the second model the noise is white noise, uncorrelated across all sensors. In most speech enhancement applications a diffuse noise model is used [12]. In the course of this thesis we also assume a diffuse noise field model. Consequently, for the calculation of the coherence matrix we use the equation 2.8 which takes into consideration only the array physical dimensions (i.e. the distance betweeen every pair of microphones) instead of the general equations 2.6 and 2.7 [12, 16, 17, 18, 19]

$$\Gamma_{km} = sinc\left(\frac{2\pi f d_{km}}{c}\right) \tag{2.8}$$

where d_{km} the distance between sensors *k* and *m* and *c* the speed of sound.

2.2 DOA Estimation

As a first step in our enhancing process, we have to estimate the direction of arrival of the signal of interest in order to be able to perform robust array processing. While many methods for DOA estimation exist in the literature, the technique used to calculate the DOA is based on the algorithm proposed by Pavlidi et al. in [20]. According to this algorithm the DOA estimation is performed in frequency zones under the assumption that there is exactly one dominant source in every zone. A frequency zone is defined as a series of consecutive frequency bins and represented as Ω in the following equations.

The first step of the algorithm is to detect the single-source zones. For a pair of signals (x_i, x_j) of consecutive microphones we calculate the cross-spectral density of an analysis zone as:

$$R_{i,j}(\Omega) = \sum_{\omega} |X_i(\omega)X_j(\omega)| \tag{2.9}$$

end we derive the correlation coefficient for this pair of microphones as:

$$r_{i,j}(\Omega) = \frac{R_{i,j}(\Omega)}{\sqrt{R_i(\Omega)R_j(\Omega)}} \tag{2.10}$$

To detect a single-source zone, the mean of the correlation coefficients of all pairs of microphones has to exceed a certain threshold as shown in the next equation.

$$\overline{r_{i,j}(\Omega)} \; > \; \alpha \tag{2.11}$$

where α is set to 0.7 for our experiments.

After defining the single-source zones, the DOA estimation is conducted over these zone using the Circular Integrated Cross Spectrum (CICS) algorithm. For a pair of consecutive microphones, the phase of the cross-spectral density is

$$G_{m_i m_{i+1}}(\omega) \; = \; \frac{R_{i,i+1}(\omega)}{|R_{i,i+1}(\omega)|}, \quad \omega \in \Omega \tag{2.12}$$

We, also, calculate the Phase Rotation Factors [21] for every angle as:

$$G^{(\omega)}_{m_i \to m_1}(\phi) \; = \; e^{-j\omega t_{m_i \to m_1}(\phi)} \tag{2.13}$$

where $t_{m_i \to m_1}(\phi)$ can be calculated as $t_{m_i \to m_1}(\phi) \; = \; t_{m_1 m_2}(\phi) - t_{m_i m_{i+1}}(\phi)$ which is the relative time delay between the signal captured by the pairs of microphones $m_1 m_2$ and $m_i m_{i+1}$.

We proceed with the estimation of CICS as follows:

$$CICS^{(\omega)}(\phi) \; = \; \sum_{i=1}^{M} G^{(\omega)}_{m_i \to m_1}(\phi) \cdot G_{m_i m_{i+1}}(\omega) \tag{2.14}$$

The DOA estimation for each frequency component is extracted from:

$$\theta(\omega) \; = \; arg \max_{\phi} |CICS^{(\omega)}(\phi)| \tag{2.15}$$

After calculating the DOA for each frequency bin for each time-frame, we create a histogram with the DOA estimation including the DOA estimations of *k* consecutive time-frames. The number of consecutive frames in our experiments is 60 which results to a 0.6s history. Assuming that only one source is active, we extract the DOA estimation as the peak of the histogram.

2.3 Beamforming

Estimating the DOA of the speech source, enables us to process the multichannel speech signals while taking advantage of the directional information. To process the multichannel signals we use beamforming techniques. In this chapter we present the basic theory on beamforming including the most common techniques such as Delay and Sum (DAS), Superdirective and Minimum Variance Distortionless beamformer.

For simplicity, we may refer to $\mathbf{x}(\omega, t)$, $\mathbf{s}(\omega, t)$ and $\mathbf{n}(\omega, t)$ as **x**, **s** and **n** respectively.

Beamforming is a widely used array processing technique on sensor arrays for transmission and reception of audio signals. Multiple microphones are often combined in a specific array topology in order to take advantage of the directionality of the input signal. To accomplish that both amplitude and phase are processed accordingly. There are two main categories of beamformers, *signal-dependent* (also known as *adaptive*) and *signal-independent* (or *fixed*). As their name implies, the first category contains beamformers that take into consideration the statistics of the input signal, while the second category contains beamformers with fixed weights calculated in advance and remaining the same through time.

Let's assume a vector with time-frames of a captured multi-channel input signal **x** and a signal of interest s as presented in 2.3. Then in an arbitrary sensor array the output of the beamformer can be calculated as presented below [22]

$$\mathbf{y} = \mathbf{w}^H \mathbf{x} \tag{2.16}$$

where *H* denotes matrix Hermitian transpose and **w** represents the complex weights for each microphone $\mathbf{w} = \begin{bmatrix} w_1 & w_2 & \dots & w_M \end{bmatrix}^T$. Every complex weight coefficient can be written as $w_k = \alpha_k e^{j\beta_k}$ where α_k and β_k are the amplitude and phase components respectively. In the following sections we will discuss about how these components can be calculated using several beamforming techniques.

2.3.1 Delay and Sum Beamformer

The most basic form of beamforming is a simple signal-independent technique called Delay And Sum. This beamformer performs a time-alignment on the input signals and adds them to a point of interest (POI). Thus, the output is given by averaging over the aligned signals as shown below.

$$\mathbf{y} \;=\; \mathbf{w}^H \mathbf{x} \;=\; \sum_{n=1}^{N} e^{j\omega T_n} \mathbf{x}_n \tag{2.17}$$

The variable T_n expresses the time the acoustic wave needs in order to propagate from a microphone to the POI. Let assume a circular sensor setup with *M* equally placed microphones on a circle with radius *R* and a point of interest being the center of the circle. The time-difference between every microphone and the POI can be calculated as

$$T_n = \frac{dx_n}{c} = \frac{Rcos\big(\theta_s - (n-1)\theta_o\big)}{c}, \qquad n = 1, 2, \dots M \tag{2.18}$$

where *c* is the speed of sound, θ_s is the angle of the source relative to the microphone array, θ_o is the angle between two adjacent microphones M_n and M_{n+1} and *dx* is the distance the sound wave has to propagate as shown in the Figure 2.2. This solution is based on the assumption that the source lies in the far-field region which gives us the ability to handle the acoustic waves as multiple rays parallel to each other and perpendicular to the wavefront.

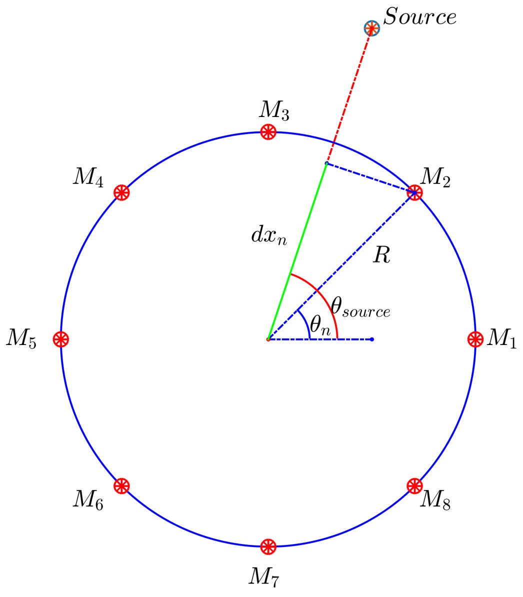


Figure 2.2: Circular microphone array with 8 elements

2.3.2 Superdirective Beamformer

As its name implies this beamformer focuses mostly on the directionality of the beam. To achieve high directionality, the algorithm takes into consideration not only the distance between the POI and the microphones but also the physical properties of the array and more specifically the distance between every pair of microphones. Nevertheless, this design has a drawback which is the poor White Noise Gain (WNG). The high directionality of the beam leads to excessive amplification of the white noise.

To achieve maximum directivity, the calculation of the weights for the Superdirective beamformer is based on the maximization of the Directivity Index (DI) of the array (more information on the Directivity Index will be given in later sections) which leads to the following optimization problem:

$$\max_{\mathbf{w}} \ 10log\frac{|\mathbf{w}^H\mathbf{d}|^2}{\mathbf{w}^H\mathbf{\Gamma}_{nn}\mathbf{w}} \tag{2.19}$$

The same problem can be written as [7]:

$$\min_{\mathbf{w}} \ \mathbf{w}^H \mathbf{\Gamma}_{nn} \mathbf{w} \qquad \textit{subject to} \qquad \mathbf{w}^H \mathbf{d}(\theta_s) = 1 \tag{2.20}$$

The minimization problem above implies that the optimal weights come from minimizing the PSD of the output when the input signal is diffuse noise [10]. The constraint, also, assures that the signal from the desired direction remains intact. Using equations 2.5 and 2.16, the PSD of the output signal, $\mathbf{\Phi}_{yy}$, equals to [23]:

$$\mathbf{\Phi}_{yy} \ = \ \mathbf{w}^H \mathbf{\Phi}_{nn} \mathbf{w} \ = \ \mathbf{w}^H \mathbf{\Gamma}_{nn} \mathbf{w} \tag{2.21}$$

The solution for the problem above is [19]:

$$\mathbf{w}_{SD} \ = \ \frac{\mathbf{\Gamma}_{nn}^{-1} \, \mathbf{d}(\theta_s)}{\mathbf{d}(\theta_s)^H \, \mathbf{\Gamma}_{nn}^{-1} \, \mathbf{d}(\theta_s)} \tag{2.22}$$

where $\mathbf{\Gamma}$ the noise coherence matrix and $\mathbf{d}(\theta_s)$ the steering vector steered towards direction θ_s. Since the matrix $\mathbf{\Gamma}$ includes small values, the inversion becomes an ill-conditioned problem. To cope with this, a scalar λ with small value is added to the diagonal of $\mathbf{\Gamma}$. The smaller this value is the more narrow the beam becomes and the more ill-condition the problem becomes. The final expression of the beamformer's weights becomes [24]:

$$\mathbf{w}_{SD} \ = \ \frac{(\mathbf{\Gamma}_{nn} + \lambda I)^{-1} \, \mathbf{d}(\theta_s)}{\mathbf{d}(\theta_s)^H \, (\mathbf{\Gamma}_{nn} + \lambda I)^{-1} \, \mathbf{d}(\theta_s)} \tag{2.23}$$

The coefficient λ has to be chosen with caution and be close to the diagonal elements of $\mathbf{\Gamma}$. If the value is significantly larger than the diagonal of $\mathbf{\Gamma}$ then $(\mathbf{\Gamma}_{nn} + \lambda I) \approx \lambda I$ and the equation 2.23 will be simplified to $\mathbf{w}_{SD} \ = \ \mathbf{d}(\theta_s)$ and thus the Superdirective beamformer becomes a Delay and Sum beamformer.

2.3.3 Minimum Variance Distortionless Response Beamformer

Another well-known beamformer is the Minimum Variance Distortionless Response beamformer, also known as Capon beamformer [25]. Despite the two previous signal independent beamformers, Delay And Sum (DAS) and Superdirective (SD), this beamformer is signal dependent. In other words, the calculation of the weights relies on the statistics of the frame we examine and more specifically on the cross spectral density between all sensors' time-frames.

The main function of the beamformer includes the minimization of the energy of the output signal while maintaining unitary gain to the desired direction [7]. More specifically, the output energy is:

$$E \;=\; E\big[|\mathbf{y}|^2\big] \;=\; E\big[|\mathbf{w}^H\mathbf{x}|^2\big] \;=\; \mathbf{w}^H E\big[\mathbf{x}\mathbf{x}^H\big]\mathbf{w} \;=\; \mathbf{w}^H \boldsymbol{\Phi}_{xx}\mathbf{w} \tag{2.24}$$

Consequently, the minimization problem becomes:

$$\min_{\mathbf{w}} \; \mathbf{w}^H \boldsymbol{\Phi}_{xx}\mathbf{w} \qquad subject\,to \qquad \mathbf{w}^H\mathbf{d}(\theta_s) = 1 \tag{2.25}$$

The solution for this minimization problem is:

$$\mathbf{w}_{MVDR} \;=\; \frac{\boldsymbol{\Phi}_{xx}^{-1}\,\mathbf{d}(\theta_s)}{\mathbf{d}(\theta_s)^H\,\boldsymbol{\Phi}_{xx}^{-1}\,\mathbf{d}(\theta_s)} \tag{2.26}$$

As we discussed in the introduction of this chapter, the power spectral density of the input signal $\boldsymbol{\Phi}_{xx}$ can not be accurately calculated using only one time-frame. For this reason we used multiple neighboring frames to acquire an estimation of $\boldsymbol{\Phi}_{xx}$. Also, the same matrix is ill-conditioned which means it can be non-invertible in some occasions or the inversion may lead to big values. To deal with this problem a constant scalar λ is added to the diagonal of $\boldsymbol{\Phi}_{xx}$ [7]. The final equation then becomes:

$$\mathbf{w}_{MVDR} \;=\; \frac{\left(\boldsymbol{\Phi}_{xx} + \lambda I\right)^{-1}\mathbf{d}(\theta_s)}{\mathbf{d}(\theta_s)^H\left(\boldsymbol{\Phi}_{xx} + \lambda I\right)^{-1}\mathbf{d}(\theta_s)} \tag{2.27}$$

We have to mention again that choosing the number of frames for $\boldsymbol{\Phi}_{xx}$ and the scalar λ are very crucial. Both of them are able to change significantly the effectiveness of the beamformer.

2.4 Beamformer Evaluation

In this section we discuss about some of the properties of each beamformer such as *Spatial Aliasing*, *Spatial Directivity Pattern*, *Directivity Index* and *White Noise Gain*. These aspects will help us to design the most suitable beamformer for our application.

2.4.1 Spatial Aliasing

In array processing, a captured signal can be reconstructed correctly under certain circumstances. The frequency of the captured signal has to be bigger than the aliasing frequency of the array which depends on the physical aspects of it. More specifically, the aliasing frequency depends on the minimum distance between every pair of microphones. Assuming an array with M microphones the captured signal has to be within the range $(0, f_{max})$ where

$$f_{max} \;=\; \frac{c}{\lambda_{min}} \tag{2.28}$$

Similarly to Nyquist's theorem, spatial aliasing occurs when the distance between a pair of consecutive microphones $M_n M_{n+1}$ is two times smaller than minimum wavelength [18]. In a circular array as in figure 2.2 this distance is expressed as

$$d_{M_n M_{n+1}} = 2R\sin\left(\frac{\theta_n}{2}\right) = 2R\sin\left(\frac{\pi}{M}\right) \tag{2.29}$$

where *R* the radius of the array and θ_n the angular distance between two sensors. The equation 2.28 then becomes

$$f_{max} = \frac{c}{2d_{M_n M_{n+1}}} = \frac{c}{4R\sin\left(\frac{\pi}{M}\right)} \tag{2.30}$$

In the course of this thesis we use an 8 microphone array in a circular topology with radius 0.1m. The aliasing frequency is then 2222 Hz.

2.4.2 Spatial Directivity Pattern

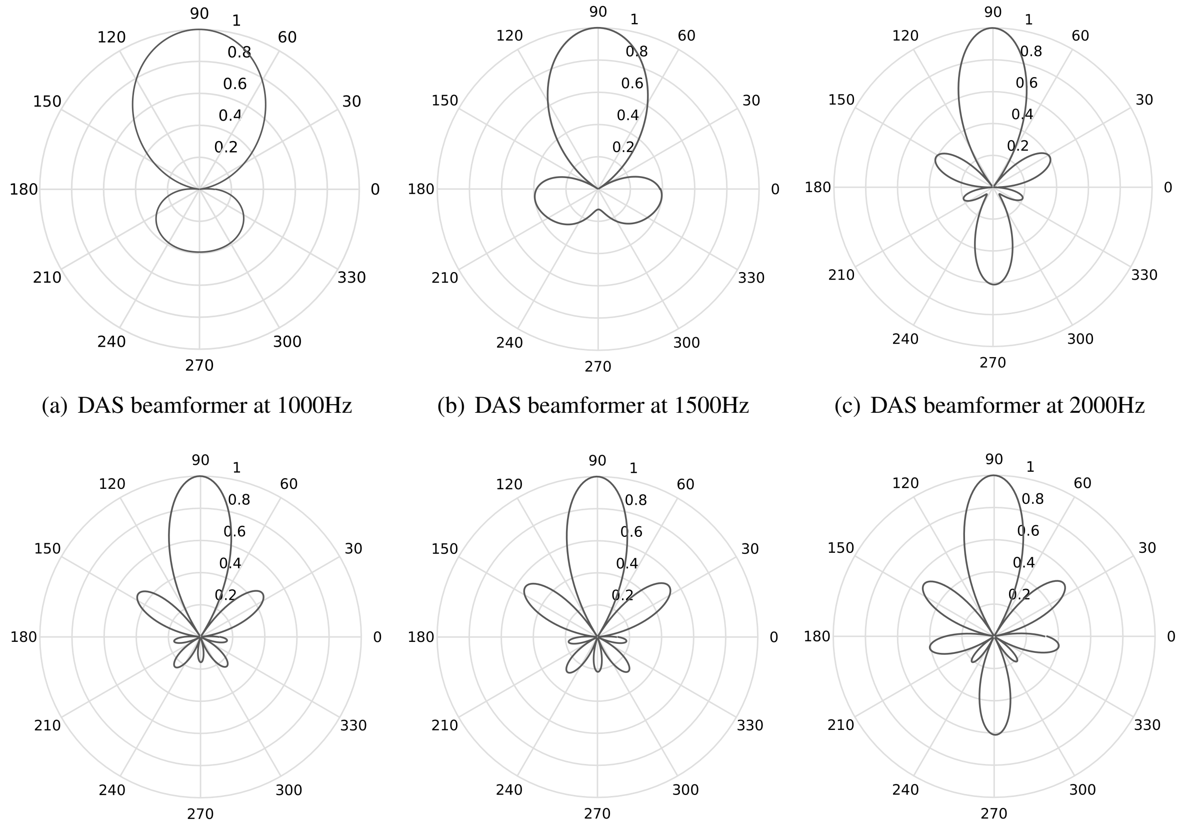
Spatial Directivity Pattern or *beam pattern* [19] is the response of the sensor array combined with the beamformer's weights towards plane waves coming from all directions. For a given set of weights $\mathbf{w}(\theta_s)$ steered to the direction θ_s, the sensitivity at a specific frequency f_o can be calculated as [26]:

$$\mathbf{B}(\theta) = \mathbf{w}^H(\theta_s)\mathbf{d}(\theta) \tag{2.31}$$

Depending on the beamformer we choose the shape of the directivity pattern can significantly change. It can make the response more or less directional by creating a wider or narrower main lobe respectively. The figure 2.3 shows the beam pattern for Delay And Sum and Superdirective beamformer. We can observe that the first and less sophisticated beamformer has much wider main lobe compare to the second beamformer in all presented frequencies. That means it is less capable to isolate the signal from the background noise. This is expected as the main goal of the superdirective beamformer is to be as directive as possible, as its name implies.

2.4.3 Directivity Index

To evaluate the ability of a beamformer to suppress the background noise we use two main expressions. The first expression called *Directivity Index* measures the performance of the beamformer towards an isotropic diffuse noise field and the second expression called *White Noise Gain* measures the performance towards white uncorrelated noise.



(a) DAS beamformer at 1000Hz (b) DAS beamformer at 1500Hz (c) DAS beamformer at 2000Hz

(d) SD beamformer at 1000Hz (e) SD beamformer at 1500Hz (f) SD beamformer at 2000Hz

Figure 2.3: Beam pattern of DAS and SD beamformer for incoming signal from 90° direction of 1000Hz, 1500Hz and 2000Hz frequency. 8 microphones placed in a circular topology with 0.1m radius.

In case of isotropic noise field, we measure the *Array Gain* (AG) as the SNR ratio between the the signal captured by one microphone,SNR_{sensor}, and the output signal,SNR_{output}.

$$AG = \frac{SNR_{output}}{SNR_{sensor}} \tag{2.32}$$

Because of STFT, we assume stationarity to all signals [10, 27, 28] and thus the equation 2.32 becomes

$$AG = \frac{|\mathbf{w}^H \mathbf{d}|^2}{\mathbf{w}^H \mathbf{\Phi}_{nn} \mathbf{w}} \tag{2.33}$$

where Φ_{nn} is the cross-spectral density of the noise signal. In equation 2.6 we showed the form of cross-spectral density in case of isotropic noise field as long as the simplified version for a circular microphone array in equation 2.8. The Array Gain is also known as *Directivity Factor* (DF). Presenting the DF in logarithmic scale will give us the *Directivity Index* (DI).

$$DI = 10\log\frac{|\mathbf{w}^H\mathbf{d}|^2}{\mathbf{w}^H\mathbf{\Gamma}_{nn}\mathbf{w}} \tag{2.34}$$

The figure 2.4 shows the DI for both DAS and SD beamformer. As expected the SD beamformer has more directive characteristics than the DAS beamformer especially in frequencies lower than the aliasing frequency. The aliasing frequencies for the three arrays with radiuses 0.1m, 0.05m and 0.02m are 2222Hz, 4444z and 11110Hz respectively.

2.4.4 White Noise Gain

Another metric to measure the performance of the beamformer is called *White Noise Gain*. This measure indicates the ability of the beamformer to suppress the white uncorrelated noise, also known as thermal noise, captured by all microphones during recording.

The expression to calculate WNG is similar to the equation we use to calculate the Array Gain (equation 2.33), although for white uncorrelated noise field the coherence matrix Φ_{nn} becomes a unitary matrix as there is no coherence between sensor signals. The expression 2.33 then becomes

$$WNG = \frac{|\mathbf{w}^H\mathbf{d}|^2}{\mathbf{w}^H\mathbf{w}} \tag{2.35}$$

The figure 2.4 shows the WNG for circular arrays with different sizes. As we can see, the size of the array changes the directivity of the SD beamformer but makes it less robust against white noise.

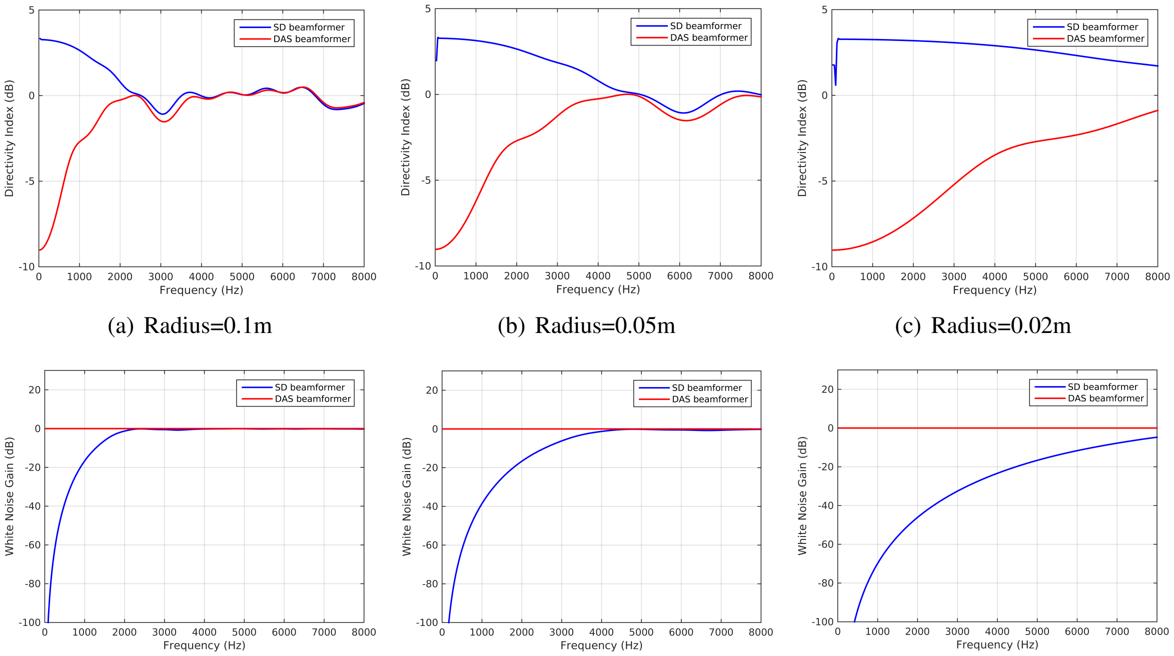
2.5 Postfiltering

A postfilter is a signal processing technique usually applied after a beamformer to further enhance the output signal since applying only a beamformer is not adequate. In recent years, different versions of postfilters are deployed in speech recognition tasks alongside with beamformers to achieve better performance. In the following section, two main postfilters based on binary masks and Wiener postfiltering are presented.

2.5.1 Binary Masks

A well established postfilter for source separation is based on binary masks [29]. This postfilter is an easy and efficient technique to enhance the output of a beamformer. Studies have shown that binary masks perform well, especially in overlapping speech [3, 30].

In a scenario with S sources talking to a microphone array, we assume that in ever time-frame at



(a) Radius=0.1m

(b) Radius=0.05m

(c) Radius=0.02m

(d) Radius=0.1m

(e) Radius=0.05m

(f) Radius=0.02m

Figure 2.4: Directivity Index and White Noise Gain of a circular microphone array consisted of 8 microphones. (a) Directivity Index of a UCA with radius 0.1m, (b) Directivity Index of a UCA with radius 0.05m, (c) Directivity Index of a UCA with radius 0.02m, (d) White Noise Gain of a UCA with radius 0.1m, (e) White Noise Gain of a UCA with radius 0.05m, (f) White Noise Gain of a UCA with radius 0.02m.

a specific frequency bin only one source is dominant. Following, based on a criterion, we assign that frequency bin to the appropriate source and eliminate the same frequency bin from all the other sources by putting zeros in their frequency spectrum. More specifically, we apply the steered beamformer to the direction of each source in order to acquire the enhanced time-frame $\tilde{y}_i(\omega, t)$ where $i = 1 \dots S$. The binary mask for each source is then defined as

$$b_j(\omega, t) = \begin{cases} 1, & \text{if } |\tilde{y}_j(\omega, t)| = \max\limits_{1 \le k \le S} |\tilde{y}_k(\omega, t)| \\ 0, & \text{otherwise} \end{cases}, \quad j = 1, \dots, S \tag{2.36}$$

The signal for each source is then given by $y_j(\omega, t) = b_j(\omega, t)\tilde{y}_j(\omega, t)$. In order for this postfilter to be used we need more than one source, for that reason we always assume two sources in all experiments, even in those with only one speaker. In the experiments with only one speaker an imaginary speaker is considered to be at the exact opposite direction.

Despite the fact that binary masks enhance effectively the output signal, in some cases they introduce some kind of distortion. This distortion occurs because of the gaps in the frequency spectrum and can be more prominent when the energy of one speaker is significantly greater than the other. To the human ear it is usually perceived as bubble noise.

2.5.2 Wiener Postfilter

Another postfilter we are going to use is a version of a Wiener postfilter. For an input signal $x(\omega, t)$ and output beamformed signal $y(\omega, t)$ as in equation 2.16, a Wiener postfilter is calculated by minimizing the mean square error, $E\big[|s(\omega,t)\ -\ \tilde{s}(\omega,t)|^2\big]$ where $s(\omega,t)$ is the desired signal and $\tilde{s}(\omega,t)$ the signal we get after both the beamformer and the postfilter. The optimum solution for the above minimization problem is given by [31]

$$\tilde{s}(\omega,t)\ =\ \mathbf{w}_{opt}(\omega,t)\mathbf{x}(\omega,t)\ =\ \phi_{ss}(\omega,t)\mathbf{d}(\omega)\mathbf{\Phi}_{xx}^{-1}(\omega,t)\mathbf{x}(\omega,t) \tag{2.37}$$

where $\phi_{ss}(\omega,t)$ the power spectral density as in equation 2.7 and $\mathbf{\Phi}_{xx}(\omega,t)$ the cross spectral density matrix as in equation 2.5. Assuming an MVDR beamformer and the output signal $y(\omega,t)$ as in 2.27, then the power spectral density ϕ_{yy} is given by

$$\phi_{yy}(\omega,t)\ =\ \frac{1}{\mathbf{d}^H(\omega,t)\mathbf{\Phi}_{xx}^{-1}(\omega,t)\mathbf{d}(\omega,t)} \tag{2.38}$$

Therefore, combining the equations 2.37 and 2.38 the output signal $\tilde{s}(\omega,t)$ can be written as

$$\tilde{s}(\omega,t)\ =\ \underbrace{\frac{\phi_{ss}(\omega,t)}{\phi_{yy}(\omega,t)}}_{Wienerfilter}\cdot\underbrace{\frac{\mathbf{d}^H(\omega)\mathbf{\Phi}_{xx}^{-1}(\omega,t)}{\mathbf{d}^H(\omega)\mathbf{\Phi}_{xx}^{-1}(\omega,t)\mathbf{d}(\omega)}}_{w_{MVDR}}\cdot\mathbf{x}(\omega,t) \tag{2.39}$$

The unknown components in the above expression are $\phi_{ss}(\omega,t)$ and $\phi_{yy}(\omega,t)$. Here, we follow the method in [32] proposed by Nobutaka Ito to calculate them.

It is known that the cross spectral density between two microphones M_k and M_m is given by the equation 2.7 which leads us to $\phi_{km}(\omega,t)\ =\ \phi^*_{mk}(\omega,t)$. However, when assuming a symmetric circular microphone array, the cross spectral density of the noise signal of the microphones is given by the equation 2.8. As we can see it depends only on the distance between the microphones, as a result

$$l_{mn} = l_{nm} \quad \Rightarrow \quad \phi_{n_m n_n}(\omega,t) = \phi_{n_n n_m}(\omega,t) \tag{2.40}$$

where l_{mn} the distance between the microphones M_m and M_n. Consequently, the cross spectral density belongs to the set of the real numbers $\mathbb{R}$.

For a plane wave $\mathbf{x}(\omega, t)$, the cross spectral density between two microphones is [31]

$$\begin{aligned} \phi_{x_k x_l}(\omega, t) &= \phi_{ss}(\omega, t) d_k(\omega) d_l^*(\omega) &+ \phi_{n_k n_l}(\omega, t) \\ &= \phi_{ss}(\omega, t) \exp^{-j\omega(\delta_k - \delta_l)} &+ \phi_{n_k n_l}(\omega, t) \end{aligned} \tag{2.41}$$

We solve the above equation for $\phi_{ss}(\omega, t)$ and keep the imaginary part. We know that the element $\phi_{n_k n_l}(\omega, t)$ is a real scalar, as a result the final equation becomes

$$\phi_{ss} = -\frac{\sum\limits_{m>n} \sin[\omega(\delta_m - \delta_n)] \Im[\phi_{x_m x_n}(\omega, t)]}{\sum\limits_{m>n} \sin^2[\omega(\delta_m - \delta_n)]} \tag{2.42}$$

The second unknown element we need in order to define the Wiener filter is $\phi_{yy}(\omega, t)$. We calculate the $\phi_{yy}(\omega, t)$ approximately by using the proposed equation in Zelinski's article [33] which in practice is the mean of the power spectral density of all microphones.

$$\phi_{yy}(\omega, t) \triangleq \frac{1}{M} \sum_{m=1}^{M} \phi_{x_m x_m}(\omega, t) \tag{2.43}$$

We now have everything we need to calculate the desired signal $\tilde{s}(\omega, t)$ by using the expression 2.39. Despite the fact that this postfilter was initially constructed for use with MVDR beamformer, we also tried it with Delay and Sum and Superdirective beamformers.

2.6 Automatic Speech Recognition Tools

An ASR system consists of two main components: the front-end and the back-end. The front-end component includes the signal processing techniques as described in this chapter. The back-end component includes the acoustic model (AM), a statistical model representing the phonemes, and the language model (LM) which is a model that represents the syntax of the input data.

Since the front-end was described previously,we will now analyze the parts of the back-end component and give some information about their specifications. The tools we used for the back-end are provided by the European Media Laboratory GmbH (EML) [34].

- The first component is the **acoustic model**. For the training of the acoustic model tandem features are used to train an HMM-GMM model with the following characteristics: single state with repetitions, 3-state phone modelling, triphone context, tied covariance matrix and fixed, variable transition probabilities and variable number of states. The lexicon we used is called Beep [35]. It contains about 250,000 English words and their phonetization.

- Apart from the acoustic model, a **language model** also contributes to the prediction of the recognized word. The LM defines the probability of a word or a sequence of words based on the training data we provide. The two most common types of LM are trained either based on a grammar or based on a large vocabulary. Our focus is on the latter which may take the form of a N-gram. An N-gram defines the probability of appearance of N or less adjacent words, calculated from the text used for training. In our experiments we use a 4-gram LM trained with all the transcriptions from the WSJCAM0 database. It is important that the context of the text of the training data has to be on the same topic with the test data.

- Even though the acoustic and language models are the most important components of the back-end system, they have to be combined in a proper way in order to achieve the best possible result. For this purpose, EML provided us with a tool called the **decoder** which gave us access to three parameters towards this scope. These parameters are AmScale, LmScale and LmFactor. The first two parameters, AmScale and LmScale, are the weights we assign to the score from the acoustic model and the language model respectively and indicate our certainty on the AM and the LM. The AmScale take values from [0,1] range and the LmScale take integer values with the default value of 34 given by EML. The figure below shows how they are combined with the AmScore and LmScore. The last parameter, LmFactor, concerns the language model and represents how many words we will examine from the LM, also known as LM pruning. In practice, the bigger the number the more options we examine from the LM.

  The output of the decoder is the prediction of the text uttered. In order to calculate the statistics about the performance of the system we use the scoring toolkit sclite [36] which gives us the ability to calculate the Insertions, Deletions, Substitutions, Word Correct Rate (WCR) and Word Error Rate (WER).

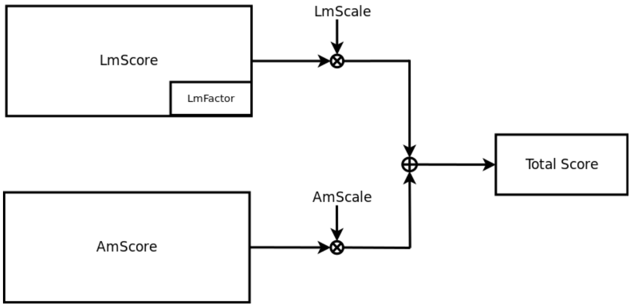


Figure 2.5: Decoder structure, hypothesis evaluation and combination of LmScore and AmScore

Chapter 3

Databases and Methodology

In this chapter, we will describe our proposed method and the experiments conducted in this work, as well as the data and parameters used in each one of them.

3.1 Databases

Two of the most well-known and widely used speech databases for speech recognition are WSJCAM0 and MC-WSJ-AV. These databases are available from the Linguistic Data Consortium (LDC) [37] for research and development purposes. Their content is taken from the Wall Street Journal (WSJ).

3.1.1 WSJCAM0

The first database, WSJCAM0, is a monophonic UK English version of a subset of the American English WSJ0 database [38]. The name of the database comes from the Wall Street Journal where the speech content comes from and the University of CAMbridge where it was recorded.

The participating speakers are divided into 3 groups, training, development and evaluation sets. The training set consists of 92 speakers who dictated 90 utterances randomly selected from the WSJ0 corpus. The same sentence might be dictated by different speakers but can be dictated by the same speaker only once. The development group consists of 20 speakers who also dictated 90 sentences each. The evaluation group is divided into two subgroups, a subgroup of 14 speakers who dictated 90 sentences from a 5,000 words vocabulary and a subgroup of 14 people who dictated 90 sentences from a 20,000 words vocabulary. In addition to that, each speaker was recorded on a common set of 18 sentences for adaptation purposes. All recordings were acquired from both a close-talking-microphone (CTM) and a distant microphone placed on the table in front of the speaker at around 0.5m. More information about the capturing equipment and setup can be found in [39, 40].

WSJCAM0 Database	Speakers	Utterances	Adaptation Utterances
Training Group	91	90	18
Development Group	20	90	18
Evaluation Group (5k voc)	14	90	18
Evaluation Group (20k voc)	14	90	18

Table 3.1: Structure of WSJCAM0 database

3.1.2 MC-WSJ-AV

MC-WSJ-AV (Multichannel Wall Street Journal Audio-Visual) is a multichannel speech database that is recorded with both a CTM and a distant microphone array. Even though "Audio-Visual" is a part of the name, there is no video included in the database. The sentences are taken from the Wall Street Journal and they are a subset of the database described above (WSJCAM0). The purpose of this database is to enable researchers to conduct experiments on source localization, source separation and speech recognition with real speech data.

The database is divided into three datasets. The development set (DEV) and two evaluation sets (EVAL1 and EVAL2). The DEV set (also referred as adaptation set) contains the same 14 sentences for all speakers and the EVAL1(5k) and EVAL2(20k) sets contain 40 randomly selected sentences from a 5,000-words corpus and a 20,000-words corpus respectively.

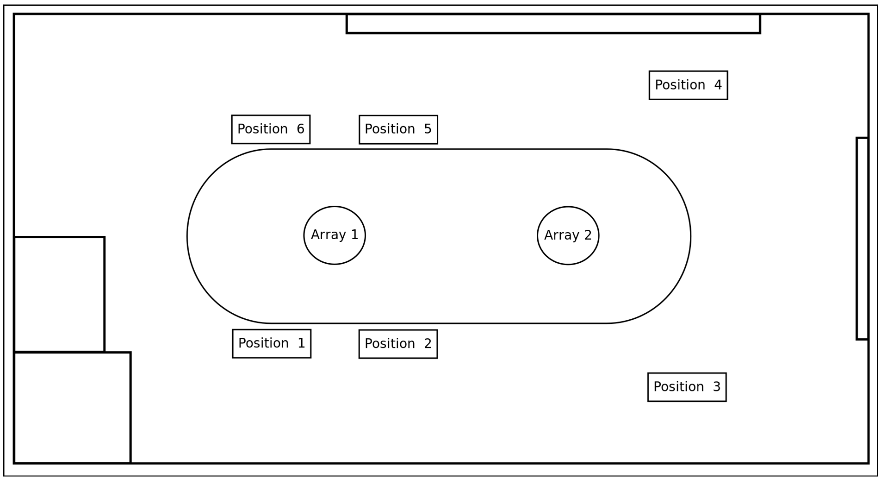


Figure 3.1: Meeting room with speaker and array positions during the recording of MC-WSJ-AV database.

The total amount of speakers is 42 and each one of them contributes to all datasets. More specifically, every speaker reads the same set of 14 sentences for the DEV set, 40 sentences for the EVAL1 set and 40 sentences for the EVAL2 set. Each sentence of the EVAL1 and EVAL2 may be dictated by more than one speaker but every speaker dictates one sentence only once. Also, every speaker is assigned only to one experiment i.e. a speaker from the stationary-speaker experiment does not participate in the moving-speaker experiment. The list of the experiments is the following.

Experiment	Speakers	Description
Stationary Speaker (stat)	15	Single stationary speaker
Moving Speaker (mov)	9	Single moving speaker
Overlapping Speakers (olap)	18 (9 pairs)	Two stationary speakers talking simultaneously

Table 3.2: Experiments of MC-WSJ-AV database and speakers' assignment to each experiment

The goal of this database is to be utilized for research and development purposes in distant speech recognition. For that reason speech is captured with different devices providing both clean and noisy recordings for developing and testing. More specifically, each sentence is captured by two microphone arrays, a headset and a lapel microphone. Each array consists of 8 microphones placed on a circle with radius 0.1m. The arrays are placed on a table as shown in figure 3.1

The recordings take place in a meeting room at the Centre for Speech Technology Research at the University of Edinburgh [41]. As in every meeting room there is plenty of background noise coming from devices such projectors and computers but also from activity outside the room. In addition, the reverberation is quite high and in this case T60 [42] is around 0.7s. These factors will degrade the performance of the ASR system as we will see later.

MC-WSJ-AV												
	Array 1			Array 2			Headset			Lapel		
	5k	20k	adap	5k	20k	adap	5k	20k	adap	5k	20k	adap
Stationary Speaker	1	1.05	0.35	1	1.05	0.35	1	1.05	0.35	1	1.05	0.35
Moving Speaker	0.5	0.6	0.2	0.5	0.6	0.2	0.5	0.6	0.2	0.5	0.6	0.2
Overlapping Speakers	0.7	0.7	0.2	0.7	0.7	0.2	0.7	0.7	0.2	0.7	0.7	0.2
							0.7	0.7	0.2	0.7	0.7	0.2

Table 3.3: Amount of speech data in h (hours) for the MC-WSJ-AV database.

The speakers positions for all three experiments are predefined as shown in the figure 3.1. In the positions 1,2,5,6 the speaker sits on a chair and his mouth is at the same level with the array while in positions 3,4 the speaker stands up to simulate an alleged presentation. Furthermore, for stat and olap experiment the speaker sits randomly at one of the predefined positions while in the mov experiment a speaker walks back and forth 3 times between two consecutive position dictating one sentence at a time (6 sentences in total).

As this database includes several experiments, testsets and recording equiments, it will be easier to understand the structure of the database from the table 3.3. This table shows the amount of speech data in h (hours). Note that for the overlapping speakers two headset and two lapel microphone signals are provided. More information about the recording procedure and the speech can be found in [43, 44]

3.2 REVERB Challenge

The REVERB (REverberant Voice Enhancement and Recognition Benchmark) Challenge [45] was a challenge that took place in 2014 and the main goal was to provide a universal framework to all researchers to evaluate their signal processing techniques and their ASR techniques. More specifically the framework was provided including a common database and common evaluation metrics for all. The challenge includes one scenario in which a single stationary distant-talking speaker is captured by one microphone (1ch), two microphones (2ch) and eight microphones (8ch).

The challenge provided two tasks to the participants: the Speech Enhancement (SE) and the ASR task. The first task aimed to provide objective and subjective evaluation metrics to determine the performance of every proposed technique. The objective metrics included: cepstrum distance between enhanced and reference signals, log likelihood ratio between enhanced and reference signals, frequency-segmental SNR, speech to reverberation energy ratio and computational cost. The subjective metrics included a MUSHRA (MUltiple Stimuli with Hidden Reference and Anchor) test where the listeners were asked to assess the perceived distance and the overall speech quality by hearing both processed and clean speech signals.

The second task was the ASR task. According to challenge's instructions the participants could use their own ASR systems which may be entirely different from the baseline system provided by the challenge. Changes in multiple levels are acceptable like different features, sophisticated multichannel front-end processing, different training criteria etc. However, the datasets were common for every participant. In the course of this thesis we focused on the eight microphone speech signal scenario only and we worked on the second task which includes the ASR task.

The data of the challenge were grouped in three sets: the training set, the development test set and the evaluation test set. The first two were published prior to the evaluation test set in order to permit the participants to train their acoustic models and test them using the development test set. Afterwards, the evaluation test set was provided to participants to assess their ASR system and present their final results.

The microphone array used to capture the multichannel signals was a circular array consisted of eight omnidirectional microphones with 10cm radius.

The evaluation test set contains both simulated (SimData) and real data (RealData). The SimData are created from clean speech signals from the WSJCAM0 database convolved with multiple room impulse responses (RIRs). The RIRs are measured from three different rooms with reverberation times 0.3, 0.6 and 0.7 s. Each of the reverberation times corresponds to the rooms Room1(small size), Room2(medium size) and Room3(large size) respectively. The RIRs for each room included two occasions with the speaker talking close to the array (50cm) and far (200cm) from the array. The RIRs where measured with the same microphone array presented above. In addition to the reverberation, 20dB white noise was also added to the multichannel signals to simulate the imperfections of the microphones.

The RealData are taken from the MC-WSJ-AV database. They are separated into two groups, Far and Near, indicating the position of the speaker relatively to the position of the array. The distances between the speaker and the microphone array are 100cm and 250cm in Near and Far occasions respectively. In these recordings, the reverberation time is estimated to be 0.7s. The following table summarizes all the evaluation test sets as they are provided by the challenge. All test sets, both SimData and RealData, include data only from the 5k vocabulary.

In addition to the test sets provided by the challenge, we added three more test sets taken from the MC-WSJ-AV database. More specifically, the first two test sets included a single moving speaker captured by the Array1 and Array2 respectively while the third test set included overlapping speakers.

Test sets of REVERB Challenge				Added Test sets	
SimData-Room1 Near Condition	SimData-Room2 Near Condition	SimData-Room3 Near Condition	RealData Near Condition	Moving Speaker Array1	Overlapping Speakers
SimData-Room1 Far Condition	SimData-Room2 Far Condition	SimData-Room3 Far Condition	RealData Far Condition	Moving Speaker Array2	

Table 3.4: Test sets of the REVERB challenge

In the context of the REVERB challenge, the participants had the opportunity to work on three different acoustic models: a clean model, a multi-condition model and an extended model. The participants were allowed to use their own tools to train their acoustic models. The first clean model was trained with the clean training set taken from WSJCAM0 database. The second multi-condition model was trained with a training set which was generated by convolving clean signals from WSJCAM0 database with 24 different room impulse responses while adding 20dB background noise. The RIRs include various reverberation times from 0.2s to 0.8s. The participants where allowed to choose the front-end applied to both training and test sets. It is important to mention that the RIRs that were applied to the training set were different from those applied to the test sets. The last acoustic model was the extended model. In this case, the participants where allowed to combine both clean training set and training sets processed with different front-ends.

3.3 Acoustic Models

In this section we present the initial acoustic models we trained based on the guidelines of the REVERB challenge using several front-end processing techniques.

3.3.1 Clean Acoustic Model

The first model we trained was a clean acoustic model as described in the REVERB challenge. The training data used were clean speech signals taken from the training set of the WSJCAM0 database. The total amount of training data was about 17h.

3.3.2 Multi-condition Acoustic Model

The second acoustic model we trained was the multi-condition model. The training data for this model where taken from WSJCAM0 database and simulated in multiple rooms as we described in the section 3.2. The amount of training data is 17h, as in the clean model. For this model the front-end we used on the training data was a superdirective beamformer followed by binary masks.

3.3.3 Extended Acoustic Model

The third acoustic model we trained based on the REVERB challenge was an extended model. In the extended model, there was no constraint on the training data. The data we picked for training included:

- the clean training set of the WSJCAM0 database (17h)
- processed training set of the multi-condition model with 20dB background noise (17h)
- processed training set of the multi-condition model with 15dB background noise (17h)
- processed training set of the multi-condition model with 10dB background noise (17h)

The three instances of the training set of the multi-condition model where processed with superdirective beamformer combined with binary masks. The total amount of training data sums up to 68h.

As we will see later in the evaluation chapter, the last extended model is superior to the clean model and the multi-condition model in terms of recognition results. Having established the best combination of training data, we proceed to the next section where we experiment only with the several processing techniques we chose from the bibliography, in order to find the most robust technique for our task.

3.4 Adaptation

Apart from the tools for training and testing, EML also provided us with a tool for adapting. As we know training an acoustic model is a time-consuming and computationally expensive task. On the other hand, adaptation is a fast and lightweight procedure that incorporate additional speech information without retraining the whole model. For that reasons, adapting speech data to an already trained model is usually preferred before retraining in order to check the tendency of the performance with the additional data. However, adaptation can sometimes distort the initial acoustic model instead of improving it.

The acoustic model we performed adaptation to was the multi-condition model presented in section 3.3.2. As we will see in the evaluation chapter, the multi-condition model performs better than the clean model, so we decided that this model was more robust to proceed with adaptation. Moreover, in order to eliminate other factors like the big diversity of the training data of the extended model, we decided not to use that model for adaptation.

As we know, the real test sets come from the MC-WSJ-AV database and the training data set for the multi-condition model comes only from the WSJCAM0 database. Thus, the microphone mismatch between the two becomes apparent and inevitably results in a downgraded WER. To overcome this problem we divided the MC-WSJ-AV database in half and used the first half for testing and the other half for adaptation. The data we used for adaptation come from the microphone array signals of the MC-WSJ-AV database processed with the same superdirective beamformer and binary masks as the training set of the original multi-condition model. The total amount of adaptation data is 3.1h. The following table shows how the division between adaptation set and test set was made.

	Stationary Speaker	Moving Speaker	Overlapping Speakers
Adaptation set	T6, T7, T8, T9	T11, T13, T14, T15	T1, T2, T3, T4, T5
Test set	T21, T22, T23, T24, T25, T36, T37, T38, T39, T40	T26, T27, T28, T29, T30	T16, T17, T18, T19, T20

Table 3.5: Speakers of the MC-WSJ-AV database grouped into adaptation and test set.

3.5 Data Imputation

After experimenting with several front-end techniques including beamformers and postfilters, we ended up to the conclusion that although there was some improvement in the performance of the recognition in terms of WER, nevertheless the amount of distortion introduced by all these techniques was not negligible. Thus, still affected the results considerably. The most prominent distortion that we had to deal with was the musical noise caused by the binary masks. To cope with this problem, we applied a data imputation algorithm, in order to "fill in" the gaps in the frequency spectrum introduced by the binary masks. The data imputation algorithm we implemented is presented in [46].

This method is divided into two steps. The training phase and the "fill in" phase. During the first phase, a set of centroids is created calculated based on reference signals while in the second phase the missing data are imputed based on those centroids.

In the training phase, we used the k-means algorithm to calculate a set of centroids based on clean speech signals from the WSJCAM0 database. Frame analysis was performed to all speech signals to formulate a matrix with 250,000 rows. This matrix was then used as an input in k-means algorithm. Since only the spectral information from every frame was needed, we kept only the normalized amplitude of each frame using the equation below. The acquired $\tilde{\mathbf{x}}(\omega, t)$ were entered to the k-means algorithm to create the set of 1024 centroids.

$$\tilde{\mathbf{x}}(\omega, t) \;=\; \log\left[\frac{|\mathbf{x}(\omega, t)|}{\sqrt{\sum_{\omega} |\mathbf{x}(\omega, t)|^2}}\right] \tag{3.1}$$

The next phase is the imputation phase. During this phase every speech signal is processed with the superdirective beamformer followed by a postfilter and the gaps in the frequency spectrum are filled up based on the closest centroid. More specifically, let's assume the processed frame $\tilde{\mathbf{s}}(\omega, t)$, we normalize it using the equation 3.1 and we save the normalization factor on the denominator, $A = \sqrt{\sum_{\omega} |\mathbf{s}(\omega, t)|^2}$, for the reconstruction step. Next, the closest centroid has to be defined using Euclidean distance between every centroid and the frame we examine. To find the closest centroid we have to minimize the following expression:

$$\min_{c} ||\tilde{\mathbf{s}}(\omega, t) - \lambda \mathbf{c}(\omega, :)||^2 \tag{3.2}$$

where $\tilde{\mathbf{s}}(\omega, t)$ is the frame we examine, $\mathbf{c}(\omega, :)$ is the centroid and λ is a scalar to align the two vectors. The solution to λ is given by

$$\lambda \;=\; \min_{c}\left(\frac{\tilde{\mathbf{s}}(\omega, t)^T \mathbf{c}(\omega, :)}{\sum \mathbf{c}(\omega, :)^2}\right) \tag{3.3}$$

Both expressions 3.2 and 3.3 are minimized only for the frequency bins that are not zeros. Having already the scalar λ, we find the best $\mathbf{c}(\omega, :)$ from the equation 3.2 and we fill the zeros on the frame we examine, $\tilde{\mathbf{s}}(\omega, t)$, with the values $\lambda \mathbf{c}(\omega, :)$. The phase is kept the same as it was before applying the binary masks. The following figure shows an example of a frame with missing frequency bins and the closest centroid.

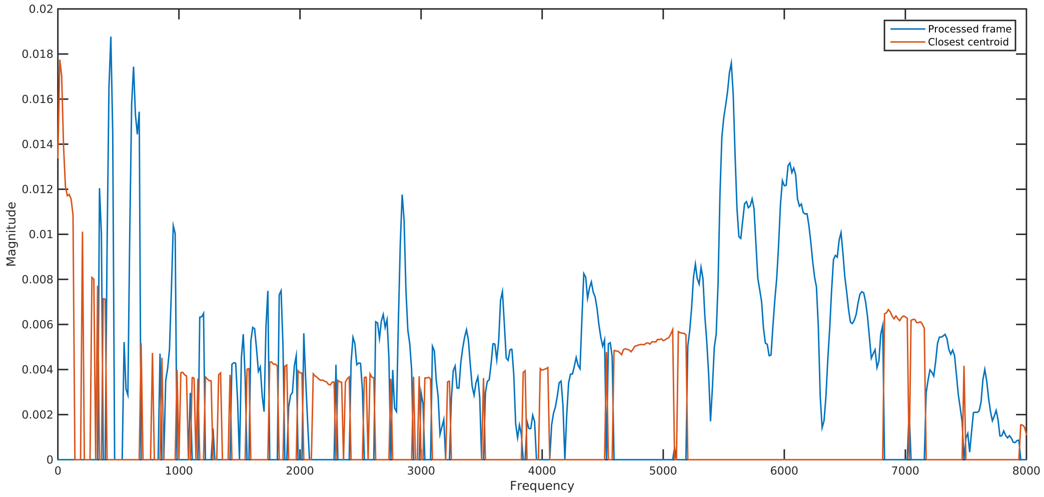


Figure 3.2: A time-frame processed with binary masks in blue and the closest centroid in red.

3.6 Front-End Experimentation

After establishing the better performance of the extended model over the others, it is time to define the most suitable parameters for the front-end techniques we used for the ASR system. To do that, we train a new clean acoustic model with training data from both databases. We tested several front-ends and examined how they performed in terms of WER results. As a next step, new extended acoustic models were trained using that front-ends.

3.6.1 Clean Acoustic Model for Front-End Tuning

The clean acoustic model we used to determine the best front-end included speech signals from both, WSJCAM0 and MC-WSJ-AV, databases. We did that in order to eliminate microphone mismatch. We used the training set from WSJCAM0 database as we did for the clean acoustic model in section 3.3.1. We divided the MC-WSJ-AV database in training and test set based on the table 3.5 and we used the lapel and headset signals as clean signals for the training set. The amount of training data was 17h from WSJCAM0 database and 6h from MC-WSJ-AV database which they both sum up to 23h. The purpose of this acoustic model was to be used as a reference model to determine the best processing technique and the best parameters for that technique. After testing several front-ends including beamformers like MVDR, superdirective, Delay and Sum and postfilters like binary masks and Wiener postfilter, we ended up to the superdirective beamformer followed by either a Wiener postfilter or binary masks. The following figure shows the WER results for front-ends we tested. In the next two sections we will tune both the beamformer and the postfilters for optimum WER results. The tuning will be conducted based only on WER results of the test sets with one speaker, stationary and moving speaker.

WER			
	Stationary Speaker	Moving Speaker	Overlapping Speakers
CTM signals	13.32	14.90	25.30
Superdirective with Binary Masks	27.07	63.55	87.30
Superdirective with Wiener postfilter	27.02	60.25	>100
Superdirective w/o postfilter	32.07	71.90	>100
Delay and Sum with Binary Masks	31.50	68.10	96.60
Delay and Sum with Wiener postfilter	31.94	66.25	>100
MVDR with Binary Masks	34.58	74.35	97.10
MVDR with Wiener postfilter	39.56	80.95	>100
Unprocessed data	51.84	89.25	>100

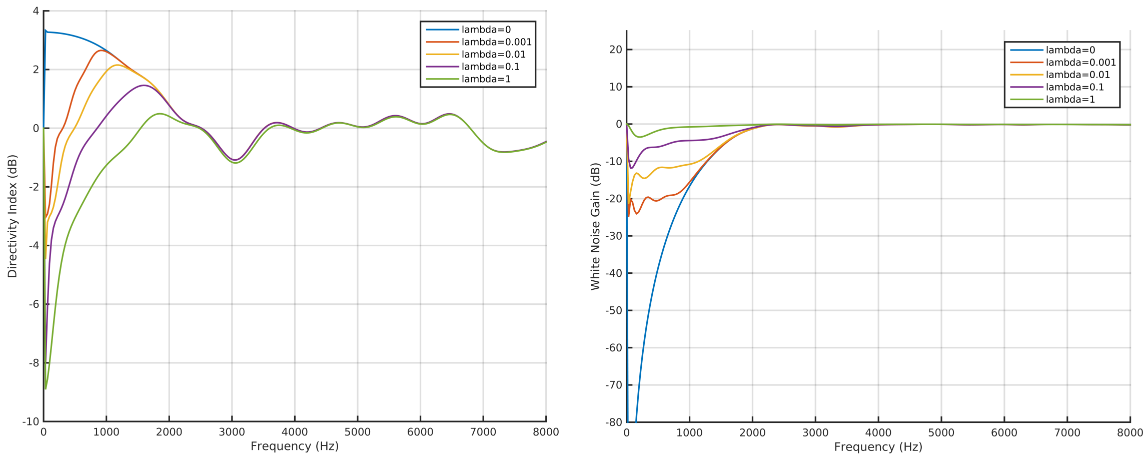
Table 3.6: WER(%) results of multiple front-ends, clean speech signals and noisy unprocessed speech signals using a clean acoustic model

3.6.2 Tuning Front-end

The front-end of our choice is a superdirective beamformer with either a Wiener postfilter or binary masks. In the following section we will discuss the parameters we changed for the beamformer and the postfilters and how they affect the WER results.

- The beamformer of our choice is the **superdirective beamformer**. The parameter we are going to change is lambda since the number of microphones and the array topology are fixed. From the equation 2.23 we can see the use of lambda. In general, a small value of lambda leads to a more directive beamformer but downgrade the WNG performance. On the other hand, a larger value of lambda may lead to better WNG but in excessively large value the component $(\mathbf{\Gamma}_{nn} + \lambda I) \approx \lambda I$ and the superdirective beamformer becomes a delay and sum beamformer. In the following figure you can see the DI and the WNG using different lambda values.

- For the calculation of the **Wiener postfilter**, we need the cross-spectral density of the input signal, $\Phi_{x_m x_n}(\omega, t)$. As we mentioned in section 2.5.2, the calculation of $\Phi_{x_m x_n}(\omega, t)$ needs several consecutive time-frames to achieve better approximation. Using many time-frames would make it difficult to estimate accurately the local spectral density while using a few times-frames would not include



(a) (b)

Figure 3.3: Directivity index(on the left) and WNG(on the right) for multiple lambda values using an 8 microphone circular array with radius 0.1m

sufficient spectral information either. The number of the consecutive frames is chosen based on the WER results as we will see later.

- The second postfilter we used is **Binary Masks**. In this postfilter we made two modifications. The first was on the criterion based on which frequency bin is assigned to a speaker. The condition presented in 2.36 becomes:

$$b_j(\omega,t) = \begin{cases} 1, & \text{if } |\tilde{y}_j(\omega,t)| > a \cdot \max\limits_{\substack{1\leq k\leq S \\ j\neq k}} |\tilde{y}_k(\omega,t)| \\ 0, & \text{otherwise} \end{cases}, \quad j = 1,\ldots,S \tag{3.4}$$

where a is a scalar and denotes that the magnitude of the frequency bin we examine has to be a times bigger the magnitude of the same frequency bin from any other speaker. If a is bigger than 1 then some frequency bins will not be assigned to anyone. In this case they are considered as reverberation or background noise. The second parameter we changed on the binary masks was the spectral floor. Instead of putting zeros in the binary masks we decided to keep a proportion of the magnitude of the initial frequency bin. The value of the proportion was set to 20%. The binary masks then becomes:

$$b_j(\omega,t) = \begin{cases} 1, & \text{if } |\tilde{y}_j(\omega,t)| > a \cdot \max\limits_{\substack{1\leq k\leq S \\ j\neq k}} |\tilde{y}_k(\omega,t)| \\ 0.2, & \text{otherwise} \end{cases}, \quad j = 1,\ldots,S \tag{3.5}$$

Next, we will present the WER results for a wide variety of values for the parameters. The results are grouped into two categories, those with single speaker and those with multiple speakers. The single speaker experiments include both stationary and moving speakers and the presented WER is the combination of the WER from all test sets with one speaker. Also, the framesize is variable. The framesize is in samples in the time domain and each frame is transfered to frequency domain using STFT with double double points. As we can see from the tables below, in single speaker experiments the combination of superdirective beamformer and Wiener postfilter works better for $\lambda = 0.01$, $framesize = 256$ and 4 consecutive frames for the postfilter. For the superdirective beamformer with the binary masks the best combination of parameters is $\lambda = 0.01$, $framesize = 512$ and binary masks with 1.4 criterion with spectral floor.

As mentioned earlier, the acoustic model used for these experiments is a clean acoustic model and the settings for the decoder are $amScale = 0.6, lmScale = 34$ and $lmFactor = 15$.

Stationary and moving speaker experiments

Superdirective beamformer with Wiener postfilter				
History \ Lambda	**0.01** (framesize=256)	**0.05** (framesize=256)	**0.01** (framesize=512)	**0.05** (framesize=512)
5	28.54	29.02	30.63	-
4	27.91	28.26	30.09	30.04
3	28.13	28.66	29.71	30.12

Table 3.7: WER results on single speaker experiments for superdirective beamformer and Wiener postfilter using different values for the tuning using a clean acoustic model.

Superdirective beamformer with Binary Masks				
Criterion(a) \ Lambda	**0.01** (framesize=256)	**0.05** (framesize=256)	**0.01** (framesize=512)	**0.05** (framesize=512)
0.9	-	-	-	31.8
1.0	30.79	30.94	30.97	31.77
1.1	-	-	-	30.95
1.0 /w spectral floor	30.84	-	31.56	32.06
1.1 /w spectral floor	29.81	-	-	31.08
1.2 /w spectral floor	-	-	29.82	30.36
1.3 /w spectral floor	-	-	29.36	30.03
1.4 /w spectral floor	-	-	29.18	-
1.5 /w spectral floor	-	-	29.29	-
1.6 /w spectral floor	-	-	29.44	-

Table 3.8: WER results on single speaker experiments for superdirective beamformer and Wiener postfilter using different values for the tuning using a clean acoustic model.

Overlapping speakers experiments

Superdirective beamformer with Wiener postfilter				
History \ Lambda	**0.01** (framesize=256)	**0.05** (framesize=256)	**0.01** (framesize=512)	**0.05** (framesize=512)
5	96.5	99.0	99.2	100.8
4	96.5	99.2	96.2	99.4
3	97.1	98.2	95.6	97.2

Table 3.9: WER results on overlapping speakers experiments for superdirective beamformer and Wiener postfilter using different values for the tuning using a clean acoustic model.

Superdirective beamformer with Binary Masks				
Criterion(a) \ Lambda	**0.01** (framesize=256)	**0.05** (framesize=256)	**0.01** (framesize=512)	**0.05** (framesize=512)
0.9	-	-	-	87.1
1.0	88.5	91.3	85.7	87.1
1.1	-	-	-	86.3
1.0 /w spectral floor	87.3	-	83.9	86.1
1.1 /w spectral floor	83.7	-	-	83.6
1.2 /w spectral floor	-	-	80.0	81.2
1.3 /w spectral floor	-	-	78.0	78.7
1.4 /w spectral floor	-	-	77.2	-
1.5 /w spectral floor	-	-	77.3	-
1.6 /w spectral floor	-	-	77.9	-

Table 3.10: WER results on overlapping speakers experiments for superdirective beamformer and Wiener postfilter using different values for the tuning using a clean acoustic model.

3.7 Updated Extended Acoustic Models

The final part of the proposed method includes all the extended models trained with both clean and processed training data. Having defined the best parameters for the beamformer and the postfilters, we trained the following extended models.

3.7.1 Extended Acoustic Model (SD+BM)

The first extended acoustic model was trained with both clean and processed speech signals from both databases. The clean signals from WSJCAM0 database include the same training set as the clean acoustic model presented in section 3.3.1 and the clean signals from MC-WSJ-AV database include the lapel and headset signals. The processed data from WSJCAM0 database are the same with the training set as the Multicondition acoustic model presented in section 3.3.2 while the processed data set from MC-WSJCAM0-AV database is the one presented in table 3.5. The processing technique for this model is a superdirective beamformer followed by binary masks using the best parameters we found in the previous section. More specifically, the training data include:

- Clean signals from WSJCAM0 database
- Headset and lapel signals from MC-WSJCAM0-AV database
- Simulated multichannel signals from WSJ with 20dB background noise, processed with superdirective beamformer and binary masks
- Multichannel signals from MC-WSJ-AV database(Array1 and Array2), processed with superdirective beamformer and binary masks

Total Amount of training data: 43.5h

3.7.2 Extended Acoustic Model (SD+Wiener)

The second extended acoustic model was trained again with both clean and processed speech signals from both databases. The clean signals are the same as the previous extended model. The processed data come also from the same data set as the previous extended model. The difference with the previous extended model is the new postfilter which in this case is the Wiener filter described in section 2.5.2. The parameters for the processing technique, i.e. the superdirective beamformer and the Wiener postfilter, are the optimum parameters as presented in the tuning sections 3.6.2. More specifically, the training data include:

- Clean signals from WSJCAM0 database
- Headset and lapel signals from MC-WSJCAM0-AV database
- Simulated multichannel signals from WSJ with 20dB background noise, processed with superdirective beamformer and Wiener postfilter
- Multichannel signals from MC-WSJ-AV database(Array1 and Array2), processed with superdirective beamformer and Wiener postfilter

Total Amount of training data: 43.5h

3.7.3 Extended Acoustic Model (SD+BM/Wiener)

In the last extended model we combined the training data from the previous two extended model. The motivation behind this was to take advantage of both front-ends and eliminate their weakness as we will see later in the results. So, the clean signals come from the CTM signals from WSJCAM0 database and the lapel and headset signals from MC-WSJ-AV database while the multichannel data are enhanced with both superdirective beamformer followed by binary masks and superdirective beamformer followed by a Wiener postfilter. For the processing techniques we used the best parameters as previously. More specifically, the training data include:

- Clean signals from WSJCAM0 database
- Headset and lapel signals from MC-WSJCAM0-AV database
- Simulated multichannel signals from WSJ with 20dB background noise, processed with superdirective beamformer and binary masks
- Multichannel signals from MC-WSJ-AV database(Array1 and Array2), processed with superdirective beamformer and binary masks
- Simulated multichannel signals from WSJ with 20dB background noise, processed with superdirective beamformer and Wiener postfilter
- Multichannel signals from MC-WSJ-AV database(Array1 and Array2), processed with superdirective beamformer and Wiener postfilter

Total Amount of training data: 63.8h

Chapter 4

Evaluation

In this chapter we present the evaluation of the proposed system and discuss about the conducted experiments concerning the proposed front-ends and the trained acoustic models.

The metric we use to define the performance of a front-end or an acoustic model in terms of speech recognition is the WER. This metric indicates the percentage of words that are recognized incorrectly. In the wrongly recognized words are included the words that are inserted, deleted or replaced. Consequently, the WER can take values greater than 100% in case of many insertions.

$$WER = \frac{Insertions + Deletions + Substitutions}{Total\ Number\ of\ Words} \tag{4.1}$$

4.1 REVERB Challenge Results

The first acoustic model we trained for the REVERB Challenge was a clean acoustic model. As we mentioned in chapter 3, the training data came only from the WSJCAM0 database. The following figure shows the WER results of all test sets including stationary speaker, moving speaker and overlapping speakers. We kept the default parameters for the decoder which are $lmFactor = 11$, $lmScale = 34$ and $amScale = 0.6$.

The figure 4.1, shows the WER results of both processed and unprocessed test sets. The unprocessed test sets contain the monophonic speech signal extracted from one of the eight channels of the multichannel signal without processing. We also include the WER results of the test sets which are processed with the superdirective beamformer followed by the binary masks, the superdirective beamformer followed by a Wiener postfilter and the superdirective beamformer without any postfilter. As we can observe, the front-end led to a significant improvement in the WER, especially in the simulated test sets with high

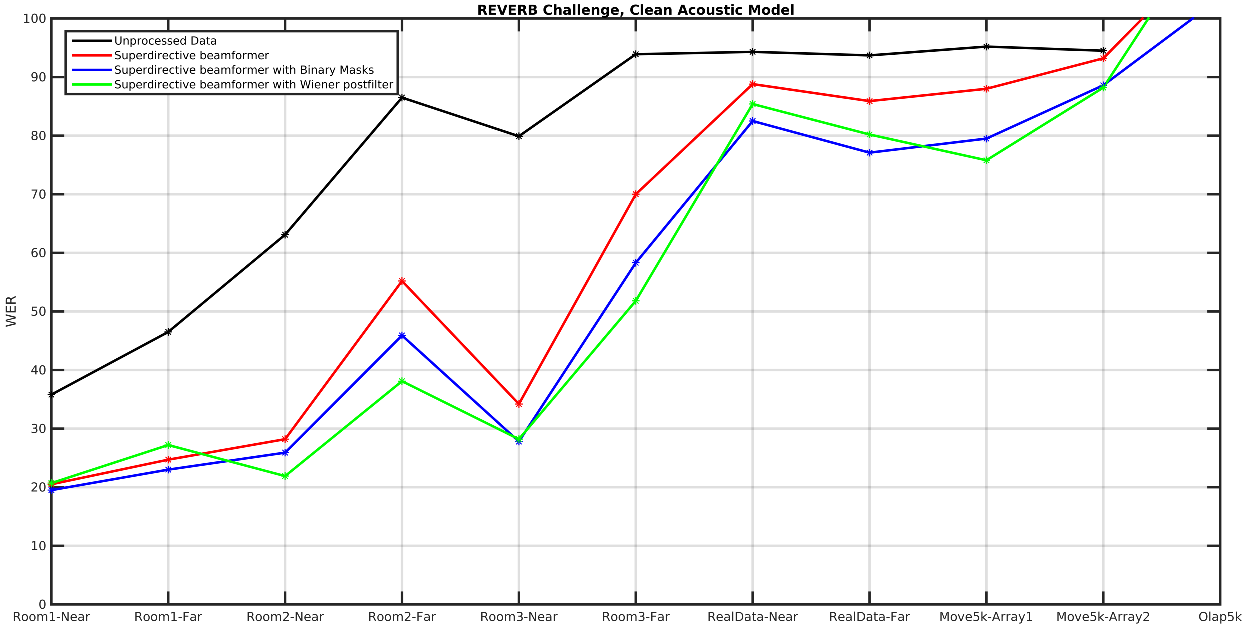


Figure 4.1: WER(%) results of stationary, moving and overlapping speakers using the clean acoustic model as proposed by the REVERB Challenge.

reverberation time such as Room2, with $RT_{60} = 0.6s$ and Room3, with $RT_{60} = 0.7s$. For the real test sets, the performance of the model is less prominent because of microphone mismatch between the databases used. All real test sets came from the MC-WSJ-AV database while the training set contains speech data only from the WSJCAM0 database and this is confirmed by the high WER(>70%). It also becomes clear that the superdirective beamformer without a postfilter cannot enhance the speech signal as robustly as a beamformer followed by either one of the postfilters. For the overlapping speakers experiment we can see a poor performance with the WER exceeding 100% and this is mainly because the training set contains only clean data.

We next trained a multicondition model based on the REVERB Challenge's guidelines. The front-end for this model, as mentioned in Section 3.3.2, is a combination of the Superdirective beamformer followed by Binary Masks with the default value for the criterion of 1.0 and no spectral floor. The figure 4.2 shows the WER results for this model.

As we can see from the figure, the WER results are improved compared to the results of the clean model. This is mostly due to the fact that we applied the same front-end to both training and test sets. Applying a speech enhancement technique to a test set introduces distortion to the signal. When processing the training data with the same technique as the test data, the trained model includes this distortion. As a result, the processed test sets perform better. More specifically, in all stationary and moving speaker

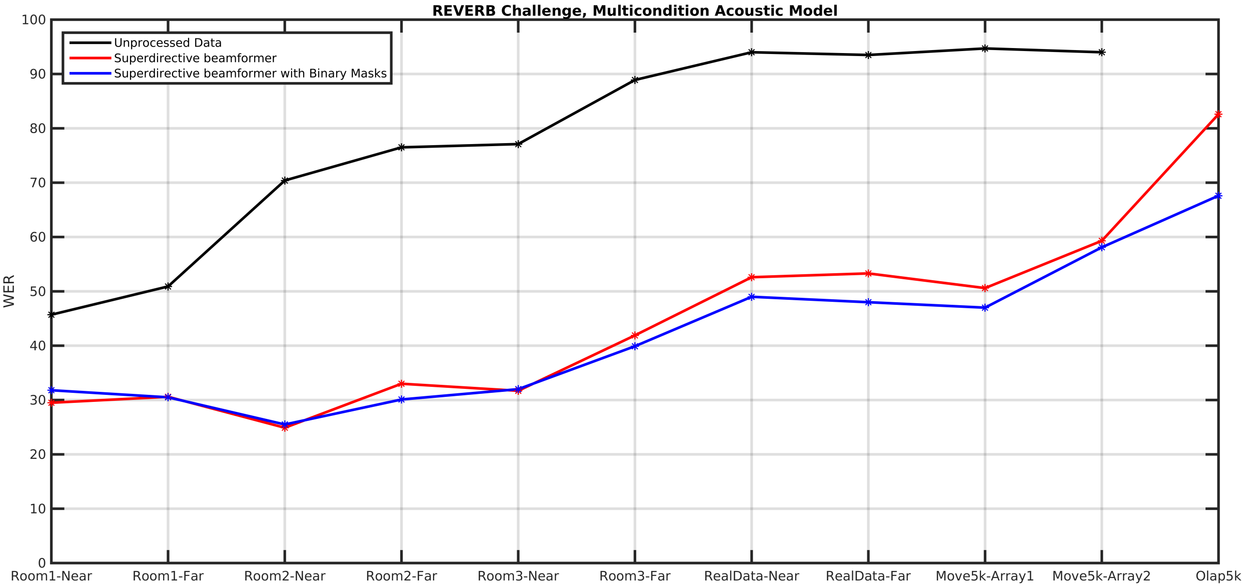


Figure 4.2: WER(%) results of stationary, moving and overlapping speakers using the multicondition acoustic model as proposed by the REVERB Challenge.

experiments the WER is improved apart from the Room1 test set. In Room1, the reverberation of the room is negligible and the processed test set is similar to a clean speech signal, as a result the WER is slightly worse compared to the clean model because clean data are now absent. Lastly, in the overlapping speakers scenario the improvement is noticeable especially when binary masks are deployed. However, due to the microphone mismatch, the WER for the real test sets is significantly high. The parameters for the decoder are $lmFactor = 11$, $lmScale = 34$ and $amScale = 0.6$.

The third model we trained, based on the trends of the REVERB Challenge, was the extended acoustic model. This model included both processed and clean training data. The processing technique was again the superdirective beamformer and binary masks with the default criterion 1.0 and no spectral floor. The figure 4.3 shows the WER results.

As we can see, there is a small improvement in the simulated test sets while the results for the overlapping speakers scenario become slightly worse compared to the results from the multicondition model. More specifically, in Room1 the WER is now close to 20% while in multicondition model it was around 30%. This occurs because the Room1 does not have high reverberation time and the corresponding test set is closer to clean speech and the extended model was trained also with clean data. For the overlapping speakers we can see a small deterioration in WER, this happens because the training data for the extended model includes clean data which are not very close to the highly processed speech signals of overlapping speakers. Lastly, processing the test sets with a beamformer followed by a postfilter has better WER results

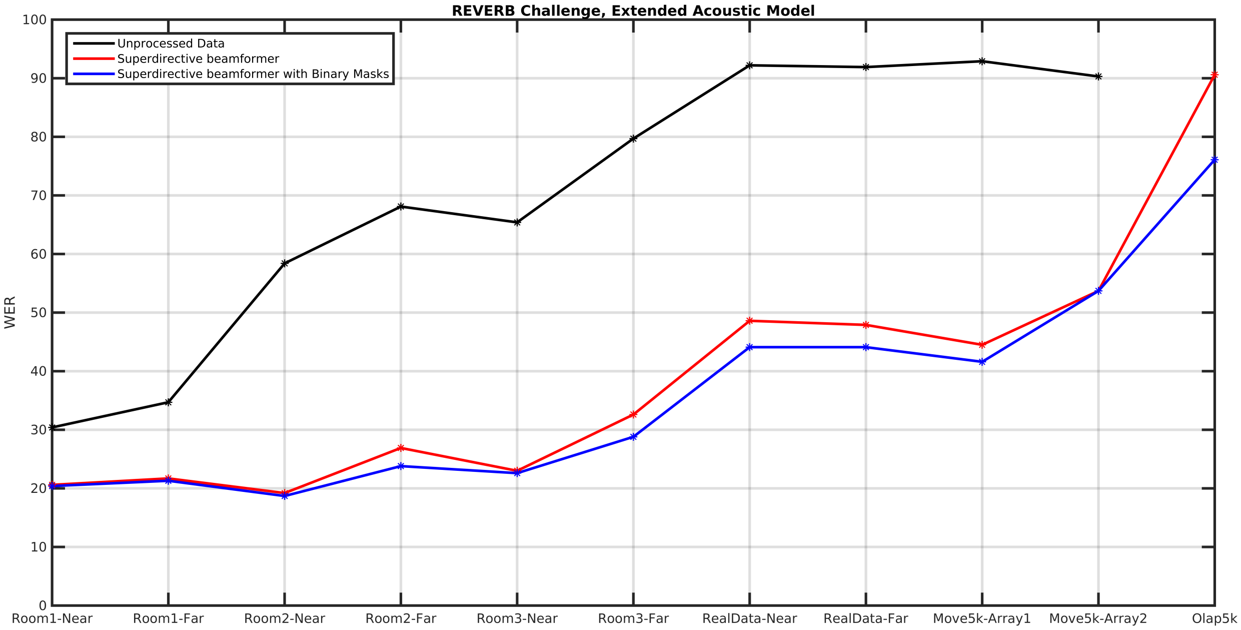


Figure 4.3: WER(%) results of stationary, moving and overlapping speakers using the extended acoustic model as proposed by the REVERB Challenge.

compared to using only a beamformer, especially in overlapping speakers. The parameters for the decoder are $lmFactor = 11$, $lmScale = 34$ and $amScale = 0.6$.

The next figure 4.4 shows the WER results from all the previous acoustic models, i.e. clean, multicondition and extended acoustic model. The presented results are only for the test sets processed with the superdirective beamformer and the binary masks. As we can observe, including diverse data in the training set as well as having a big amount of data, makes the model more robust to the training conditions. Thus, we continued our experimentation in this sense, training three more extended models, which will be evaluated later on.

4.2 Adaptation

As described in Chapter 3.4, we adapted the multicondition model using speech signals from MC-WSJ-AV database in order to see if these data help to cope with the channel mismatch. The following figure shows the WER results from the original multicondition model and the adapted multicondition model. As we can see the new adapted multicondition model indicates a small improvement of about 5% to 10% only on the real test sets as expected. The difference is relatively small because the adaptation data were only a small fraction of the total training data of the multicondition model.

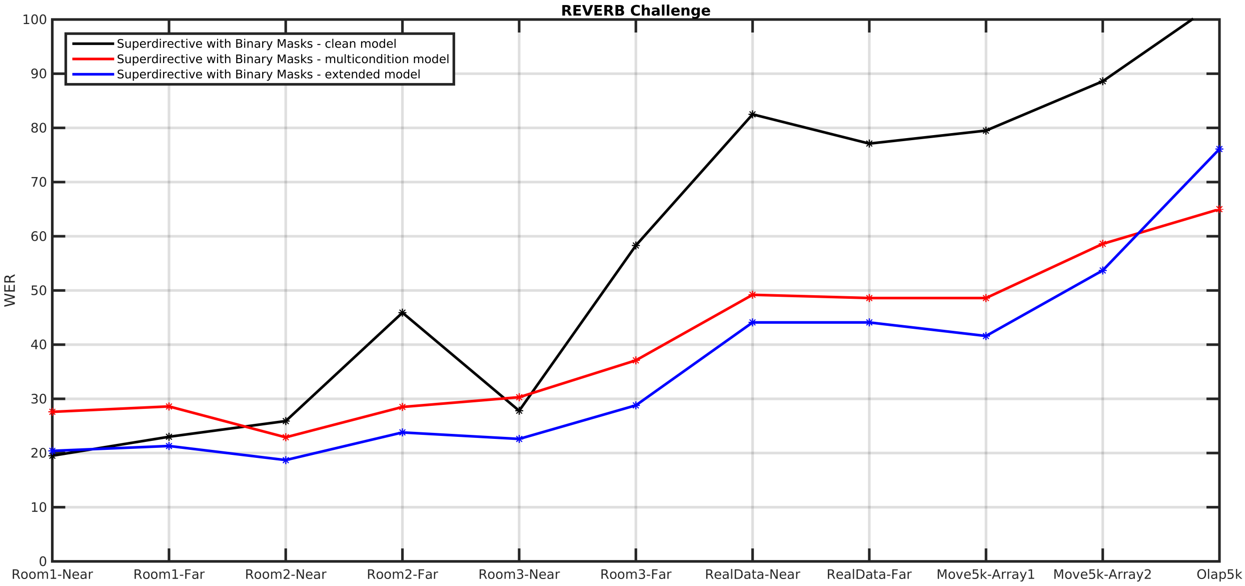


Figure 4.4: WER(%) results of stationary, moving and overlapping speakers using the clean, multicondition and extended acoustic model as proposed by the REVERB Challenge.

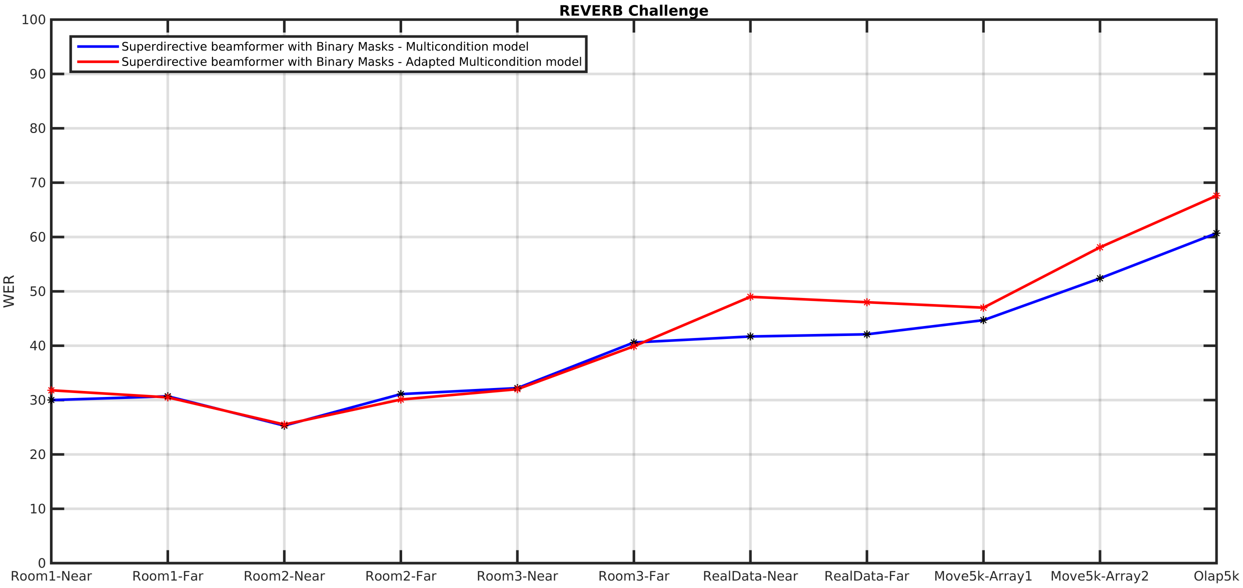


Figure 4.5: WER(%) results of stationary, moving and overlapping speakers using the original multicondition and the adapted multicondition acoustic model.

4.3 Data Imputation

As described in section 3.5, the data imputation technique was applied to the test sets processed with the superdirective beamformer and the binary masks in order to fill in the gaps in the frequency spectrum introduced by the binary masks. The following figure shows the WER results of the test sets processed with the superdirective beamformer followed by the binary masks with and without data imputation using a clean acoustic model. In the same figure, we also provide the results of the test sets processed with the superdirective beamformer followed by binary masks with spectral floor and the superdirective beamformer followed by a Wiener postfilter. As we can observe, the WER results of the data imputation technique are better than the original beamformer with binary masks. However it is not better than the tuned beamformer followed by binary masks and the tuned beamformer with a Wiener postfilter.

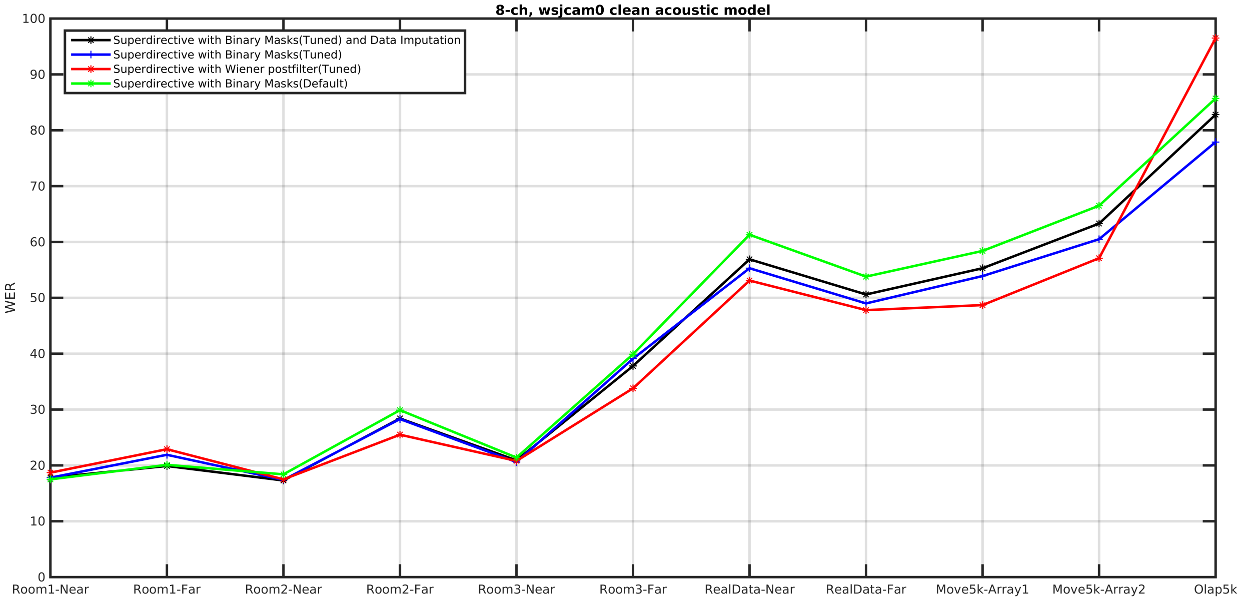


Figure 4.6: WER(%) results of stationary, moving and overlapping speakers processed with Superdirective beamformer(SD) and Binary Masks(BM) with Data Imputation(DI) using a clean acoustic model.

4.4 Extended Models Results

After training the acoustic models according to the REVERB challenge, we then trained our extended acoustic models. The first extended acoustic model was trained with both clean and processed data from both databases, WSJCAM0 and MC-WSJ-AV. The front-end we used was the tuned front-end presented in section 3.7.1. More specifically, the front-end included a superdirective beamformer with lambda=0.01

followed by binary masks with criterion 1.4 and 20% spectral floor. The parameters we used for the decoder are $lmFactor = 15$, $lmScale = 34$ and $amScale = 0.6$. The following figure shows the WER results for all testsets. We also provide the WER results for the corresponding clean test sets in order to define the best results expected from this acoustic model.

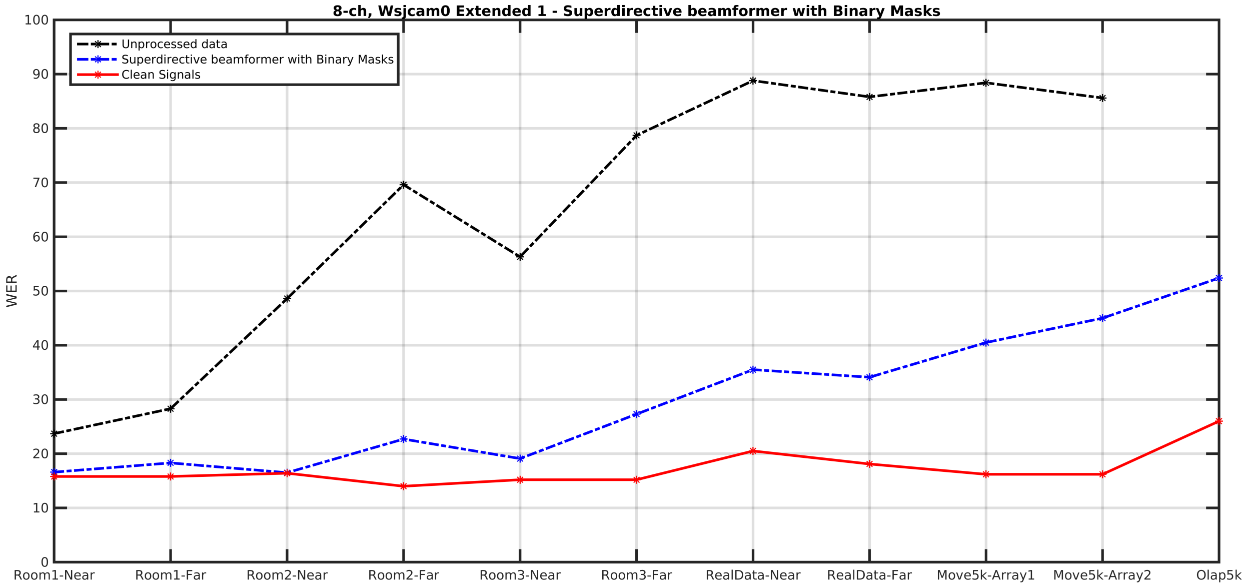


Figure 4.7: WER(%) results of stationary, moving and overlapping speakers using the Extended acoustic model with superdirective beamformer and Binary Masks as front-end.

	Room1 Near	**Room1 Far**	**Room2 Near**	**Room2 Far**	**Room3 Near**	**Room3 Far**	**RealData Near**	**RealData Far**	**Move Array1**	**Move Array2**	**Olap**
Unprocessed	23.7	28.3	48.6	69.6	56.3	78.7	88.8	85.8	88.4	85.6	-
SD with BM	16.6	18.3	16.5	22.7	19.1	27.3	35.5	34.1	40.5	45	52.4
Clean	15.8	15.8	16.4	14	15.2	15.2	20.5	18.1	16.2	16.2	26

Table 4.1: WER results of clean, processed and unprocessed test sets for stationary, moving and overlapping speaker using Extended acoustic model with Superdirective(SD) beamformer and Binary Masks(BM) as front-end.

As we can see from the table and the figure above, the performance of the extended model is now improved compared to the previous extended model from the REVERB Challenge. In the simulated test sets, i.e. Room1, Room2 and Room3 the WER is less than 30% and in some cases less than 20%. Moreover, for the overlapping speakers, the WER went down to 52.4% compare to the 76% of the extended model from the REVERB Challenge.

As a next step, we trained a new extended model with the same training set as the previous extended model with different front-end. The front-end of our choice for this model was a superdirective beamformer with lamdba=0.01 and a Wiener postfilter with a history of 4 frames as presented in section 3.7.2. The parameters we used for the decoder are $lmFactor = 15$, $lmScale = 34$ and $amScale = 0.6$.

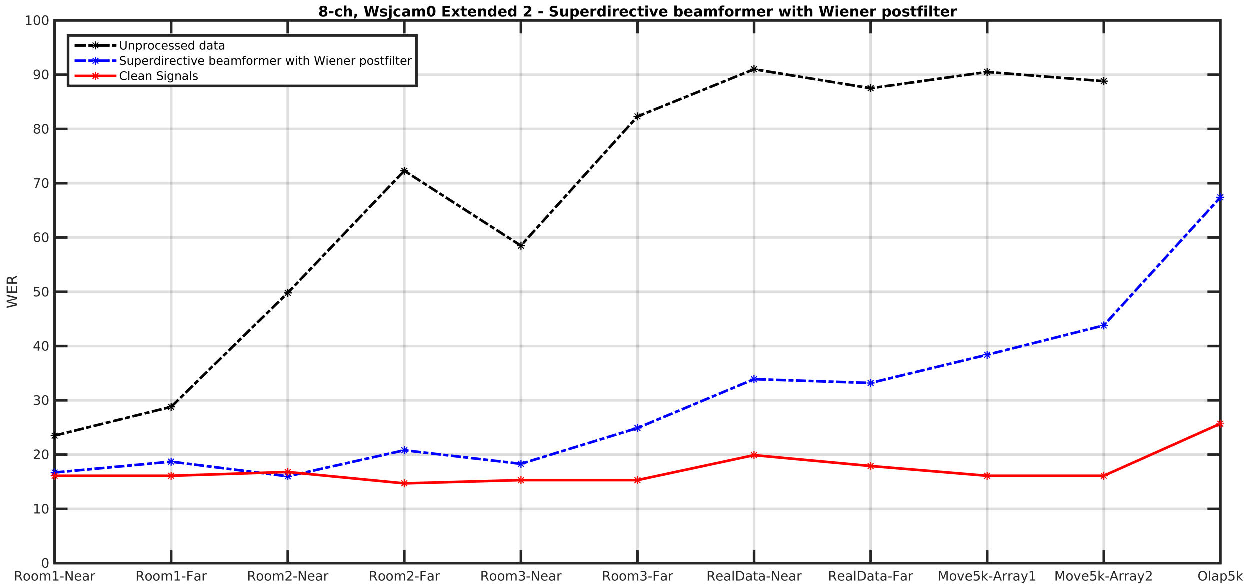


Figure 4.8: WER(%) results of stationary, moving and overlapping speakers using the Extended acoustic model with superdirective beamformer and Wiener postfilter as front-end.

	Room1 Near	**Room1 Far**	**Room2 Near**	**Room2 Far**	**Room3 Near**	**Room3 Far**	**RealData Near**	**RealData Far**	**Move Array1**	**Move Array2**	**Olap**
Unprocessed	23.5	28.8	49.8	72.3	58.5	82.3	91	87.5	90.5	88.8	-
SD with Wiener	16.7	18.7	16	20.8	18.3	24.9	33.9	33.2	38.4	43.8	67.4
Clean	16.1	16.1	16.8	14.7	15.3	15.3	19.9	17.9	16.1	16.1	25.7

Table 4.2: WER results of clean, processed and unprocessed test sets for stationary, moving and overlapping speaker using Extended acoustic model with Superdirective(SD) beamformer and Wiener postfilter as front-end.

From the above figure and table we can see that the WER results for single speaker experiments are slightly better than the previous extended model. For the simulated test sets and especially for the Near occasions, the WER results are very close to the corresponding clean test sets which means that we get the best possible performance using this acoustic model. For the overlapping speakers experiment, the WER is significantly worse compare to the previous acoustic model because Wiener postfilter is not as effective as binary masks to separate two sources.

The last extended acoustic model we trained was with the combination of the training data of the previous two extended acoustic models. We included both clean and processed training data. The front-end used for this model was the tuned superdirective beamformer followed by the binary masks or the Wiener postfilter. The decoder parameters used are $lmFactor = 15$, $lmScale = 34$ and $amScale = 0.6$.

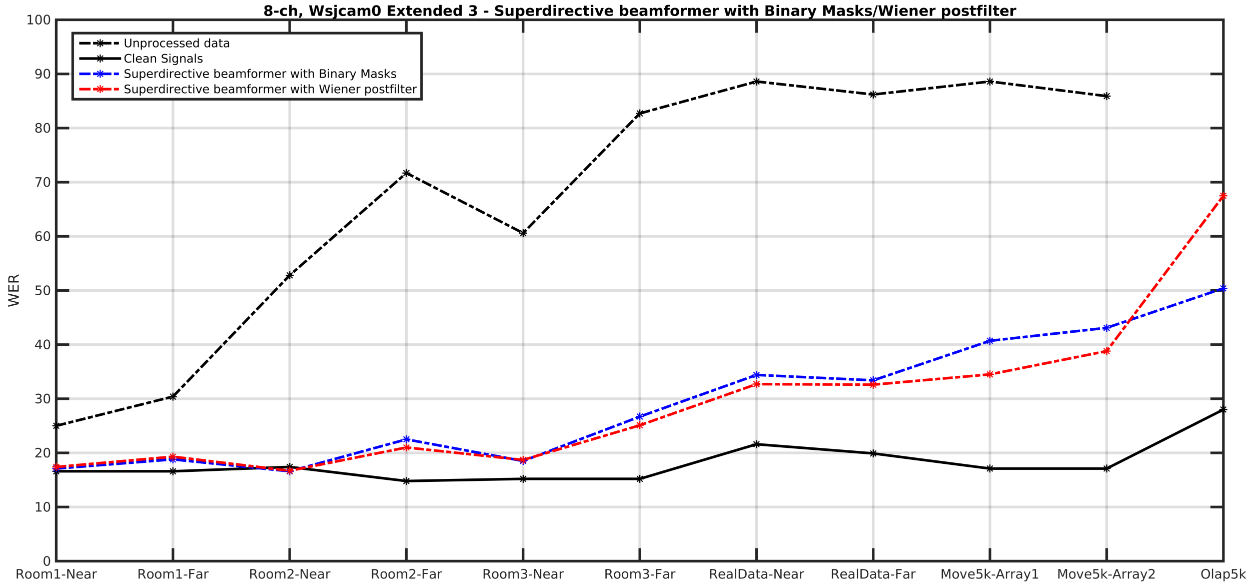


Figure 4.9: WER(%) results of stationary, moving and overlapping speakers using the Extended acoustic model with superdirective beamformer and Binary Masks or Wiener postfilter as front-end.

	Room1 Near	**Room1 Far**	**Room2 Near**	**Room2 Far**	**Room3 Near**	**Room3 Far**	**RealData Near**	**RealData Far**	**Move Array1**	**Move Array2**	**Olap**
Unprocessed	25.0	30.4	52.8	71.7	60.6	82.7	88.6	86.2	88.6	85.9	-
SD with BM	17.1	18.8	16.6	22.5	18.5	26.7	34.4	33.4	40.7	43.1	50.4
SD with Wiener	17.4	19.3	16.7	21.0	18.7	25.1	32.7	32.6	34.5	38.8	67.5
Clean	16.6	16.6	17.4	14.8	15.2	15.2	21.6	19.9	17.1	17.1	28.0

Table 4.3: WER results of clean, processed and unprocessed test sets for stationary, moving and overlapping speaker using Extended acoustic model with Superdirective(SD) beamformer and Binary Masks(BM) or Wiener postfilter as front-end.

The last extended model we trained was the one with the best performance compare to the other two. In single speaker experiments, the WER is less than 35% in the test sets including real data and less than 20% in most of the simulated test sets. That means we are very close to the best recognition performance this model can provide. Additionally, we can easily notice that in single speaker experiment the Wiener postfilter outperforms the binary masks while in overlapping speakers scenario the binary masks perform better. To sum up, the next table shows the WER results from all extended models and as we can see the results from the processed data, especially on single speaker experiments, are very close to the clean test sets.

	Room1 Near	**Room1 Far**	**Room2 Near**	**Room2 Far**	**Room3 Near**	**Room3 Far**	**RealData Near**	**RealData Far**	**Move Array1**	**Move Array2**	**Olap**
Extended 1											
SD with BM	16.6	18.3	16.5	22.7	19.1	27.3	35.5	34.1	40.5	45	52.4
Extended 2											
SD with Wiener	16.7	18.7	16	20.8	18.3	24.9	33.9	33.2	38.4	43.8	67.4
Extended 3											
SD with BM	17.1	18.8	16.6	22.5	18.5	26.7	34.4	33.4	40.7	43.1	50.4
SD with Wiener	17.4	19.3	16.7	21.0	18.7	25.1	32.7	32.6	34.5	38.8	67.5
Clean	16.6	16.6	17.4	14.8	15.2	15.2	21.6	19.9	17.1	17.1	28.0

Table 4.4: WER results of clean, processed and unprocessed test sets for stationary, moving and overlapping speaker using Extended acoustic model with Superdirective(SD) beamformer and Binary Masks(BM) or Wiener postfilter as front-end.

Chapter 5

Conclusions and Future Work

In this thesis we proposed a front-end for robust Automatic Speech Recognition for the scenarios with one stationary speaker, one moving speaker and two overlapping speakers. We performed an extensive study on various front-ends combined with different acoustic models to achieve the best combination.

The first step of the approach was to train three acoustic models based on the guidelines of the REVERB Challenge. The purpose of this step was to compare the performance of the three models depending on the training data. The first model was trained with clean data, the second with processed data and the last model with a mix of clean and processed data. The outcome of this step was to establish the superior performance of the last extended model and use this assumption for our next steps. Indeed, the results validated the assumption.

The next step was to find the best front-end to apply to both training and test data to create a robust acoustic model. We tried multiple front-ends with fixed and adaptive beamformers such as the superdirective, the delay and sum and the MVDR beamformers and postfilters like binary masks and a Wiener postfilter. The superdirective beamformer performed better compared to the other beamformers, and even better when paired with a postfilter. The WER results showed a significant improvement in single speaker experiments when the beamformer was followed by either the binary masks or the Wiener postfilter while in overlapping speaker the binary masks performed better. Following to that, we proceeded to tuning the parameters of the front-end in both the beamformer and the postfilters. For the binary masks we proposed a new method with a different criterion of 1.4 combined with binary masks with a spectral floor which led to an improvement of 2.6% in stationary speaker experiments, 7.3% in moving speaker experiments and 9.9% in overlapping speakers compared to the original binary masks with criterion 1.0 and without spectral floor. In addition, we tried a data imputation method to fill the gaps in the frequency spectrum caused by the binary masks. Although, this method performed better compared to the original binary masks by 4.6% for stationary speaker, 5% for moving speaker and 3.3% for overlapping speakers, it could not outperform the proposed binary masks with the different criterion and the spectral floor or the Wiener postfilter.

The final step of this thesis was to train a new extended acoustic model with the best front-ends found above. The extended model we trained included both clean and processed data. The front-end we used included the superdirective beamformer with binary masks and a Wiener postfilter. The performance of this model in simulated test sets with a single stationary speaker, like Room1, Room2 and Room3, was close to the WER results of the corresponding clean test sets. Also, the relative improvement in WER of the same front-end using this extended model compared to the clean acoustic model was 22% for stationary speaker, 27.6% for moving speaker and 34.7% for overlapping speakers.

Even though our ASR system performs adequately, in all reverberant rooms, we could further enhance its performance by deploying useful techniques such as Neural Networks (NN). Neural Networks can be trained and used alongside with binary masks for better speaker separation or with data imputation for better speech enhancement. Moreover, on the beamformer's side we could replace the steering vector used for the calculation of the beamformer's weights with real measurements. As we know, in real life occasions the steering vector does not accurately represent the sound field. Furthermore, it would be useful to observe the following metrics of Signal to Noise Ratio (SNR), Signal to Interference Ratio (SIR) and Signal to Distortion Ration (SDR) and study how they correlate with the WER results.

Bibliography

[1] P. P. Parada, D. Sharma, J. L. D. Barreda, T. V. Waterschoot, and P. A. Naylor, "A single channel non-intrusive c50 estimator correlated with speech recognition performance," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 4, pp. 719–732, April 2016.

[2] M. Matassoni, M. Ravanelli, S. Jalalvard, A. Brutti, and D. Falavigna, "The fbk system for the chime-4 challenge," *CHIME - Speech Separation and Recognition Challenge*, 2016.

[3] I. McCowan, M. H. Krishna, D. Gatica-Perez, D. Moore, and S. Ba, "Speech acquisition in meetings with an audio-visual sensor array," *IEEE International Conference on Multimedia and Expo*, January 2005.

[4] M. Delcroix, T. Yoshioka, A. Ogawa, Y. Kubo, M. Fujimoto, N. Ito, K. Kinoshita, M. Espi, T. Hori, T. Nakarani, and A. Nakamura, "Linear prediction-based dereverberation with advanced speech enhancement and recognition technologies for the reverb challenge," *REVERB Workshop*, 2014.

[5] F. Weninger, S. Watanabe, J. L. Roux, J. R. Hershey, Y. Tachioka, J. Geiger, B. Schuller, and G. Rigoll, "The merl/melco/tum system for the reverb challenge using deep recurrent neural network feature enhancement," *REVERB Challenge*, 2014.

[6] J. Neymann and L. D. R. Haeb-Umbach, "Wide residual blstm network with discriminative speaker adaptation for robust speech recognition," *REVERB Challenge*, 2014.

[7] M. R. Bai, J.-G. Ih, and J. Benesty, *Acoustic Array Systems: Theory, Implementation and Application*. Singapore: John Wiley and Sons Singapore Pte. Ltd, 2013.

[8] I. Cohen, J. Benesty, and S. Gannot, *Speech Processing in Modern Communication: Challenges and Perspectives*, ser. Springer Topics in Signal Processing. Heidelberg: Springer-Verlan Berlin, 2010, vol. 3.

[9] N. Ito, “Robust microphone array signal processing against diffuse noise,” Ph.D. dissertation, University of Tokyo, April 2012.

[10] J. Bitzer and K. U. Simmer, *Microphone Arrays: Signal Processing Techniques and Applications.* Heidelberg: Springer-Verlag Berlin, 2001, ch. Superdirective Microphone Arrays, pp. 19–38.

[11] M. Wolfel and J. McDonough, *Distant Speech Recognition.* John Wiley and Sons Singapore Pte. Ltd, 2009.

[12] I. A. McCowan and H. Bourland, “Microphone array post-filter based on noise field coherence,” *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 6, pp. 709–716, November 2003.

[13] J. Benesty, J. Chen, and Y. Huang, *Microphone Array Signal Processing*, ser. Springer Topics in Signal Processing. Heidelberg: Springer-Verlan Berlin, 2008, vol. 1.

[14] E. Habets and S. Gannot, “Generating sensor signals in isotropic noise fields,” *Acoustical Society of America*, pp. 3464–3470, 2007.

[15] E. A. Lehmann and A. M. Johansson, “Diffuse reverberation model for efficient image-source simulation of room impulse responses,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 18, no. 6, pp. 1429–1439, August 2010.

[16] S. Lefkimmiatis and P. Maragos, “Optimum post-filter estimation for noise reduction in multichannel speech processing,” *14th European Signal Processing Conference (Eusipco 2016)*, September 2006.

[17] I. McCowan, M. Lincoln, and I. Himawan, “Microphone array shape calibration in diffuse noise fields,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 16, no. 3, pp. 666–670, February 2008.

[18] I. A. McCowan, “Robust speech recognition using microphone arrays,” Ph.D. dissertation, Speech Research Laboratory, School of Electrical and Electronic Systems Engineering, Queensland University of Technology, Australia, 2001.

[19] J. Benesty, J. Chen, and I. Cohen, *Design of Circular Differential Microphone Arrays*, ser. Springer Topics in Signal Processing. Springer Internation Publishing, 2015, vol. 12.

[20] D. Pavlidi, A. Griffin, and A. Mouchtaris, “Real-time multiple sound source localization and counting using a circular microphone array,” *IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 21, no. 10, pp. 2193 – 2206, October 2010.

[21] A. Karbasi and A. Sugiyama, “A new doa estimation method using a circular microphone array,” *15th European Signal Processing Conference(EUSIPCO)*, September 2007.

[22] H. Cox, R. M. Zeskind, and M. M. Owen, “Robust adaptive beamforming,” *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. 35, no. 10, pp. 1365–1376, October 1987.

[23] J. Benesty, J. Chen, Y. Huang, and I. Cohen, *Noise Reduction in Speech Processing*, ser. Springer Topics in Signal Processing. Heidelberg: Springer-Verlan Berlin, 2009, vol. 2.

[24] S. Doclo and M. Moonen, “Superdirective beamforming robust against microphone mismatch,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 2, pp. 617–631, February 2007.

[25] J. Capon, “High-resolution frequency-wavenumber spectrum analysis,” *Proceedings of the IEEE*, vol. 57, no. 8, pp. 1408–1418, August 1969.

[26] M. R. P. Thomas, J. Ahrens, and I. Tashev, “Optical 3d beamforming using measured microphone directivity patterns,” *International Workshop on Acoustic Signal Enhancement, Aachen*, September 2012.

[27] G. W. Elko, *Microphone Arrays: Signal Processing Techniques and Applications*. Heidelberg: Springer-Verlag Berlin, 2001, ch. Future Directions for Microphone Arrays, pp. 382–387.

[28] M. Brandstein and D. Ward, *Microphone Arrays: Signal Processing Techniques and Applications*. Springer-Verlag Berlin, 2001.

[29] A. Alexandridis, A. Griffin, and A. Mouchtaris, “Capturing and reproducing spatial audio based on a circular microphone array,” *Journal of Electrical and Computer Engineering*, p. 16, February 2013.

[30] H. K. Maganti, D. Gatica-Perez, and I. McCowan, “Speech enhancement and recognition in meetings with an audio-visual sensor array,” vol. 15, no. 8. IEEE Transactions on Audio, Speech, and Language Processing, October 2007, pp. 2257 – 2269.

[31] K. U. Simmer, J. Bitzer, and C. Marro, *Microphone Arrays: Signal Processing Techniques and Applications*. Heidelberg: Springer-Verlag Berlin, 2001, ch. Post-filtering Techniques, pp. 39–60.

[32] N. Ito, N. Ono, E. Vincent, and S. Sagayama, “Designing the wiener post-filter for diffuse noise suppression using imaginary parts from inter-channel cross-spectra,” *2010 IEEE International Conference on Acoustics, Speech and Signal Processing (ICCASP)*, pp. 2818–2821, March 2010.

[33] R. Zelinski, “A microphone array with adaptive post-filtering for noise reduction in reverberant rooms,” *ICASSP-88, International Conference on Acoustics, Speech, and Signal Processing*, pp. 2578 – 2581, April 1988.

[34] European Media Laboratory GmbH (EML), Mathematikon, Berliner Str. 45, 69120 Heidelberg. [Online]. Available: www.eml.org/

[35] Cambridge University, Engineering Department, Trumpington Street, Cambridge, UK. [Online]. Available: http://svr-www.eng.cam.ac.uk/comp.speech/Section1/Lexical/beep.html

[36] Scoring Toolkit, NIST, National Institute of Standards and Technology. [Online]. Available: http://www1.icsi.berkeley.edu/Speech/docs/sctk-1.2/sctk.htm

[37] “Linguistic data consortium,” 441 Williams Hall, University of Pennsylvania, Philadelphia, PA 19104-6305, USA. [Online]. Available: https://www.ldc.upenn.edu/

[38] D. B. Paul and J. M. Baker, “The design of the wall street journal-based csr corpus.” *DARPA Speech and Language Workshop*, 1992.

[39] T. Robinson, J. Fransen, D. Pye, J. Foote, and S. Renals, *ICASSP-95, International Conference on Acoustics, Speech and Signal Processing*, vol. 5, May 1995.

[40] J. Fransen, D. Pye, T. Robinson, P. Woodland, and S. Young, “Wsjcam0 corpus and recording description,” 1994.

[41] The Centre for Speech Technology Research, University of Edinburgh, United Kingdom. [Online]. Available: http://www.cstr.ed.ac.uk/

[42] “Reverberation time,” Georgia State University. [Online]. Available: http://hyperphysics.phy-astr.gsu.edu/hbase/Acoustic/revtim.html

[43] M. Lincoln, I. McCowan, J. Vepa, and H. K. Maganti, “The multi-channel wall street journal audio corpus (mc-wsj-av): Specification and initial expreriments,” *IEEE Workshop on Automatic Speech Recognition and Understanding*, November 2005.

[44] M. Lincoln, E. Zwyssig, and I. McCowan, “Multi-channel wsj audio,” Linguistic Data Consortium, Philadelphia, April 2014. [Online]. Available: https://catalog.ldc.upenn.edu/LDC2014S03

[45] K. Kinoshita, M. Delcroix, T. Yoshioka, T. Nakatani, E. Habets, R. Haeb-Umbach, V. Leutnant, A. Sehr, W. Kellermann, R. Maas, S. Gannot, and B. Raj, “A summary of the reverb challenge:

state-of-the-art and remaining challenges in reverberant speech processing research," *EURASIP Journal on Advances in Signal Processing*, 2016.

[46] J. F. Gemmeke, H. V. Hamme, and B. Cranen, *IEEE Journal on Selected Topics in Signal Processing*, vol. 4, pp. 272 – 287, April 2010.