Design of Speech Enhancement System Based on Microphone Array

by

Kairan Liang

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2024

© Kairan Liang 2024

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Beamforming is an important speech enhancement technology that can be employed to separate the speech signal from interfering noise. This technique has a decades-long history of application on microphone arrays, ranging from the basic Delay-Sum beamformer to the recently developed CGMM-MVDR beamformer based on iterative optimization algorithm.

This work evaluates the capability and stability of different beamforming methods and variously shaped microphone arrays in isolating human speech and enhancing speech intelligibility through simulation experiments. Furthermore, the issues and advantages of each method are discussed by examining the beampattern and the spectrum of the processed signal. In particular, this work discovered a deficiency in the Time-Frequency masking estimation of the CGMM-MVDR method, analyzed the causes of the issue, and proposed a post-processing method to process the mask, significantly improving the final result.

In addition, based on the results of the simulation experiments, this work designed and fabricated a prototype that includes a microphone array and data processing devices, presenting a viable solution for implementing the improved CGMM-MVDR beamforming algorithm on a portable device. An Android app was also designed to control the steering direction of the beamformer via gesture interaction, and simple tests were carried out in real-world scenarios.

Acknowledgements

I would like to thank all the little people who made this thesis possible.

Table of Contents

Author's Declaration ii

Abstract iii

Acknowledgements iv

List of Figures viii

List of Tables xi

1 Introduction and Literature Review 1

1.1 History and Motivation for Study 1

1.2 Literature Review 2

1.3 Introduction to Beamforming 3

1.3.1 Signal Model and Problem Formulation 4

1.3.2 Performance Metrics of Beamformer 7

1.3.3 Delay-Sum Beamformer 8

1.3.4 MVDR Beamformer 8

1.3.5 Robust MVDR Beamformers 9

1.4 Common Speech Quality Evaluation Criteria 13

1.4.1 MOS 14

1.4.2 MCD 14

1.4.3 STOI . . . 16
1.5 MQTT Protocol . . . 16
1.6 Organization of Thesis . . . 17

2 Data Collection, Simulation, and Evaluation Techniques 18
2.1 Introduction . . . 18
2.2 Real-World Data Collection . . . 18
2.2.1 Microphone and Sound Source Placement . . . 18
2.2.2 Sound Sources and Recording . . . 19
2.3 Simulation Techniques . . . 19
2.3.1 The McRoomsim software . . . 19
2.3.2 Four microphone linear array . . . 20
2.3.3 Six microphones circular array . . . 21

3 Simulation Experiments and Results 23
3.1 Datasets . . . 23
3.1.1 Speech Dataset . . . 25
3.1.2 Noise Dataset . . . 25
3.1.3 Dataset generation with simulation software . . . 25
3.2 Experimental Setup . . . 26
3.2.1 Delay-Sum Beamformer Implementation . . . 26
3.2.2 MVDR Beamformer Implementation . . . 27
3.2.3 AudioSep-MVDR Beamformer Implementation . . . 28
3.2.4 CGMM-MVDR Beamformer Implementation . . . 28
3.2.5 The configurations of signal sources with microphone array . . . 29
3.3 Results and Discussion . . . 30
3.3.1 Introduction . . . 30
3.3.2 Beampattern and Result of Delay-Sum Beamformer . . . 31

3.3.3 Beampattern and Result of MVDR Beamformer . . . 34
3.3.4 Beampattern and Result of AudioSep-MVDR Beamformer . . . 37
3.3.5 Beampattern and Result of CGMM-MVDR Beamformer . . . 40
3.3.6 Results of circular microphone array . . . 41
3.3.7 Speech Quality results of microphone arrays . . . 43

4 Discussion of results and improvement method 45
4.1 Comparison and Discussion of Simulation Results . . . 45
4.2 A post-processing method of CGMM-MVDR Beamformer . . . 45
4.2.1 Processing for pixel consistency . . . 46
4.2.2 Mask classification using CNN . . . 47
4.3 Beampattern and Results of CGMM-MVDR method with post-processing algorithm . . . 49
4.3.1 Results of linear microphone array . . . 49
4.3.2 Results of circular microphone array . . . 51
4.3.3 Speech Quality results of post-processed CGMM-MVDR beamformer 53

5 System Design 55
5.1 Introduction . . . 55
5.2 Hardware Design . . . 56
5.2.1 Microphone and Board . . . 56
5.2.2 Edge device . . . 56
5.2.3 Software Design . . . 57
5.2.4 Edge device and Server Software Design . . . 58
5.3 Results and Conclusions . . . 61

6 Final Conclusions and Future Work 62
6.1 Conclusions . . . 62
6.2 Limitations and Future Work . . . 63

References 64

List of Figures

1.1 A uniform linear array with M sensors 6
1.2 A uniform circular array with M sensors 6
1.3 Framework of AudioSep model 10
1.4 The flow chart of MFCC procedure 15
1.5 The architecture of MQTT 17

2.1 Four microphone linear array 20
2.2 Four microphone linear array 21
2.3 Six microphone circular array 21
2.4 RIRs of six microphone array 22

3.1 Linear microphone array target sources and noises 23
3.2 Circular microphone array target sources and noises 24
3.3 Overview of how the multi-channel audio data generated 26
3.4 Overview of Delay-Sum beamformer analysis technique 27
3.5 Overview of MVDR beamformer analysis technique 27
3.6 Overview of AudioSep-MVDR beamformer analysis technique 28
3.7 Overview of CGMM-MVDR beamformer analysis technique 29
3.8 Linear microphone circular array configurations 29
3.9 Linear microphone circular array configurations 30
3.10 The spectrogram of clean speech signal 30

3.11 The spectrogram of noise . . . 31
3.12 Spectrogram of received noisy signal . . . 31
3.13 Beampattern of 4 mic linear array Delay-Sum beamformer . . . 32
3.14 Beampattern of 6 mic circular array Delay-Sum beamformer . . . 32
3.15 Spectrograms of signal applied with Delay-Sum beamformer . . . 33
3.16 Spectrograms of signal applied with Delay-Sum beamformer . . . 34
3.17 Beampatterns of MVDR beamformer . . . 35
3.18 Spectrograms of signal applied with MVDR beamformer . . . 36
3.19 Beampatterns of MVDR beamformer . . . 37
3.20 Spectrograms of signal applied with MVDR beamformer . . . 37
3.21 The spectrogram of separated noise . . . 38
3.22 Beampatterns of AudioSep-MVDR beamformer . . . 38
3.23 Spectrograms of signal applied with AudioSep-MVDR beamformer . . . 39
3.24 Beampatterns of AudioSep-MVDR beamformer . . . 39
3.25 Spectrograms of signal applied with AudioSep-MVDR beamformer . . . 40
3.26 Beampatterns of CGMM-MVDR beamformer . . . 41
3.27 Spectrograms of signal applied with CGMM-MVDR beamformer . . . 41
3.28 Beampatterns of CGMM-MVDR beamformer . . . 42
3.29 Spectrograms of signal applied with AudioSep-MVDR beamformer . . . 42
3.30 MCD assessment results . . . 43
3.31 STOI assessment results . . . 44

4.1 The noise masks with different SNR . . . 46
4.2 Architecture of CNN classification model . . . 48
4.3 Classifier training process . . . 49
4.4 Noise masks after post-processing . . . 50
4.5 Beampattern of post-processed CGMM-MVDR beamformer . . . 50

4.6 Spectrograms of signal applied with post-processed CGMM-MVDR beamformer . . . 51
4.7 Beampattern of post-processed CGMM-MVDR beamformer . . . 52
4.8 Spectrograms of signal applied with post-processed CGMM-MVDR beamformer . . . 52
4.9 MCD assessment results . . . 53
4.10 STOI assessment results . . . 54

5.1 Beamformer orientation control schematic diagram . . . 56
5.2 Detailed specification of microphones . . . 57
5.3 The layout of the board . . . 57
5.4 Six-microphone circular array . . . 57
5.5 The edge device . . . 58
5.6 The prototype . . . 58
5.7 The flow chart of data processing procedures . . . 58
5.8 The states of UI buttons . . . 60
5.9 The states and flow of the app . . . 60
5.10 TF masks in real test . . . 61
5.11 Beampatterns of prototype . . . 61

List of Tables

1.1 MOS Score interpretation table 14

2.1 The locations of microphones in linear array 20
2.2 The locations of microphones in circular array 22

3.1 Combinations of target sources and noises in linear array 24
3.2 Combinations of target sources and noises in circular array 24

Chapter 1

Introduction and Literature Review

1.1 History and Motivation for Study

Speech refers to the sounds produced by human vocal organs for use in social activities. As a carrier of information, speech is one of the most effective means of communication between people. In real life, the process of conveying speech information is often inevitably subject to interference from external noise, which can degrade the quality of speech and affect the performance of speech communication and recognition.

Speech enhancement is one of the most important and challenging tasks in speech applications where the goal is to separate clean speech from noise when noisy speech is given as an input. As a fundamental component of speech-related systems, the applications of speech enhancement vary from speech recognition front-end modules to hearing aid systems for the hearing-impaired.

Single-channel speech enhancement algorithms focus on processing the time-frequency characteristics of speech signals captured by a single microphone, and the available speech information is limited. Traditional single-channel speech enhancement methods include spectral subtraction [27], Wiener filtering [3], and signal subspace algorithms [4]. Spectral subtraction involves analyzing the statistical distribution of speech and noise to estimate and subtract noise from the noisy speech, which can effectively suppress stationary noise. However, due to the limitations of its underlying mechanical, it often fails to provide effective noise reduction for non-stationary noise. For single-channel speech enhancement algorithms based on deep learning, there are methods based on time-frequency masking [29] and spectral mapping [31], which can achieve good results even under non-stationary noise conditions.

Multi-channel speech enhancement algorithms process signals from microphone arrays. Common array signal processing techniques include Independent Component Analysis (ICA) [20], spatial spectrum estimation, and beamforming. Beamforming algorithms have a well-defined and accurate mathematical foundation, enabling the formation of a directional receiving system. It can steer the beam towards the desired direction in space, filter out spatial interference and environmental noise from other directions, and significantly improve signal quality. By selectively enhancing signals from a specific direction, beamforming can separate frequency-domain overlapping signals from different spatial directions, which is a limitation of single-channel speech enhancement algorithms. In complex acoustic environments, beamforming algorithms are beneficial for extracting the required speech signals [30, 14].

1.2 Literature Review

Beamforming is an important task in array signal processing, have been extensively studied and utilized in many applications, including monaural, binaural, and distributed microphone arrays. It uses spatial sensors array to collect physical field[6, 1] (sound field and electric field) data that includes target signal and noise, then do a linear weighted combination processing to the signal and output a scalar signal.

The performance of beamformer can normally be evaluated by the following metrics: Beamwidth, Array Gain, Robustness. Narrow beamwidth can improve the spatial resolution; High array gain can improve the detection capability of weak task; High robustness can decrease the influence of mismatch. These metrics of beamformer are not independent, the purpose of beamformer design is to find the best trade-off of these metrics.

Flanagan et al. proposed the Fixed Beamformer (FBF) algorithm [19], which employs a Delay-Sum approach to suppress non-coherent noise in stable environments, ignoring changes in the acoustic environment and providing a fixed response to the signal. Capon introduced the minimum variance distortionless response (MVDR) beamformer in 1969 [15], which aims to maximize the ideal array gain. Cox et al. proposed robust adaptive processing methods in 1987 to enhance the robustness of the array beamformer against array errors [15]. Griffiths et al. proposed the Generalized Side-lobe Canceller (GSC) method, which is an unconstrained adaptive beamforming algorithm [16]. Many subsequent methods have been improvements based on the MVDR beamformer, there have been studies related to robust MVDR beamforming in the literature [28, 13, 18]. These studies aimed at making MVDR robust against steering vector estimation errors and sound reflections rather than improving steering vector estimation accuracy.

Time-Frequency (TF) masking is an analysis technique useful for isolating an audio source of interest in the presence of other interfering sources. By taking into account the differences in the spectral characteristics between the signal and the interferer, the isolation of the target source can be improved significantly. TF masking has been studied at length in binaural and monaural applications and has been shown to improve speech intelligibility by both automated and subjective measurements [26, 11, 17, 23, 33]. [17] showed use of ideal TF masking, also referred to as Ideal Binary Mask (IBM), in a monaural cocktail party environment and suggested IBM results as a performance goal for other algorithms. A mask-based beamforming approach was proposed in [11]. In [12], a study of complex Gaussian mixture model (CGMM) for the mask estimation can deal flexibly with the spatial fluctuation of the steering vector was proposed to do a speech enhancement task. Further more, the authors employed a new form of beamforming method that uses a speech spectral model based on a CGMM to estimate the time-frequency masks and the steering vector without providing technical details.

In the study of speech enhancement performance, a variety of evaluation methods have been proposed. Subjective testing is an effective measure of human intelligibility and has been used in studies such as [26]. [22] presented a comparison of several objective measures for speech intelligibility such as the Short-Time Objective Intelligibility (STOI) and Mel Cepstrum Distortion (MCD).

1.3 Introduction to Beamforming

Beamforming is a filtering technique that can be used to help isolate a target audio source from an environment with interfering noise sources. In our case, an array of microphones is distributed in a far-field acoustical environment to record the audio from various sources in the environment. The beamforming technique relies on knowledge of microphone and source positions within the environment and requires some level of incoherence between observed signals from competing sources. Additionally, a beamformer's performance is a function of the actual distribution of microphones within the space, but our study will only utilize simple planar array geometries; however, the simulation and evaluation techniques used in this work can be applied to any far-field geometry. The following sections describe the signal's mathematical models and is agnostic of the specific array geometry being used.

Microphone array beamforming response is manipulated such that the array is "steered" to the source of interest, improving its isolation and intelligibility. The simple Delay-Sum beamformer has been shown to be effective for this purpose and can be easily applied to the audio response of the microphone array [24]. Adaptive beamforming techniques have

been developed and studied for microphone arrays and have been shown to further improve speech intelligibility of a speech recording [16].

The far-field condition of Microphone array

Based on the distance of the sound source to the microphone array, acoustic field models can be categorized into two types: near-field and far-field models. The difference between the near-field and the far-field: in the far-field case, the array waves can be considered as a simple plane wave. The direction each signal arrived at each array element is at the same, and the difference between the received signal amplitude in each array element is ignored. However, in the near-field case, the array waves should adopt the spherical wave model, and the signal arrives at each array element in different directions and amplitudes. Therefore, near-field array considers the amplitude attenuation and phase shift. Clearly, the far-field model is a simplification of the actual model, greatly reducing the complexity of processing. Most general speech enhancement methods are based on the far-field model.

The distinction between near-field and far-field models is not absolute; it is generally considered that when the distance from the sound source to the center reference point of the microphone array is much greater than the wavelength of the signal, it is classified as a far-field scenario; otherwise, it is considered a near-field scenario. Here is a Empirical formula:

$$r > \frac{2L^2}{\lambda_{min}}, \tag{1.1}$$

where r is the distance between the source and the microphone array, L is the distance of two successive microphones in the array and λ_{min} is the minimum wavelength of signal. If the value of r is larger than the right hand side in Equation 1.1, the model can be considered as far-field.

For the remainder of this paper, it will be assumed under the far-field condition, as the frequency range of human speech typically lies between 300 Hz and 8 kHz, with the minimum wavelength being 0.043 meters. Therefore, the far-field condition for distance r is greater than 0.5 meters when the successive distance L is less than or approximate to 10 cm.

1.3.1 Signal Model and Problem Formulation

We consider a plane wave, in the far field, that is far enough from the array and that propagates in an anechoic environment at speed c , and impinges on a sensor array consisting

of M omnidirectional microphones. The observation signal vector (of length M) is:

$$\begin{array}{rl} \mathbf{y}(f) & = \left[\begin{array}{cccc} Y_1(f) & Y_2(f) & \ldots & Y_M(f) \end{array} \right]^T \\ & = \beta(f)\mathbf{x}(f) + \mathbf{n}(f) \\ & = \beta(f)\mathbf{d}(f, \cos(\theta_d)X(f) + \mathbf{n}(f)) \end{array} , \tag{1.2}$$

where $Y_m(f)$ is the mth sensor signal in frequency domain, $X(f)$ is the target signal in frequency domain, $\mathbf{d}(f, \cos(\theta_d))$ is the steering vector at $\theta = \theta_d$ (direction of the target source), $\beta(f,t) = 0\,\text{or}\,1$ denotes the category index represents if there is target signal in observation, and $\mathbf{n}(f)$ is the additive noise signal vector. Usually, the array processing or beamforming is performed by applying a filter to each sensor signal and summing the filtered signals. In the frequency domain, this is equivalent to adding a complex weight to the output of each microphone and summing across the aperture:

$$\begin{array}{rl} Z(f) & = \sum_{m=1}^{M} H_m^*(f) Y_m(f) \\ & = \mathbf{h}^H(f)\mathbf{y}(f) \\ & = X_{fd}(f) + N_{rn}(f), \end{array} \tag{1.3}$$

where $Z(f)$ is the beamformer frequency-domain output signal, and the time-domain output signal is given as:

$$z(t) = F^{-1}\left\{Z(f)\right\} \tag{1.4}$$

$$h(f) = \left[\begin{array}{cccc} H_1(f) & H_2(f) & \ldots & H_M(f) \end{array} \right]^T \tag{1.5}$$

is the beamforming weight vector, which is suitable for performing spatial filtering at frequency f ,

$$X_{fd}(f) = X(f)\mathbf{h}^H(f)\mathbf{d}(f, \cos\theta_d) \tag{1.6}$$

is the filtered target signal, and

$$V_{fd}(f) = \mathbf{h}^H(f)\mathbf{n}(f) \tag{1.7}$$

is the residual noise. For linear array model, the distance between two successive microphones is equal to δ and the direction of the source signal to the array is parameterized by the azimuth angle θ (see Figure 1.1). In this context, the steering vector (of length M) is given by:

$$\mathbf{d}_L(f, \cos\theta_d) = \left[\begin{array}{cccc} 1 & e^{-j2\pi f\tau_0\cos\theta_d} & \ldots & e^{-j(M-1)2\pi f\tau_0\cos\theta_d} \end{array} \right], \tag{1.8}$$

where $j = \sqrt{-1}$ is the imaginary unit, $f > 0$ is the temporal frequency, and $\tau_0 = \frac{\delta}{c}$ is the delay between two successive sensors at the angle $\theta = 0$. We denote by $\omega = 2\pi f$

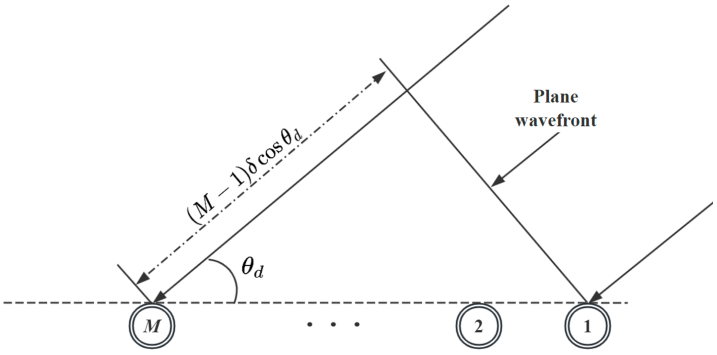


Figure 1.1: A uniform linear array with M sensors

the angular frequency and by $\lambda = \frac{c}{f}$ the wavelength. Since $\cos\theta$ is an even function, so is $\mathbf{d}_L(f, \cos\theta_d)$. In this thesis, θ is limited to angles $\theta \in [0, 2\pi]$.
For a circular array with M sensors, the radius of the circle is r and microphones are uniformly distributed (see Figure 1.2).

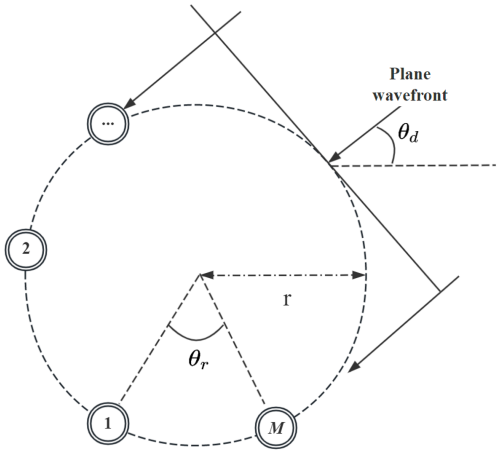


Figure 1.2: A uniform circular array with M sensors

The time delay of the mth microphone to the first one alone the direction of plane wave θ_d is defined as:

$$delay(m, \cos\theta_d) = \frac{r\cos\theta_d(\sin(\frac{2m+1}{2}\theta_r) - \sin(\frac{1}{2}\theta_r)) + r\sin\theta_d(\cos(\frac{2m+1}{2}\theta_r) - \cos(\frac{1}{2}\theta_r))}{c}, \tag{1.9}$$

where θ_r is the angular interval between two successive microphones. The steering vector of a circular array can be written as:

$$\mathbf{d}_C(f, \cos\theta_d) = \left[\begin{array}{cccc} 1 & e^{j2\pi f \times delay(m, \cos\theta_d)} & \ldots & e^{j2\pi f \times delay(M, \cos\theta_d)} \end{array} \right], \tag{1.10}$$

1.3.2 Performance Metrics of Beamformer

Each beamformer has a pattern of directional sensitivity: it has different sensitivities from sounds arriving from different directions. The beampattern or directivity pattern describes the sensitivity of the beamformer to a plane wave (source signal) impinging on the array from the direction θ. Mathematically, it is defined as:

$$\begin{aligned} \beta\left[\mathbf{h}(f), \cos\theta\right] &= \mathbf{d}^H(f, \cos\theta)\mathbf{h}(f) \\ &= \sum_{m=1}^{M} H_m(f) e^{j(m-1)2\pi f \tau_0 \cos\theta} \ , \end{aligned} \tag{1.11}$$

Usually, $\left|\beta\left[\mathbf{h}(f), \cos\theta\right]\right|^2$, which is the power pattern, is illustrated with a polar plot.
For beamformer, input signal-to-noise ratio (SNR) is defined as the power ration of input signal to noise:

$$\mathrm{SNR_{in}(f)} = \frac{\sigma_{\mathrm{s}}^2(\mathrm{f})}{\sigma_{\mathrm{n}}^2(\mathrm{f})}, \tag{1.12}$$

where σ_s^2 , σ_n^2 are power of target signal and noise.
Output SNR of beamformer is defined as:

$$\mathrm{SNR_{in}(f)} = \frac{\sigma_{\mathrm{s}}^2(\mathrm{f})}{\sigma_{\mathrm{n}}^2(\mathrm{f})} = \frac{\mathbf{h(f)}^{\mathrm{H}}\mathrm{R_s(f)}\mathbf{h(f)}}{\mathbf{h(f)}^{\mathrm{H}}\mathrm{R_n(f)}\mathbf{h(f)}} \tag{1.13}$$

where R_s and R_n are covariance matrix of target signal and noise. R_s can be estimated from the convolutive mixing process by the complex-valued multiplication Equation 1.2:

$$R_s(f) = \sigma_s^2(f)\mathbf{d}(f, \cos\theta_d)\mathbf{d}(f, \cos\theta_d)^H \tag{1.14}$$

R_n is the covariance matrix of noise, which is assumed to be independent with R_s.

$$R_n(f) = \sigma_n^2(f)\rho_n(f), \tag{1.15}$$

where ρ_n is the normalized noise covariance matrix.
Array gain is defined as the ratio of SNR_{out} and SNR_{in}:

$$G(f) = \frac{\mathrm{SNR}_{out}(f)}{\mathrm{SNR}_{in}(f)} = \frac{\left|\mathbf{h}(f)^H d(f, \cos\theta_d)\right|^2}{\mathbf{h}(f)^H \rho_n(f)\mathbf{h}(f)} \tag{1.16}$$

If the noise is spatial white noise, then $\rho_n = I$, and replace it in Equation 1.16:

$$G_w = \frac{\left|\mathbf{h}(f)^H d(f, \cos\theta_d)\right|^2}{\mathbf{h}(f)^H \mathbf{I}\mathbf{h}(f)} = \|\mathbf{h}(f)\|^{-2} \tag{1.17}$$

1.3.3 Delay-Sum Beamformer

Delay-Sum beamformer is the most well-known fixed beamformer, which is created if the complex weights $h(f)$ are selected to be phase terms with unity amplitude, and it is derived by maximum the WNG:

$$\begin{array}{cc} \min_{\mathbf{h}(f)} & \mathbf{h}(f)^H \mathbf{h}(f) \\ \text{subject to} & \mathbf{h}(f)^H d(f, \cos\theta_d) = 1 \end{array} \tag{1.18}$$

From the constrain of Equation 1.18, we can get:

$$1 = \left|\mathbf{h}(f)^H d(f, \cos\theta_d)\right|^2 \leq \|\mathbf{h}(f)\|^2 \|d(f, \cos\theta_d)\|^2 = M\|\mathbf{h}(f)\|^2 \tag{1.19}$$

The equal condition of Equation 1.19 is $h(f)$ and $\mathbf{d}(f, \cos(\theta_d))$ are in same direction. Then we get the optimal filter:

$$\begin{array}{rl} \mathbf{h_{DS}}(f, \cos\theta_d) & = \frac{\mathbf{d}(f, \cos\theta_d)}{\mathbf{d}^H(f, \cos\theta_d)\mathbf{d}(f, \cos\theta_d)} \\ & = \frac{\mathbf{d}(f, \cos\theta_d)}{M} \end{array} \tag{1.20}$$

1.3.4 MVDR Beamformer

The array gain of Delay-Sum (DS) beamformer is limited, so the minimum variance distortion-less response (MVDR) beamformer is proposed to improve the array gain. The principle of MVDR beamformer is to make the signal of target position to be distortion-less, then minimize the variance of output noise.

The objective function of MVDR is:

$$\begin{array}{cc} \min_{\mathbf{h}(f)} & \mathbf{h}(f, \cos\theta_d)^H R_n(f) \mathbf{h}(f, \cos\theta_d) \\ \text{subject to} & \mathbf{h}(f, \cos\theta_d)^H \mathbf{d}(f, \cos\theta_d) = 1 \end{array} \tag{1.21}$$

By using the Lagrange multiplier, the Lagrangian is defined as:

$$F(\mathbf{h}, \lambda) \doteq \mathbf{h}^H R_n(f)\mathbf{h} + \lambda(\mathbf{d}(f, \cos\theta_d)^H \mathbf{h} - 1) + \lambda^*(\mathbf{h}^H \mathbf{d}(f, \cos\theta_d) - 1) \tag{1.22}$$

The solution of Equation 1.21 can be get with the optimal condition of Equation 1.22:

$$\mathbf{h_{MVDR}}(f, \cos\theta_d) = \frac{R_n^{-1}(f)\mathbf{d}(f, \cos\theta_d)}{\mathbf{d}(f, \cos\theta_d)^H R_n^{-1}(f)\mathbf{d}(f, \cos\theta_d)} \tag{1.23}$$

1.3.5 Robust MVDR Beamformers

It can be seen from Equation 1.23 that the design of the MVDR beamformer requires the prior knowledge of the noise covariance matrix R_n. It can be estimated in some slot when there is only noise, which means we should do the design of MVDR beamformer when there is no target signal. But in most cases, the received signal contains both target signal and noise, so the covariance matrix of noise cannot be calculated directly. The solution of this issue is to use the observed signal covariance matrix R_x to approximate the noise covariance matrix R_n.

Hence the beamformer weights can be expressed as:

$$\mathbf{h_{MVDR}}(f, \cos\theta_d) = \frac{R_x^{-1}(f)\mathbf{d}(f, \cos\theta_d)}{\mathbf{d}(f, \cos\theta_d)^H R_x^{-1}(f)\mathbf{d}(f, \cos\theta_d)} \tag{1.24}$$

When there is no target signal in the received signal, Equation 1.24 will transform to Equation 1.23. And when it contains the target signal, the performance will be sensitive to the steering vector $\mathbf{d}(f, \cos\theta_d)$. Because the MVDR beamformer will consider the signal received from any direction except target as inference and suppress it, and if the steering vector is different with direction of target signal, then the target signal will also be suppressed. At that time, the performance of the beamformer will be very bad.
To improve the robustness of the MVDR beamformer, some methods are developed to get better estimation of R_n.

Language-queried audio source separation and AudioSep model

Language-queried audio source separation (LASS) is a new paradigm for computational auditory scene analysis (CASA). LASS aims to separate a target sound from an audio mixture given a natural language query, this method primarily employs a frequency-domain ResUNet model as the backbone.
Here, we briefly introduce the separation model and method based on Li's work [21], with the framework illustrated in the following figure:
Here we can see, the spectrogram, which has been processed through the Short-Time

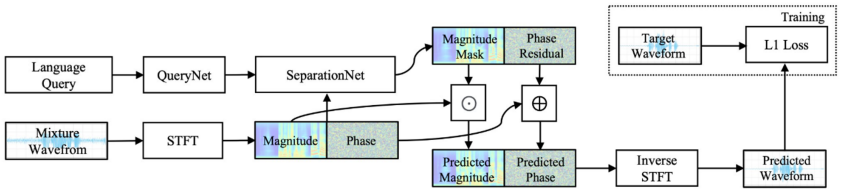


Figure 1.3: Framework of AudioSep model

Fourier Transform (STFT), serves as the input to the SeparationNet part, which utilizes the ResUnet model. Subsequently, the authors construct an encoder-decoder network to process the magnitude spectrogram. The ResUNet model takes as input the complex spectrogram X and outputs the magnitude mask $|M|$ and the phase residual $\angle M$ conditioned on the text embedding equation. The magnitude mask $|M|$ controls the degree to which the magnitude of $|X|$ should be scaled, while the phase $\angle M$ controls the extent to which the phase of $\angle X$ should be rotated. The separated complex spectrogram can be obtained by multiplying the STFT of the mixture and the predicted magnitude mask $|M|$ and phase residual $\angle X$:

$$S = |M| \odot |X| \, e^{j(\angle X + \angle M)}, \tag{1.25}$$

where $\odot$ is the Hadamard product.

CGMM and EM algorithm

Complex Gaussian Mixture Model (CGMM) is a probability model that can be considered as a sum of some complex gaussian distributions:

$$\begin{aligned} CGMM &= \lambda_1 \mathcal{N}_c(\mu_1, \sigma_1) + \ldots + \lambda_K \mathcal{N}_c(\mu_K, \sigma_K) \\ &= \sum \lambda_k \mathcal{N}_c(\mu_k, \sigma_k) \end{aligned}, \tag{1.26}$$

where λ_k is the weight of distribution $\mathcal{N}_c(\mu_k, \sigma_k)$, μ_k is the mean and σ_k is the variance. For a vector $\mathbf{x}$, the probability of it can be expressed as:

$$\begin{aligned} p(\mathbf{x}) &= \lambda_1 \mathcal{N}_c(\mathbf{x}|\mu_1, \sigma_1) + \ldots + \lambda_K \mathcal{N}_c(\mathbf{x}|\mu_K, \sigma_K) \\ &= \sum \lambda_k \mathcal{N}_c(\mathbf{x}|\mu_k, \sigma_k) \end{aligned} \tag{1.27}$$

It needs to note that parameters $\theta_k = (\lambda_k, \mu_k, \sigma_k)$ are unknown, and the number of distributions K is an hyperparameter. So, when we have a set of samples $(\mathbf{x}_1, \ldots, \mathbf{x}_N)$, we can

use them to estimate the parameters.
The log-likelihood can be expressed as:

$$\begin{aligned} \log[L(\mathbf{x}_1, \ldots, \mathbf{x}_N|\theta)] &= \log[\prod_{i=1}^{N} p(\mathbf{x}_i)] \\ &= \sum_{i=1}^{N} \log[\sum_{k=1}^{K} \lambda_k N(\mathbf{x}_i|\theta_k)] \end{aligned} \tag{1.28}$$

EM (Expectation-Maximization) algorithm is an iterative algorithm that can maximum the likelihood when there is a hidden variable $\mathbf{Z}$. We can apply it on CGMM model to do parameters estimation, and the hidden variable $z_i \in \mathbf{Z}$ can be set as the probability of sample x_i belongs to distributions.

The iteration of EM can be expressed as:

Algorithm 1 EM

1: **Initalize randomly** : $\theta^{(0)}$
2: **for** $t \leftarrow 1$ **to** n **do**
3: **E step** : $Q(\theta, \theta^{(t)}) \leftarrow E_{\mathbf{Z}|\mathbf{X},\theta^{(t)}}[\log p(\mathbf{X}, \mathbf{Z}|\theta)]$
4: **M step** : $\theta^{(t+1)} \leftarrow \arg\max_\theta Q(\theta, \theta^{(t)})$
5: **end for**

AudioSep-MVDR

This method uses the LASS model to separate the noise and target signal from receiving. It is intuitive because if the target signal can be separated from receiving then there will be no suppression on target signal. The spectrogram of noise can be calculated as:

$$\mathbf{n}(f) = F\left\{AudioSep(\mathbf{y}(t))\right\}, \tag{1.29}$$

where *AudioSep* is the separation model mentioned in Section 1.3.5 and $y(t)$ denotes received signal in time domain.
Then the estimation of noise covariance matrix is:

$$R_n = E\left\{\mathbf{n}(f)\mathbf{n}^H(f)\right\} \tag{1.30}$$

The weights of beamformer can be expressed as:

$$\mathbf{h}_{\mathbf{AudioSep}}(f) = \frac{R_n^{-1}(f)\mathbf{d}(f, \cos\theta_d)}{\mathbf{d}(f, \cos\theta_d)^H R_n^{-1}(f)\mathbf{d}(f, \cos\theta_d)} \tag{1.31}$$

CGMM-MVDR

CGMM-MVDR is a method to design MVDR beamformer with time-frequency mask which is estimated based on CGMM. Time-frequency mask is a matrix that used to separate the noise and signal. In Equation 1.2, the observed signal is $\beta(f)\mathbf{x}(f)+\mathbf{n}(f)$, and $\beta(f)$ could be 1 or 0, which denotes if the observed signal contains target signal. Here we extent the signal model Equation 1.2 alone time axis and use the sparse mixing model of speech [5].

$$\mathbf{y}(f,t)=\mathbf{d}(f,\cos\theta_d)s_v(f,t)\quad (where\ \beta(f,t)=v)\ , \tag{1.32}$$

where $\beta(f,t)$denotes the category index at the time-frequency point (f,t) , v may take x or n , where the categories represent noisy speech and noise respectively. $s_x(f,t)$ denotes a mixed signal of speech and noise at frequency f and time t , while $s_n(f,t)$ denotes a noise signal at frequency indice f and time frame t .
In CGMM-MVDR, we denote the probability of the time-frequency point (f,t) only contains noise with $\lambda_n(f,t)$. Then we can estimate the covariance matrix of noise as:

$$R_n(f)=\frac{1}{\sum_t\lambda_n(f,t)}\sum_t\lambda_n(f,t)\mathbf{y}(f,t)\mathbf{y}^{\mathbf{H}}(f,t), \tag{1.33}$$

where $\mathbf{y}(f,t)$ denotes the time-frequency spectrum of the received signal, we can use Short-Time Fourier Transform (STFT) to get it.
Similarly, we can estimate the probability of the time-frequency point (f,t) contains both target signal and noise with $\lambda_x(f,t)$. Then the estimation of noisy target signal's covariance matrix is:

$$R_x(f)=\frac{1}{T}\sum_t\mathbf{y}(f,t)\mathbf{y}^{\mathbf{H}}(f,t), \tag{1.34}$$

First, we assume the signal $s_v(f,t)$ follows a complex Gaussian distribution as:

$$s_v(f,t)\sim\mathcal{N}_c(0,\phi_v(f,t)), \tag{1.35}$$

where $\phi_v(f,t)$ corresponds to the variance of the signal at that time-frequency point.

From Equation 1.32 and Equation 1.35, the multichannel observed signal follows a complex Gaussian distribution:

$$\mathbf{y}(f,t)|\beta(f,t)\sim\mathcal{N}_c(0,\phi_v(f,t)R_v(f,t)), \tag{1.36}$$

where $v=n\ or\ x$ based on the condition of $\beta(f,t)$.
The CGMM parameters, i.e., $\phi_v(f,t)$ and $R_v(f,t)$, can be estimated with EM algorithm

0.
The Q function to be maximized in each EM iteration is defined as:

$$Q(\theta) = \sum_{f,t} \sum_{v} \lambda_v(f,t) \log \mathcal{N}_c(\mathbf{y}(f,t); 0, \phi_v(f,t) R_v(f,t)), \tag{1.37}$$

where $\lambda_v(f,t)$ represents the posterior probability of $\beta(f,t)$ being v, and it can be computed as:

$$\lambda_v(f,t) \leftarrow \frac{p(\mathbf{y}(f,t)|\beta = v, \theta)}{\sum_v p(\mathbf{y}(f,t)|\beta = v, \theta)}, \tag{1.38}$$

where $p(\mathbf{y}(f,t)|\beta = v, \theta) = \mathcal{N}_c(\mathbf{y}(f,t); 0, \phi_v(f,t) R_v(f,t))$. The parameter values can be updated as follows:

$$\phi_v(f,t) \leftarrow \frac{1}{M} tr(\mathbf{y}(f,t)\mathbf{y}^H(f,t) R_v(f,t)^{-1}), \tag{1.39}$$

$$R_v(f,t) \leftarrow \frac{1}{\sum_t \lambda_v(f,t)} \sum_t \lambda_v(f,t) \frac{1}{\phi_v(f,t)} \mathbf{y}(f,t)\mathbf{y}^H(f,t) \tag{1.40}$$

The time-frequency mask for point (f,t) can be obtained as the value of $\lambda_n(f,t)$ after convergence. Then, with the noise mask $\lambda_n(f,t)$ and Equation 1.33 we can get the covariance matrix of noise $R_n(f,t)$. The weights of beamformer are:

$$\mathbf{h_{CGMM}}(f, \cos\theta_d) = \frac{R_n^{-1}\mathbf{d}(f, \cos\theta_d)}{\mathbf{d}(f, \cos\theta_d)^H R_n^{-1} \mathbf{d}(f, \cos\theta_d)}, \tag{1.41}$$

where θ_d is the target direction.

1.4 Common Speech Quality Evaluation Criteria

Metrics for evaluating speech quality can be broadly categorized into two types: subjective evaluation and objective evaluation. Subjective evaluation involves conducting auditory tests where listeners assess the quality of speech signals based on their personal perceptions; objective evaluation, on the other hand, utilizes quantitative parameters to measure the quality of speech signals. For subjective assessment, a group of listeners is typically selected to compare the original and processed speech signals, rating them according to a grading scale based on their subjective experience. Objective evaluation compares the original and processed speech signals using objective parameter metrics. While subjective evaluation requires the participation of numerous subjects and is subject to individual variability in evaluation criteria, objective evaluation criteria are generally more feasible and reliable to implement. There are numerous objective speech quality evaluation criteria [32]. Below is a brief explanation of the objective evaluation indicators involved in this paper.

1.4.1 MOS

The most widely used method for subjective quality evaluation is the category rating method, where listeners use a five-point numerical scale to rate the quality of the test signals (see Table 1.1), with 5 indicating "Excellent" quality and 1 indicating "Unsatisfactory" or "Bad" quality. This method is one of the recommended subjective methods by the IEEE subcommittee and the ITU. The measured quality of the test signals is obtained from the average of the scores given by all listeners. This average score is commonly referred to as the Mean Opinion Score (MOS).

Rating	Speech Quality	Degree of Distortion
5	Excellent	Extremely slight, hardly noticeable
4	Good	Slightly noticeable, but not annoying
3	Fair	Noticeable, slightly annoying
2	Poor	Annoying, but tolerable
1	Bad	Very annoying, hard to tolerate

Table 1.1: MOS Score interpretation table

1.4.2 MCD

Mel Cepstrum Distortion (MCD) is an objective metric for measuring the quality of speech, which calculates distortion by comparing the Mel cepstral coefficients of the original speech signal with those of the processed speech signal. Mel cepstral coefficients (MFCC) are extracted from the spectrum of the speech signal and reflect the perceptual characteristics of human hearing. The lower the MCD value, the smaller the difference between the processed speech and the original speech in terms of auditory perception, indicating higher speech quality.

Mel Scale

The Mel scale establishes a mapping from the frequency of sound, known as pitch, to the human auditory perception of frequency. The human ear has a higher resolution for low-frequency sounds compared to high-frequency sounds. By transforming frequency into the Mel scale, features can better match the human auditory perception. The formula for

converting frequency to Mel frequency is as follows:

$$M(f) = 1127\ln(1 + \frac{f}{700}) \tag{1.42}$$

MFCC

In sound processing, the Mel-Frequency Cepstrum (MFC) is a representation of the short-term power spectrum of a sound, based on a linear cosine transform of a log power spectrum on a nonlinear Mel scale of frequency.
Mel-Frequency Cepstral Coefficients (MFCC) are coefficients that collectively make up an MFC. They are derived from a type of cepstral representation of the audio clip (a nonlinear spectrum-of-a-spectrum). The difference between the cepstrum and the Mel-Frequency Cepstrum is that in the MFC, the frequency bands are equally spaced on the Mel scale, which approximates the human auditory system's response more closely than the linearly-spaced frequency bands used in the normal spectrum. This frequency warping can allow for better representation of sound, for example, in audio compression that might potentially reduce the transmission bandwidth and the storage requirements of audio signals. The following chart shows how to derive MFCC:

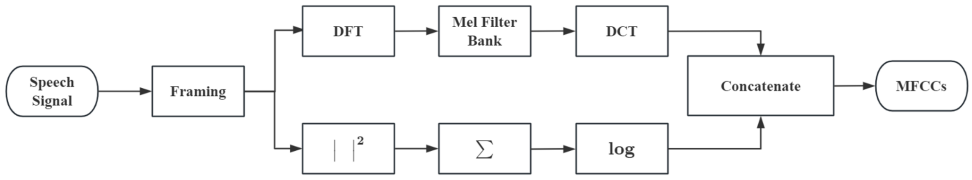


Figure 1.4: The flow chart of MFCC procedure

MCD

MCD can be obtained by calculating the root mean sqrt of Euclidean distance between two sets of MFCCs:

$$\text{MCD} == \sqrt{E(\|\mathbf{a} - \mathbf{b}\|^2)}, \tag{1.43}$$

where a and b are two sets of MFCCs.

1.4.3 STOI

The Short-Time Objective Intelligibility (STOI) measures the correlation of the short-time envelopes between the target speech and the enhanced speech [10] to evaluate the fidelity of the target speech. First, the speech is decomposed using the Discrete Fourier Transform (DFT), and the short-time speech envelopes are normalized and segmented. Then, the correlation coefficients are used to compare the short-time envelopes of the clean speech and the processed speech segments. Finally, the intermediate values of these short-time intelligibility measures are averaged to form a scalar value ranging from [0, 1]. This value has a monotonically increasing relationship with speech intelligibility, meaning that the higher the value, the higher the intelligibility.

1.5 MQTT Protocol

The MQTT (Message Queuing Telemetry Transport) protocol [25] is a lightweight messaging protocol designed for limited bandwidth, high latency, and unreliable networks. It is widely used for IoT (Internet of Things) applications where small code footprint and low power consumption are crucial. Here is an overview of the MQTT protocol:

- Broker: MQTT is based on a publish/subscribe messaging pattern where messages are published to topics and delivered to subscribers of those topics. The broker is responsible for routing messages between publishers and subscribers.
- Client: Any device or application that communicates with the broker is considered an MQTT client.
- Topics: Topics in MQTT are used to categorize messages, allowing for efficient distribution and filtering.

As previously stated, the MQTT protocol is based on a Client/Broker architecture, where data is transmitted from various Clients to the Broker, which then forwards the data based on the Topics subscribed by the Clients. The diagram above illustrates an example involving two subscribers and one publisher, with two topics: a higher-level topic named Weather and a subtopic named temperature under weather. Subscribing to the higher-level topic implies that all subtopics are also subscribed to.

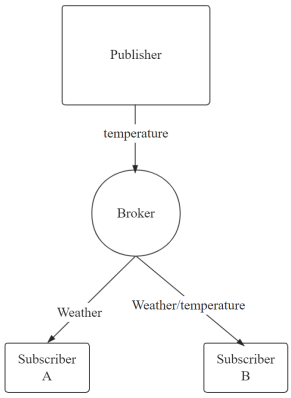


Figure 1.5: The architecture of MQTT

1.6 Organization of Thesis

The organization of the remainder of this thesis is as follows. Chapter 2 introduces the how to conduct simulation and produce datasets in this study, along with the details of setting up the microphone array models in simulation. Chapter 3 provides detailed procedures for implementing the beamforming methods and show the outcomes of these methods. Chapter 4 discusses the performance of these beamforming methods and propose a post-processing algorithm to improve the CGMM-MVDR method. Chapter 5 presents the specific methods for implementing our prototype, which includes two hardware components as well as three software components interconnected via the MQTT protocol. Finally, Chapter 6 summarizes the research and results of this study and provides suggestions for future work on this topic.

Chapter 2

Data Collection, Simulation, and Evaluation Techniques

2.1 Introduction

The purpose of this study is to design a microphone array system that can pick the speech voice in target position and isolate the noise. Test data from a microphone array in a noisy scene is created for analysis by means of both real microphone recordings and simulations. The real microphone recordings are used to demonstrate the practical performance of these analysis techniques, while the simulations are created to provide a wide variety of testing parameters and allow for a more ideal setup in which to test these analyses.

2.2 Real-World Data Collection

2.2.1 Microphone and Sound Source Placement

For the real data collected in this study, a microphone array was set up in a typical office-type environment. This environment has certain background noise which due to the air condition, and it is almost distributed in low frequency. The sound we collected for simulation is mono, which means it is collected by single microphone. The volume of the source was subjectively set to approximately equal that of a human speaking at conversational volume.

2.2.2 Sound Sources and Recording

A collection of 200 different audio recordings is used as the sound sources in this study. Each recording contains a human voice speaking approximately 15 seconds of sentences. Both men and women voices are included in the set of recordings, and each recording contains a unique set of spoken words.

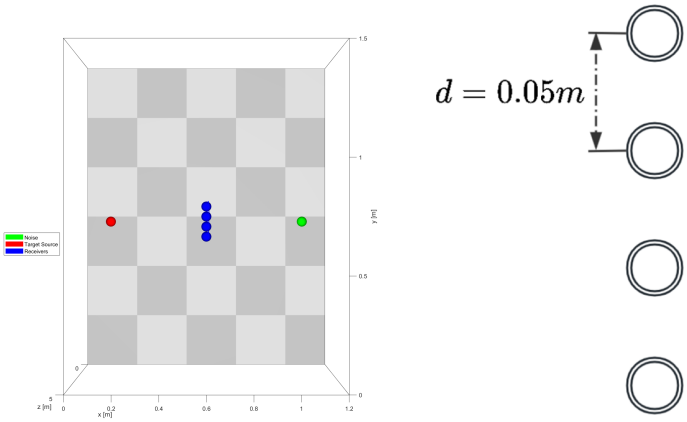
2.3 Simulation Techniques

To understand the performance of the beamforming methods over a wide range of scenarios, virtual simulations are performed. A virtual environment can be defined to emulate that of a real-life scenario (source and microphone positioning, room reverberation effects, etc.). A toolkit McRoomsim [2] is used to create such virtual environments. Placing sound sources at defined locations in the virtual environment allows for a simulation in a room. Additionally, microphone arrays of varying geometries can be defined and simulated. Because there is no position measurement errors (as there could be in real world), simulations can provide an understanding of analysis performance in "ideal" scenarios, while also providing freedom from the requirements of a stationary and quiet lab space for recordings. Finally, simulations allow for great flexibility in quickly changing an environment, reducing time required to test various analysis techniques. For this study, the simulated audio environment (microphone positions, source positions, and propagation speed of sound) were chosen to equal or closely match those parameters from the real. This allows for the most meaningful comparison between results created from simulated and real microphone recordings. In the simulation environment, the dimensions of the room is set as (1.2m $\times$ 1.5m $\times$ 5m) , absorption coefficient is 1, absorption coefficients is 0, so the room is set with no additional noise and reverberation.

2.3.1 The McRoomsim software

The McRoomsim software is a new and efficient room acoustics simulation software package for MATLAB, which can simulate recordings of arbitrary microphone arrays within an echoic room. This simulator supports research related to developing and experimenting with multichannel microphone arrays and higher order ambisonics playback. This room acoustics simulator is implemented as a C program that interfaces with MATLAB and is freely available from the authors. In the following sections, we will utilize this package to model the microphone array and sound sources.

2.3.2 Four microphone linear array



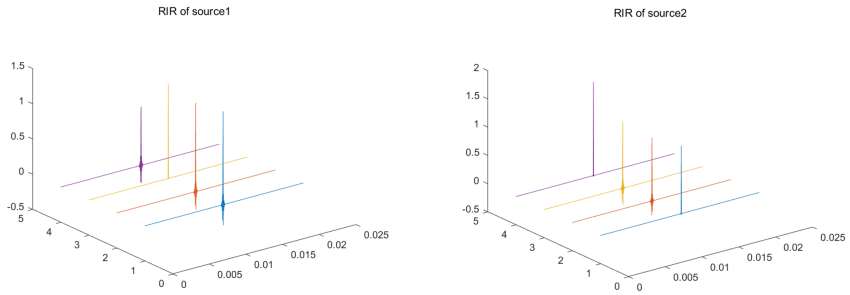
(a) (b)

Figure 2.1: Four microphone linear array

Locations of microphones
$[0.6, 0.8, 0.5]$
$[0.6, 0.75, 0.5]$
$[0.6, 0.70, 0.5]$
$[0.6, 0.65, 0.5]$

Table 2.1: The locations of microphones in linear array

With the McRoomsim software, we can calculate the room impulse response (RIR) of receivers from each source, which is shown in Figure 2.2:

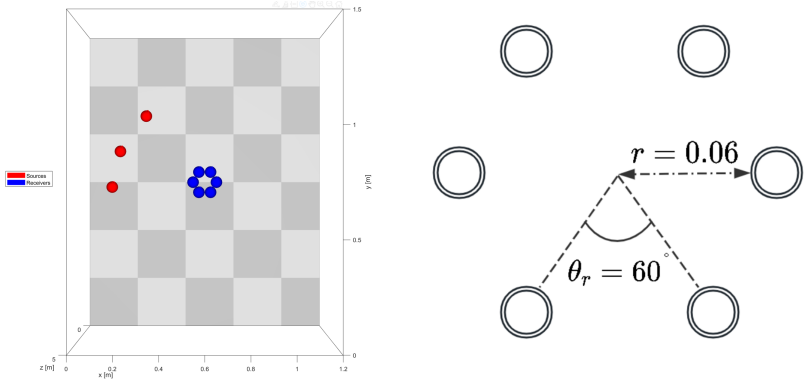


(a) (b)

Figure 2.2: Four microphone linear array

2.3.3 Six microphones circular array

The Figure 2.3 shows the geometry of the six microphones and several sound sources.



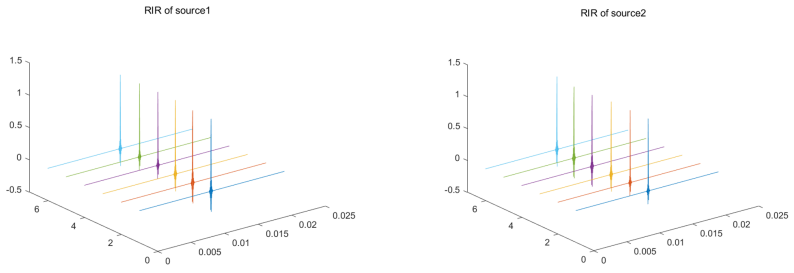
(a) (b)

Figure 2.3: Six microphone circular array

With same method, the RIRs are shown in Figure 2.4:

Locations of microphones
$[0.6300, 0.7770, 0.5000]$
$[0.5700, 0.7770, 0.5000]$
$[0.5400, 0.7250, 0.5000]$
$[0.5700, 0.6730, 0.5000]$
$[0.6300, 0.6730, 0.5000]$
$[0.6600, 0.7250, 0.5000]$

Table 2.2: The locations of microphones in circular array



(a) (b)

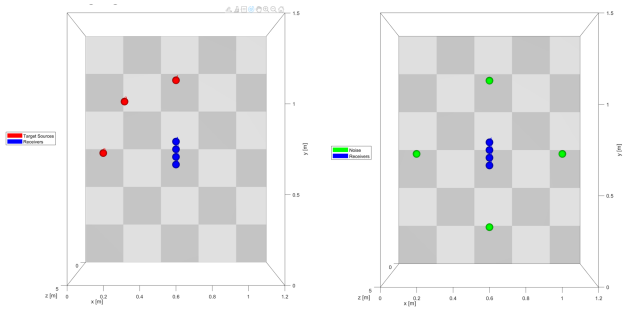
Figure 2.4: RIRs of six microphone array

Chapter 3

Simulation Experiments and Results

3.1 Datasets

To conduct more comprehensive tests of the methods and enhance the model's generalization capabilities, we choose 3 target signal positions and 4 noise positions for each design of microphone array. For each location of the target, we made combinations of different noise locations with it and because of the symmetry of microphone arrays, these combinations can represent most cases.
The following figures and table show the combinations of target and noise with microphone arrays.

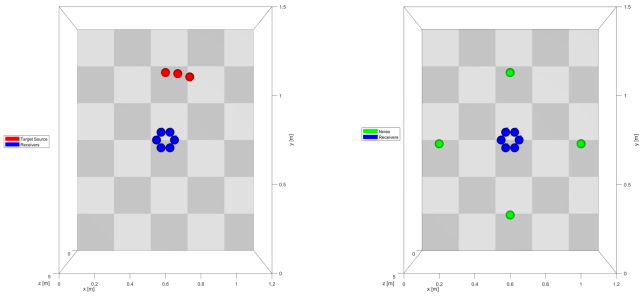


(a) (b)

Figure 3.1: Linear microphone array target sources and noises

Target source locations	Noise locations
$[0.125, 0.725, 0.5]$	$[0.6, 1.2, 0.5]$ $[0.6, 0.25, 0.5]$ $[1.075, 0.725, 0.5]$
$[0.6, 1.2, 0.5]$	$[0.125, 0.725, 0.5]$ $[0.6, 0.25, 0.5]]$ $[1.075, 0.725, 0.5]$
$[0.264, 1.061, 0.5]$	$[0.6, 1.2, 0.5]$ $[0.125, 0.725, 0.5]$ $[0.6, 0.25, 0.5]$ $[1.075, 0.725, 0.5]$

Table 3.1: Combinations of target sources and noises in linear array



(a) (b)

Figure 3.2: Circular microphone array target sources and noises

Target source locations	Noise locations
$[0.762, 1.171, 0.5]$	$[0.6, 1.2, 0.5]$ $[0.125, 0.725, 0.5]$ $[0.6, 0.25, 0.5]$ $[1.075, 0.725, 0.5]$
$[0.682, 1.1927, 0.5]$	$[0.6, 1.2, 0.5]$ $[0.125, 0.725, 0.5]$ $[0.6, 0.25, 0.5]$ $[1.075, 0.725, 0.5]$
$[0.6, 1.2, 0.5]$	$[0.125, 0.725, 0.5]$ $[0.6, 0.25, 0.5]$ $[1.075, 0.725, 0.5]$

Table 3.2: Combinations of target sources and noises in circular array

3.1.1 Speech Dataset

The target signal is evaluated and verified using data from the TIMIT corpus [7], which consists of recordings from 630 speakers from eight major dialect regions in the United States. We selected 200 sentences from it for assessment and validation, with 180 sentences of clean speech used to generate the training set, and the remaining 20 sentences for generating the test set.

3.1.2 Noise Dataset

The noise is a 10-second-long speech segment from the WSJ0 dataset [8]. Five types of noise are from the NOISE-92 database, all of which are non-stationary noises. These noises are Babble, F16, Destroy Engine, Volvo, and Factory, each with a duration of approximately 4 minutes. Random segments from each type of noise are extracted and mixed with the selected clean speech to generate the training set data.

3.1.3 Dataset generation with simulation software

We already defined the real-world mono recording standard in Section 2.2, with impulse responses generated by the simulation software in Section 2.3.1, the multi-channel audio can be created.

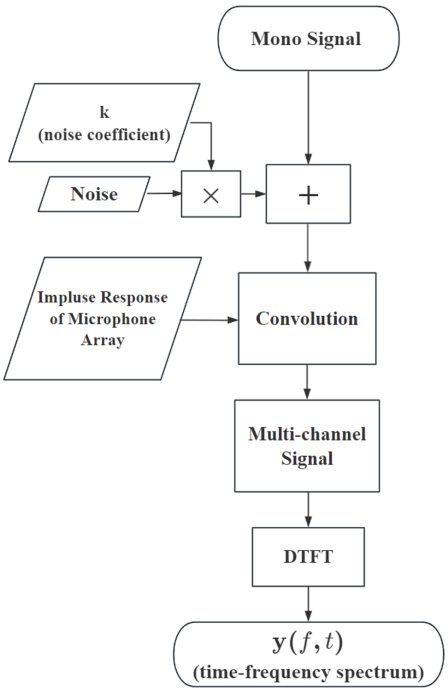


Figure 3.3: Overview of how the multi-channel audio data generated

Here the additional noise coefficient k is used to control the SNR of input signal, in this study, we will gradually increase the power of noise to test the methods.

3.2 Experimental Setup

3.2.1 Delay-Sum Beamformer Implementation

The delay-sum beamformer is not dependent on the nature of the signal itself, as it only applies a constant time-delay (equal to propagation time from source to microphone) to each sample in the signal. Because of this, the application and analysis of the delay-sum beamformer effects are relatively simple.

By Equation 1.8 and Equation 1.10 we can get the steering vectors of linear and circle microphone arrays with dimensions in Figure 2.1 ($M = 4, \delta = 5\,\text{cm}$) and Figure 2.3 ($M = 6, \text{r} = 6\,\text{cm}$) . Then, with Equation 1.20, the weights of delay-sum beamformers can be calculated.

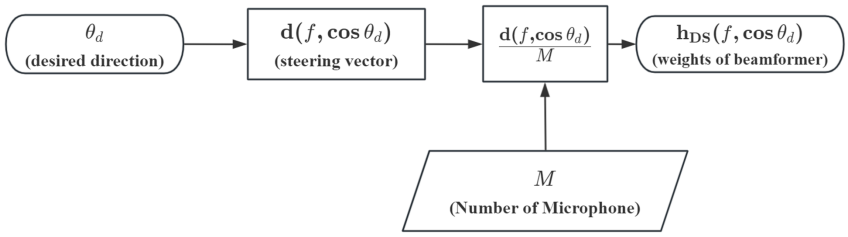


Figure 3.4: Overview of Delay-Sum beamformer analysis technique

As depicted in Figure 3.4, the weights of the fixed beamformer are independent of the received signals and are determined solely by the steering vector and the number of microphones, denoted as M.

3.2.2 MVDR Beamformer Implementation

MVDR beamformer is an adapting method depend on the received signal, as shown in Equation 1.23 and Equation 1.24, we can use the received signal to approximate the covariance matrix of noise signal. And here we will use the mean value to approximate the covariance matrix.
When we received an audio signal, we can use Discrete-Time Fourier Transform (DTFT) to get the time-frequency spectrum $\mathbf{y}(f, t)$, and then do the estimation along time axis with Equation 1.35. With such R_x and desired direction θ_d, the weights of beamformer could be calculated with Equation 1.24.

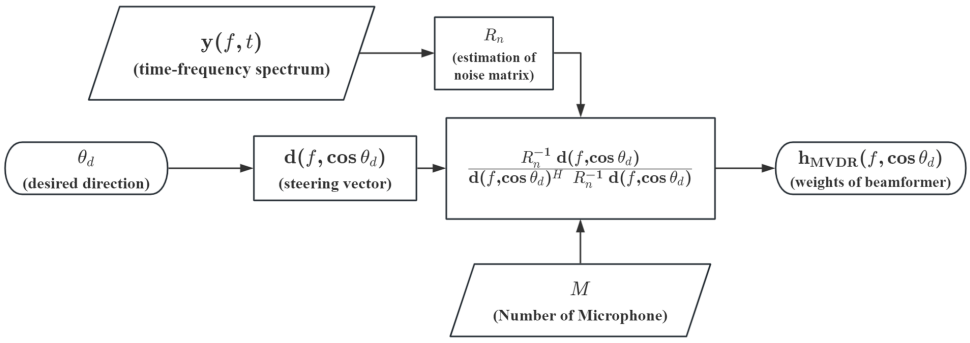


Figure 3.5: Overview of MVDR beamformer analysis technique

3.2.3 AudioSep-MVDR Beamformer Implementation

We will utilize the AudioSep model from Section 1.3.5, with the instruction text designated as "noise". According to Equation 1.30, the processed input signal will be used to estimate the noise spectrogram covariance matrix R_n. Subsequently, the beamformer is obtained through the MVDR algorithm.

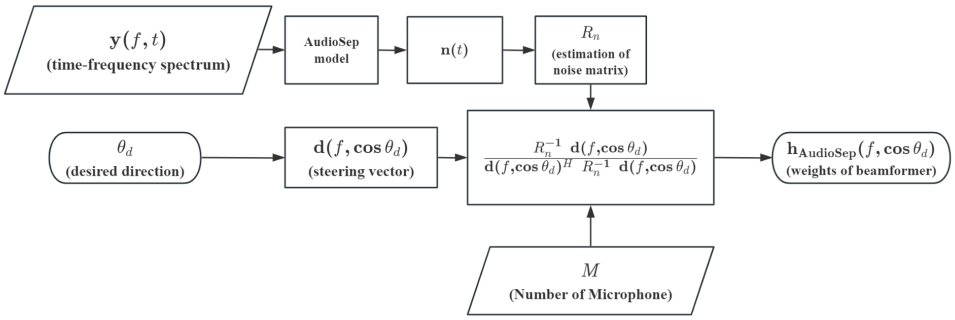


Figure 3.6: Overview of AudioSep-MVDR beamformer analysis technique

3.2.4 CGMM-MVDR Beamformer Implementation

CGMM-MVDR is also an adapting method, and it is independent with the geometry of microphone array. It can separate different sounds based on their statistical features, and it depends on the initiation values in the CGMM iterative algorithm. In this work, we will use the mean value of the covariance matrix as initiation of noisy target signal and identity matrix as initiation of noise signal to start the iteration.
As we talked about earlier, with the time-frequency spectrum $\mathbf{y}(f,t)$, we can use the received signal covariance matrix and identity matrix I as the initiation value of R_x and R_n.
The converged solution of the EM algorithm are two time-frequency masks that separate the target signal with other interference. With the mask of noise $\lambda_n(f,t)$, we can calculate the weights of beamformer with Equation 1.33.

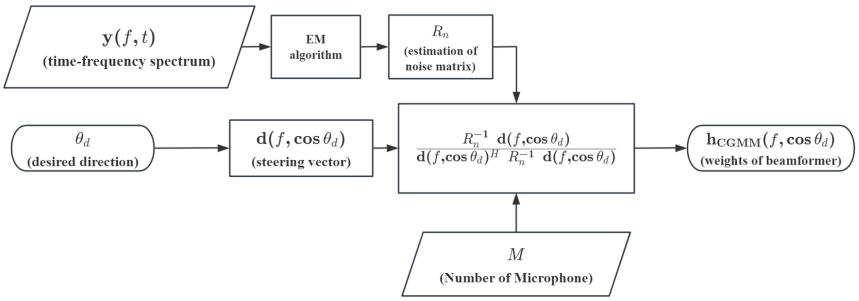
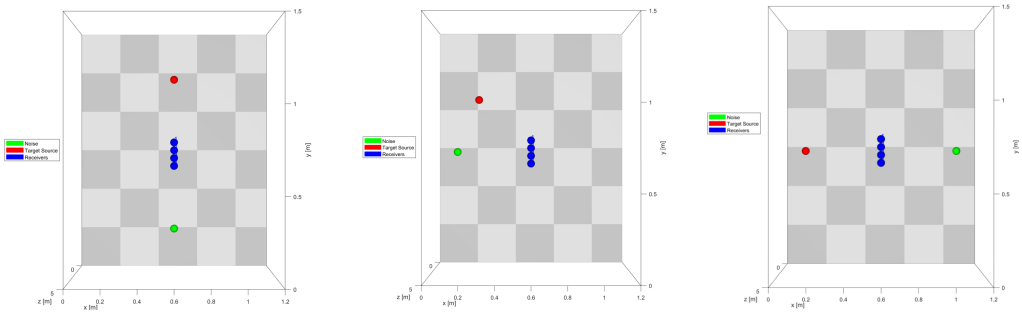


Figure 3.7: Overview of CGMM-MVDR beamformer analysis technique

3.2.5 The configurations of signal sources with microphone array

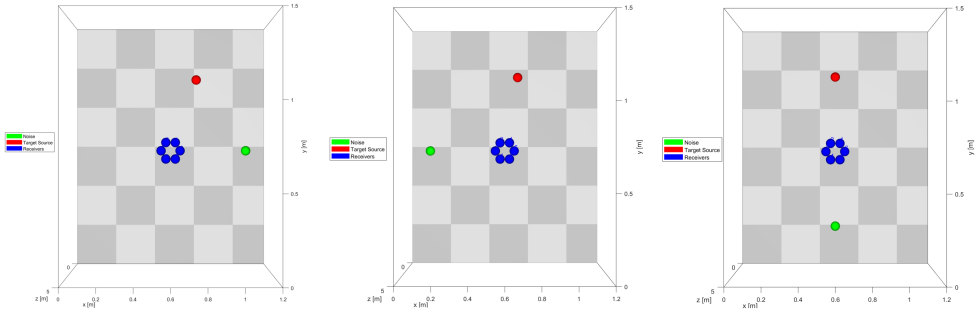
In Section 3.1, we present a variety of combinations to test the beamforming methods with linear array and circular array. However, due to space limitations, only a selection of results will be displayed in the chapter. The Figure 3.8 and Figure 3.9 below represent all the scenarios that will be showcased, and the results presented in following sections will be based on these configurations.



(a) $\theta_d = 90°$ (b) $\theta_d = 135°$ (c) $\theta_d = 180°$

Figure 3.8: Linear microphone circular array configurations

In each configuration of the microphone array, there is only one noise source and one target source, the position of the target source varying in each case. Therefore, we can use the steering direction to the target signal which is defined in Figure 1.1 and Figure 1.2, denoted as θ_d, to represent which configuration of target source and noise is being considered.



(a) $\theta_d = 70°$ (b) $\theta_d = 80°$ (c) $\theta_d = 90°$

Figure 3.9: Linear microphone circular array configurations

3.3 Results and Discussion

3.3.1 Introduction

As described in Section 3.1, the received noisy signal consists of one target source and noises, so the dataset contains hundreds of combinations of target source and noise at different positions. We will only display one case of the noise and speech signal here. Furthermore,for the sake of comparison, we will do the experiments under the same SNR of noise and speech signal.
The Figure 3.10 shows the spectrogram of clean speech signal received from the target source, in the following sections, we will use different beamforming methods to separate it from the received noisy signal.

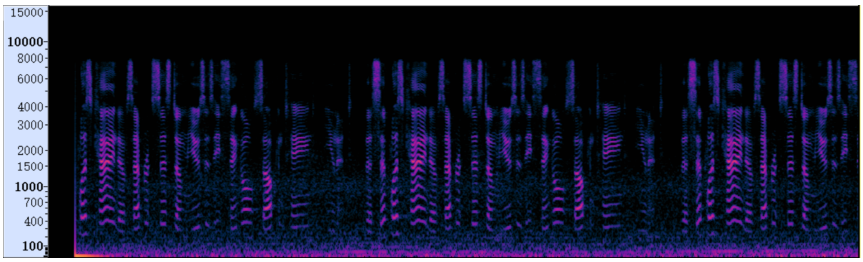


Figure 3.10: The spectrogram of clean speech signal

As mentioned in Section 3.1.2, a total of five types of non-stationary noise were utilized in the testing. Here, we select one of them, the Factory noise, for demonstration. The spectrogram of this type of noise is illustrated in Figure 3.11. It can be observed that

this type of noise is distributed across the range of 0Hz-8kHz, with the primary energy concentrated below 4kHz, and it is non-stationary.

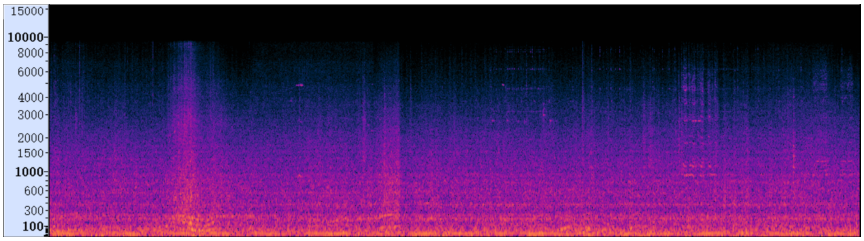


Figure 3.11: The spectrogram of noise

Given the distance between the sound source and the center of the microphone array (as indicated in Table 3.1 and Table 3.2), the noise-laden signals received in these combinations are uniform and its spectrogram is depicted in Figure 3.12 below.

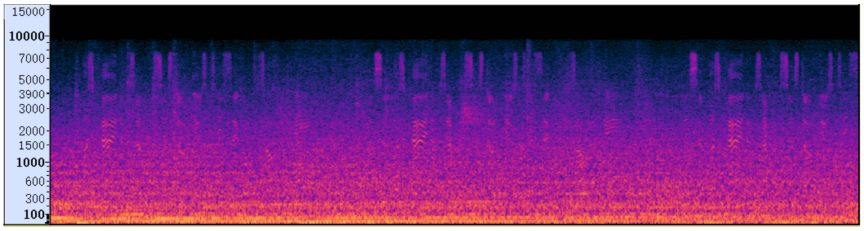
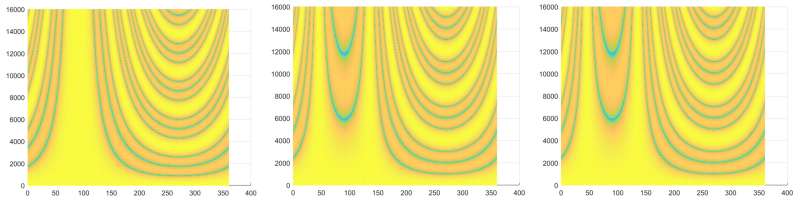


Figure 3.12: Spectrogram of received noisy signal

3.3.2 Beampattern and Result of Delay-Sum Beamformer

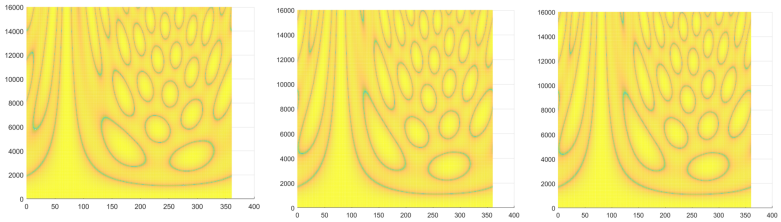
Beampattern of Delay-Sum Beamformer

Delay-Sum Beamformer is a fix beamformer, so the beampattern will not depend on the received signal. Here we will plot three beampattern figures for each array and the azimuth angles of target signal in Figure 3.13 and Figure 3.14.



(a) $\theta_d = 90°$ (b) $\theta_d = 135°$ (c) $\theta_d = 180°$

Figure 3.13: Beampattern of 4 mic linear array Delay-Sum beamformer

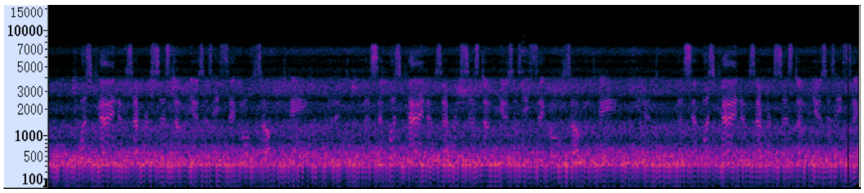


(a) $\theta_d = 70°$ (b) $\theta_d = 80°$ (c) $\theta_d = 90°$

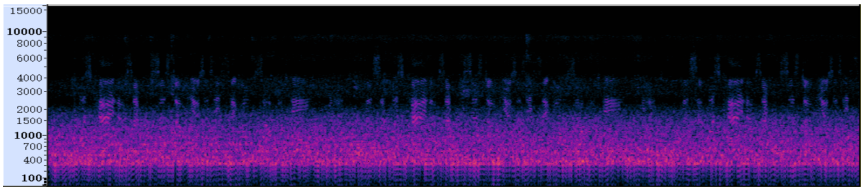
Figure 3.14: Beampattern of 6 mic circular array Delay-Sum beamformer

Results of linear microphone array

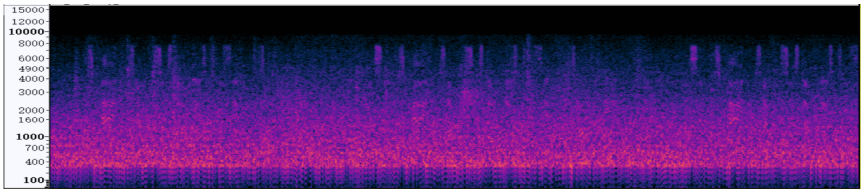
The Figure 3.15 shows that, for the low-frequency parts, due to the wide main lobe of the DS beamformer in Figure 3.13, there is no directional selectivity for the received signal, resulting in significant low-frequency noise in all configurations, for the mid and high-frequency parts, the beamformer's performance is relatively good. As the microphone array is linear, it lacks the ability to discern sound signals with azimuth angles of θ and $\pi - \theta$, as shown in Figure 3.15c, the performance is bad. From the figures it can also be observed that noise isolation is more effective when the sound source is positioned parallel to the linear array, and the effectiveness deteriorates as the source approaches a perpendicular orientation.



(a) $\theta_d = 90°$



(b) $\theta_d = 135°$

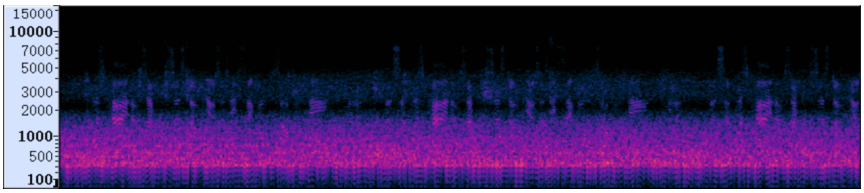


(c) $\theta_d = 180°$

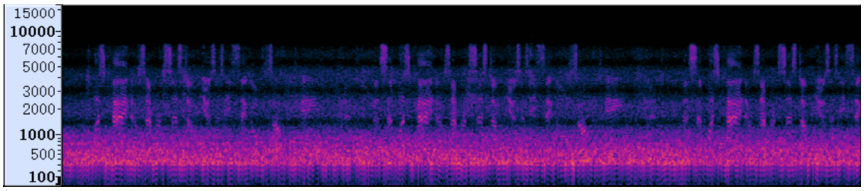
Figure 3.15: Spectrograms of signal applied with Delay-Sum beamformer

Results of circular microphone array

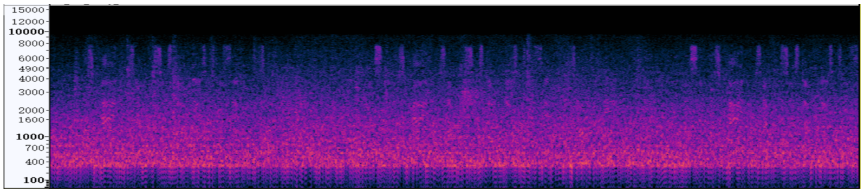
From the Figure 3.16, we can discern that it is like the case of linear array, the noisy signal after beamforming still retains a significant amount of low-frequency noise. However, due to the circle six microphones array having distinct steering vectors in almost all directions, the beampattern diagram in Figure 3.14 does not exhibit symmetrical lobes. Consequently, the circular array outperforms the linear array in terms of versatility.



(a) $\theta_d = 70°$



(b) $\theta_d = 80°$



(c) $\theta_d = 90°$

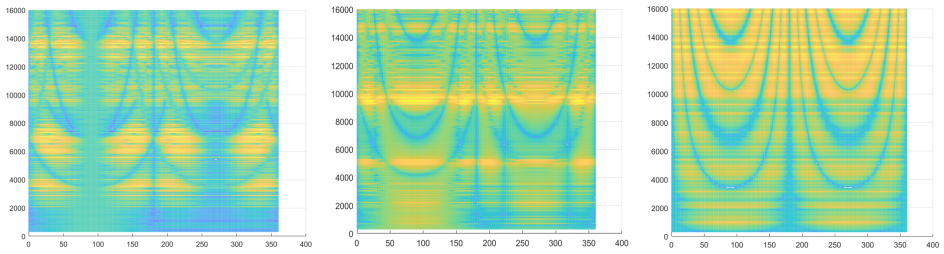
Figure 3.16: Spectrograms of signal applied with Delay-Sum beamformer

3.3.3 Beampattern and Result of MVDR Beamformer

As before, we applied MVDR beamformer on these configurations and obtained figures of the spectrogram results.

Results of linear microphone array

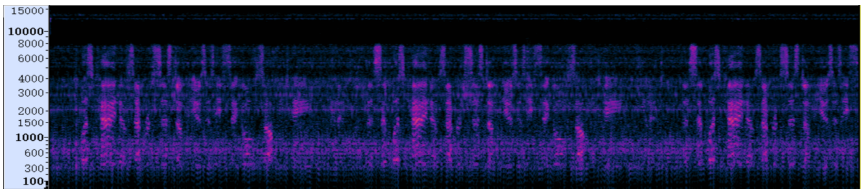
From the Figure 3.17, we can discern that the MVDR beamformer can adaptively generate nulls in the direction of noise sources, while producing peaks in the direction of the target signal, depending on their respective positions, for example, in Figure 3.17a, the null value appears at the azimuth angle of 270°, at the same time, some peaks appear in the target signal we selected, which is at the 90°. Consequently, it can be observed from Figure 3.18a that the elimination of noise in the low-frequency part is significant.



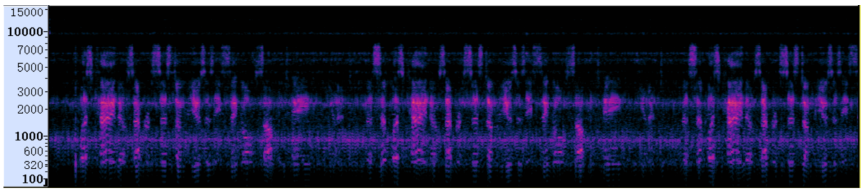
(a) $\theta_d = 90°$ (b) $\theta_d = 135°$ (c) $\theta_d = 180°$

Figure 3.17: Beampatterns of MVDR beamformer

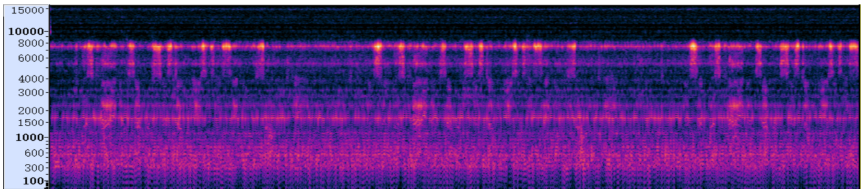
However, noticeable noise is still present in certain frequency bands. This is attributed to the instability of the MVDR beampattern across different frequencies, which is also a core issue in the design of beamformers for speech enhancement. As the speech signal is a broadband signal, it requires consistent gain across various frequencies to minimize distortion. The MVDR beampattern, as illustrated in Figure 3.17b, exhibits numerous discontinuous horizontal lines, which is the cause of continuous noise at certain frequencies. Furthermore, from these three beampattern figures we can discern that the consistent presence of nulls at the azimuth angle of 180°. This occurs because the signals received in this direction are completely coherent, which also results in the incapability to properly receive the target signal in this direction, as shown in the Figure 3.18c.



(a) $\theta_d = 90°$



(b) $\theta_d = 135°$

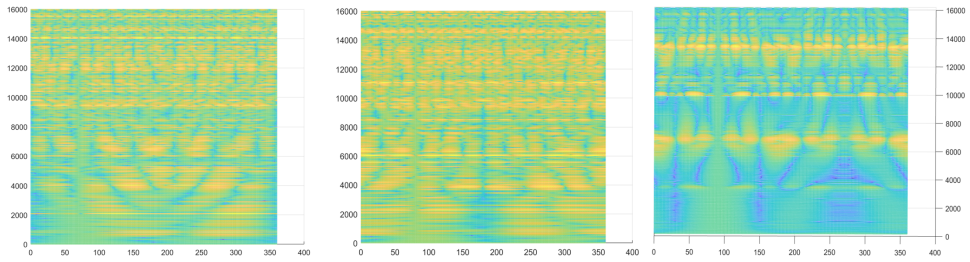


(c) $\theta_d = 180°$

Figure 3.18: Spectrograms of signal applied with MVDR beamformer

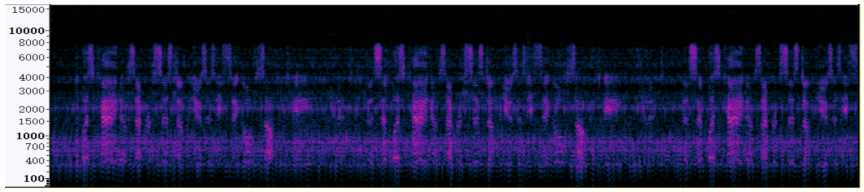
Results of circular microphone array

From Figure 3.19b, we can observe nulls directed towards the noise and prominent peaks in the direction of the target signal, there is also a noted instability along frequencies. However, unlike the beampattern of a linear array, the circular array does not have fixed nulls in specific directions; all nulls are depending on the location of the noise and steering direction, offering greater flexibility without inherent limitations.

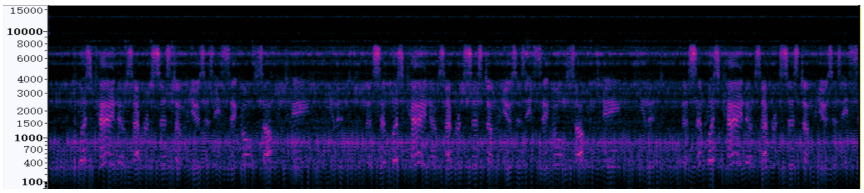


(a) $\theta_d = 70°$ (b) $\theta_d = 80°$ (c) $\theta_d = 90°$

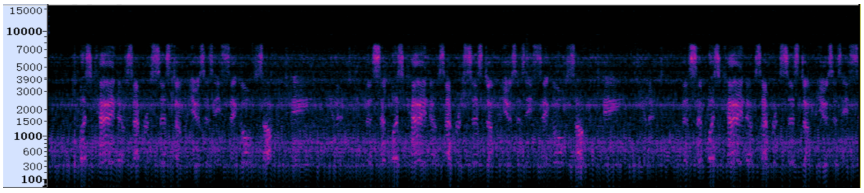
Figure 3.19: Beampatterns of MVDR beamformer



(a) $\theta_d = 70°$



(b) $\theta_d = 80°$



(c) $\theta_d = 90°$

Figure 3.20: Spectrograms of signal applied with MVDR beamformer

3.3.4 Beampattern and Result of AudioSep-MVDR Beamformer

By implementing the AudioSep model, we have obtained the spectrograms of the noise, as illustrated in Figure 3.17. By comparing it with the spectrogram of the noisy speech

presented in Figure 3.14, it can be observed that the spectral components of the speech in the high-frequency range have been removed. This indicates that the model has achieved a certain degree of noise and speech separation. We will proceed to utilize the noise spectrum here to calculate the coefficients for the MVDR beamformer.

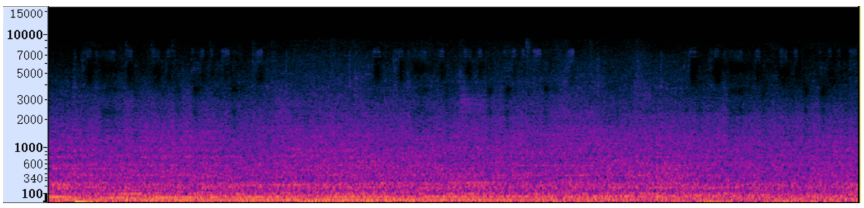
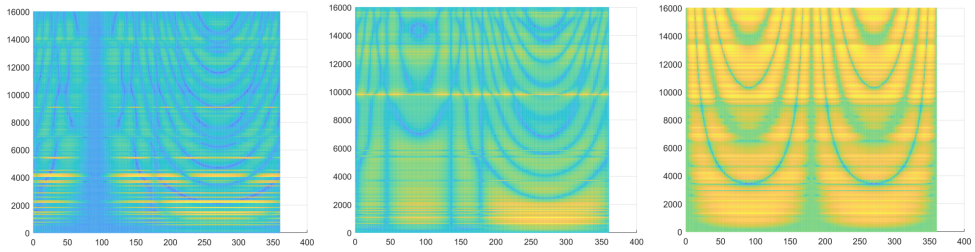


Figure 3.21: The spectrogram of separated noise

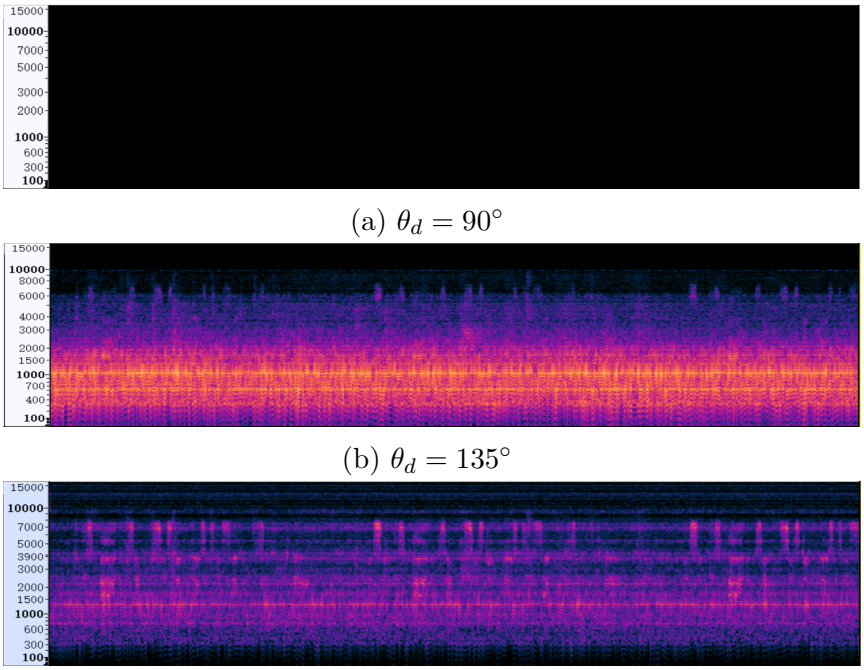
Results of linear microphone array

It can be observed that the performance of this method is highly inadequate. Compared with the conventional MVDR approach in Section 3.3.3, it is evident that the primary issue lies in the estimation of R_n. The root cause is the noise separation performance of AudioSep method. The mono frequency-domain separation model has difficulty in differentiating between the spectra of noise and speech when they overlap, in Figure 3.21, all high-frequency end of noise is removed. Additionally, when apply the model to multiple channels case, the separation results may not be stable, ultimately leading to the self-cancellation of the speech and the retention of noise.



(a) $\theta_d = 90°$ (b) $\theta_d = 135°$ (c) $\theta_d = 180°$

Figure 3.22: Beampatterns of AudioSep-MVDR beamformer



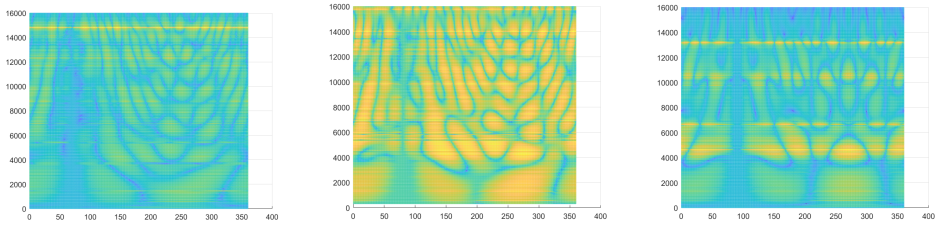
(a) $\theta_d = 90°$

(b) $\theta_d = 135°$

(c) $\theta_d = 180°$

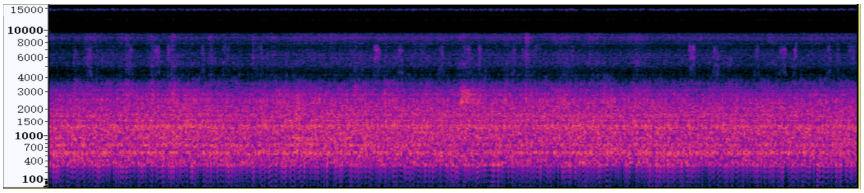
Figure 3.23: Spectrograms of signal applied with AudioSep-MVDR beamformer

Results of circular microphone array

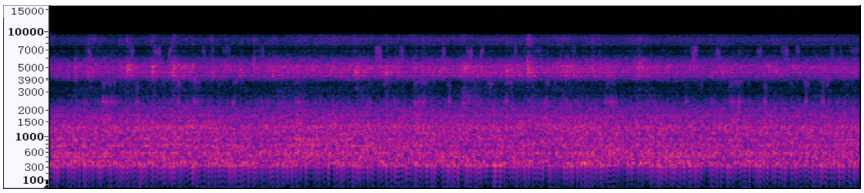


(a) $\theta_d = 70°$ (b) $\theta_d = 80°$ (c) $\theta_d = 90°$

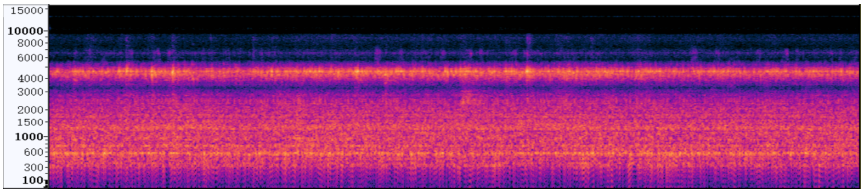
Figure 3.24: Beampatterns of AudioSep-MVDR beamformer



(a) $\theta_d = 70°$



(b) $\theta_d = 80°$



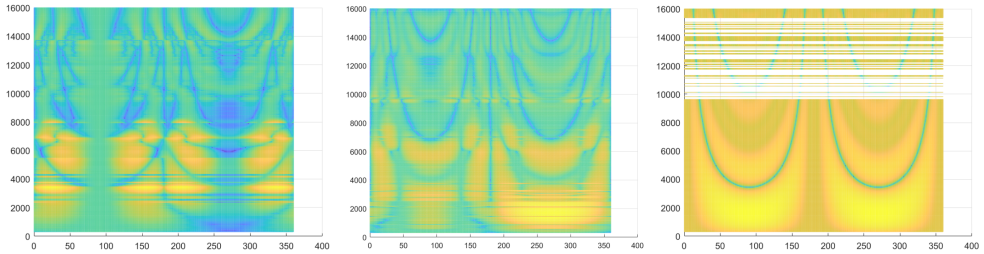
(c) $\theta_d = 90°$

Figure 3.25: Spectrograms of signal applied with AudioSep-MVDR beamformer

3.3.5 Beampattern and Result of CGMM-MVDR Beamformer

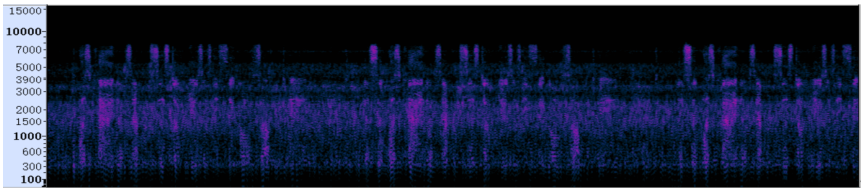
Results of linear microphone array

Through the Figure 3.17 and Figure 3.26, it can be observed that there is a discernible difference between the beampatterns of a linear array when employing the CGMM method to estimate the covariance matrix R_n and when using the mean value for estimation. For instance, there is no fixed null at the 180° direction when using CGMM method. The commonality of them is that the gain in the direction of noise is low, and coupled with significant fluctuations across different frequencies.
These fluctuations also result in noise in certain frequency bands, we will analyze the causes and attempt to improve upon them in the next chapter 4. Additionally, from the Figure 3.26c and Figure 3.27c, we can discern issues due to the geometry of linear array, when the sound sources are close to 180° or 0°, this problem is evident.

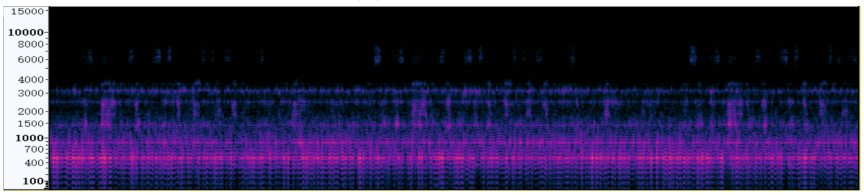


(a) $\theta_d = 90°$ (b) $\theta_d = 135°$ (c) $\theta_d = 180°$

Figure 3.26: Beampatterns of CGMM-MVDR beamformer



(a) $\theta_d = 90°$



(b) $\theta_d = 135°$



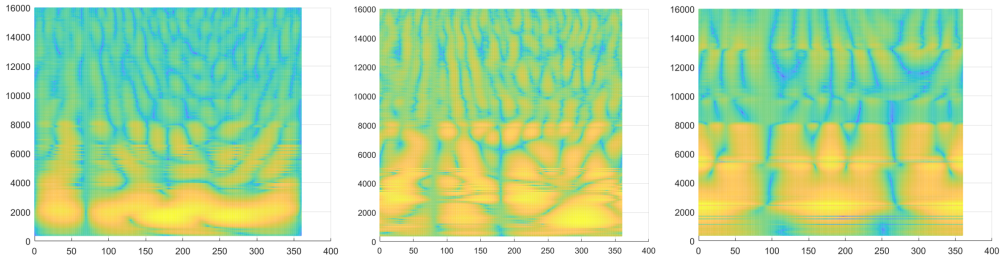
(c) $\theta_d = 180°$

Figure 3.27: Spectrograms of signal applied with CGMM-MVDR beamformer

3.3.6 Results of circular microphone array

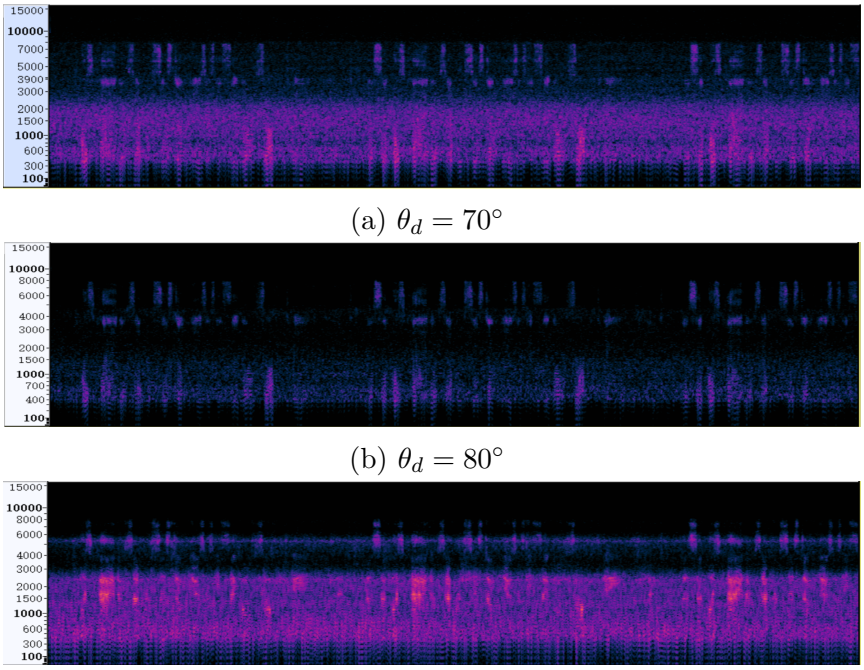
The following figures illustrate the use of the CGMM method with a circular array. We can discern that, like the linear array case presented earlier, there is significant noise in certain frequency bands. This issue stems from the beampattern, where nulls appear in the direction of the target signal, and peaks are directed towards the noise. This is the

primary cause of the poor performance, which we will address in the next chapter.



(a) $\theta_d = 70°$ (b) $\theta_d = 80°$ (c) $\theta_d = 90°$

Figure 3.28: Beampatterns of CGMM-MVDR beamformer



(a) $\theta_d = 70°$

(b) $\theta_d = 80°$

(c) $\theta_d = 90°$

Figure 3.29: Spectrograms of signal applied with AudioSep-MVDR beamformer

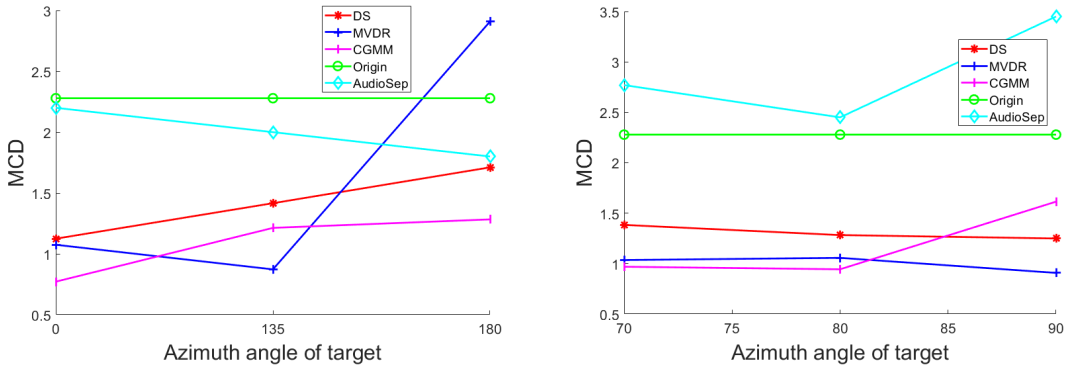
3.3.7 Speech Quality results of microphone arrays

In this section, we conducted an evaluation using subjective metrics in Section 1.4.2 and 1.4.3, which included the four methods mentioned above and the unprocessed noisy signal, denoted as "Origin". The subjective assessment confirmed three key points:

1. The linear array exhibited significant performance variation under different configurations, with the strongest discrimination in the horizontal plane but very bad, almost non-existent, discrimination in the vertical plane. In contrast, the circular array demonstrated a more consistent performance across all directions of targets and noises.

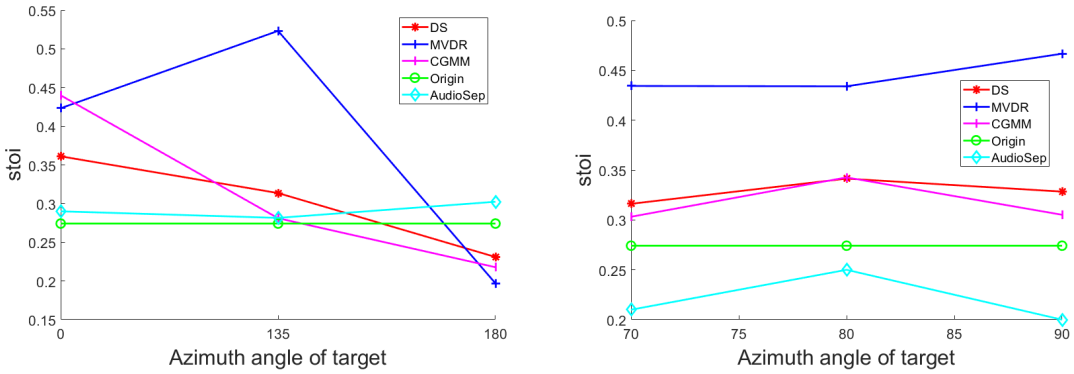
2. The MVDR method and the CGMM-MVDR method performed relatively well, outperforming the other methods, and each showed strengths and weaknesses in certain configurations.

3. The AudioSep-MVDR method performed very poorly, even worse than the unprocessed 'Origin' signal. This indicates that the model may have retained a significant amount of the original noise, leading to the cancellation of the target signal. Additionally, there is a considerable issue with the retention of noise.



(a) with linear array (b) with circular array

Figure 3.30: MCD assessment results



(a) with linear array (b) with circular array

Figure 3.31: STOI assessment results

Chapter 4

Discussion of results and improvement method

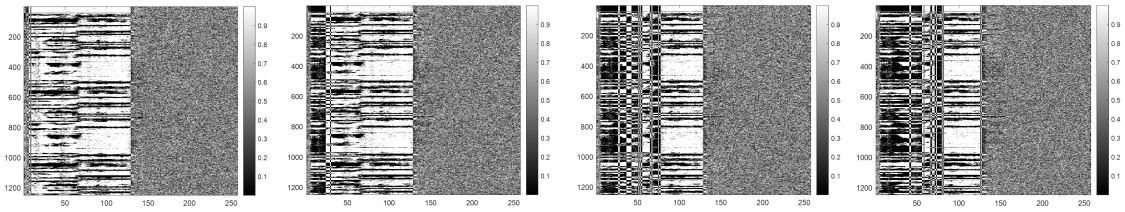
4.1 Comparison and Discussion of Simulation Results

In the previous chapter, we evaluated the performance of linear and circular arrays using various methods and conducted an intelligibility assessment of the processed signal. It was observed that circular array outperformed linear array in terms of stability and adaptability in almost all configurations. Among the different methods, the CGMM-MVDR and MVDR showed superior performance compared to AudioSep and Delay-Sum. However, there is still room for improvement to the CGMM-MVDR method. In this chapter, we will attempt to address the issue and design a post-processing algorithm for it.

4.2 A post-processing method of CGMM-MVDR Beamformer

From the comparison in Section 3.3.7, it was observed that the CGMM-MVDR method underperformed the MVDR method in certain scenarios, which is counterintuitive and unexpected. Through more experiments, a phenomenon that leads to the degradation of performance was identified.
From the Figure 4.1, we can see that when the SNR of input signal less than 1.8dB, the mask we get will be inconsistent along frequency axis. The precision of the mask

directly influences the estimation of R_n. The value of the noisy mask should approximate the time-frequency (TF) mask of the target signal. The noise mask, which excludes the portion of the target signal, essentially serves as the complement of the target signal. It can be observed that the mask estimation is relatively accurate, albeit with inconsistencies across different frequencies. To overcome this problem, we can do some post-processing to reorganize these outcomes to get the correct masks. This post-processing method consists of two steps, the first is to find the frequency part that inconsistent with reference and reorganize them, the second is do classification.



(a) 1.80dB (b) −4.17dB (c) −7.44dB (d) −9.28dB

Figure 4.1: The noise masks with different SNR

4.2.1 Processing for pixel consistency

Here we define an algorithm to find the inconsistent frequency part and reorganization them.
Here, we have the noise mask $\lambda \in R^{M \times N}$, it can be written as:

$$\lambda = \begin{bmatrix} a_{11} & \dots & a_{1N} \\ \vdots & \ddots & \vdots \\ a_{M1} & \dots & a_{MN} \end{bmatrix} \tag{4.1}$$

The objective function is defined as:

$$\begin{aligned} &\min \sum_{i=1}^{M} \sum_{j=1}^{N} |x_{i,j+1} - x_{i,j}| \\ &\text{subject to} \begin{bmatrix} x_{1,k} \\ x_{2,k} \\ \vdots \\ x_{M,k} \end{bmatrix} = \begin{bmatrix} a_{1,k} \\ a_{2,k} \\ \vdots \\ a_{M,k} \end{bmatrix} \text{ or } \begin{bmatrix} 1 - a_{1,k} \\ 1 - a_{2,k} \\ \vdots \\ 1 - a_{M,k} \end{bmatrix} \end{aligned} \tag{4.2}$$

This indicates that the noise mask matrix is consistent across each column. We will attempt to use the columns directly obtained from the EM algorithm or invert their values (by subtracting its values from 1) to achieve a result with the smallest column-wise difference. Here, we propose a heuristic algorithm to solve this problem.

Algorithm 2 Mask reorganization

Initalize : $n \leftarrow 1$
for $l \leftarrow 1$ **to** $N-1$ **do**
    if $\sum_{i=1}^{M} |x_{i,l} - x_{i,l+1}| > \sum_{i=1}^{M} |x_{i,l} - (1 - x_{i,l+1})|$ **then**
        $q_n \leftarrow l$
        $n \leftarrow n + 1$
    end if
end for
for $m \leftarrow 1$ **to** n **do**
    for $l \leftarrow 1$ **to** $N-1$ **do**
        if $(m \bmod 2) = 1$ and $l > q_m$ **then**
            for $k \leftarrow 1$ **to** m **do**
                $x_{k,l} \leftarrow 1 - x_{k,l}$
            end for
        end if
    end for
end for

The explanation of the mask reorganization algorithm is as follows:

1. Calculate the difference for every pair of columns from the first time frame. If the difference between two columns is greater than the difference with one reversed column, it is considered that there is an inconsistency column.

2. Pairing each two inconsistent columns as a segment and invert them.

4.2.2 Mask classification using CNN

We did the reorganization in Section 4.2.1, but we use the first several time-frequency points as reference, so when the reorganization is finished, we will have two unknown masks, one is noise mask, and another is the noisy mask. In practical scenario, sometimes

there is only noise in received signal, we still need to identify such case. To find out which mask is noise mask and which is noisy, we trained a CNN classification model.

Structure of CNN model

Because individual speech signals exhibit a relatively common similarity in spectrogram, such as being distributed within the frequency range of 200 Hz to 8000 Hz and having a broad frequency spread, and in terms of time, individual speech signals typically consist of quasi-steady-state signals ranging from 10ms to 30ms in length, a simple convolutional neural network (CNN) can accomplish the classification.

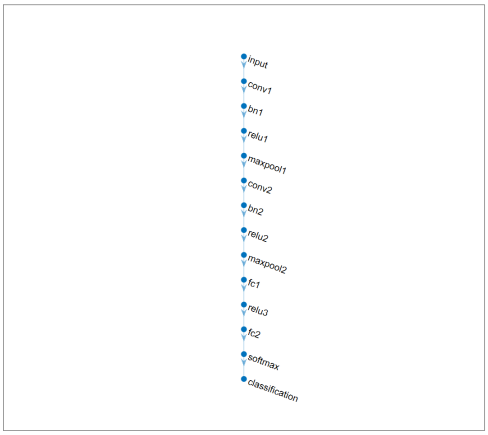


Figure 4.2: Architecture of CNN classification model

Train and Test of CNN model

As mentioned in Section 3.1, we will utilize the masks generated from 200 speech segments to train the classifier, with a training set to test set ratio of 9:1.

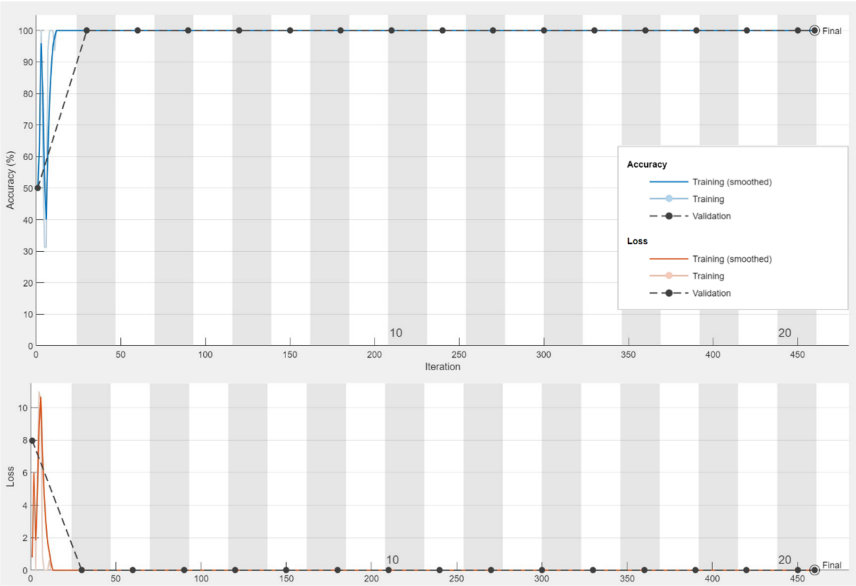
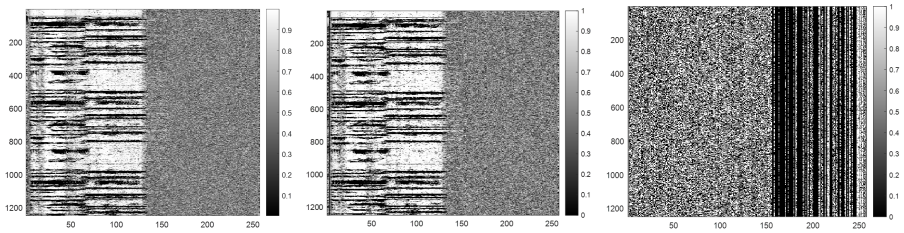


Figure 4.3: Classifier training process

4.3 Beampattern and Results of CGMM-MVDR method with post-processing algorithm

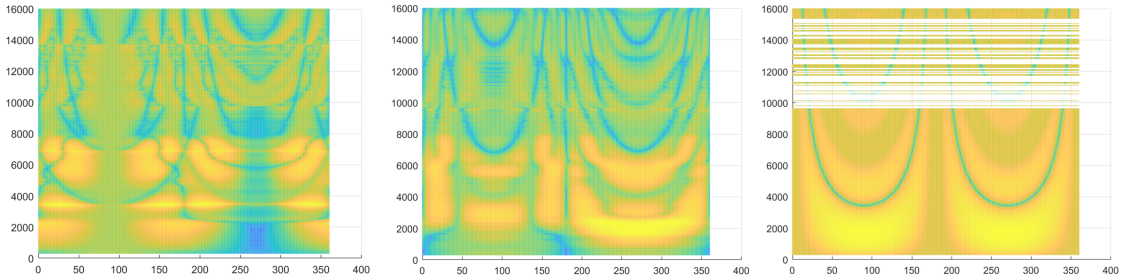
4.3.1 Results of linear microphone array

In this section, we utilize the imprved CGMM-MVDR method and the post-processed masks to estimate R_n and to generate the beamformer. From Figure 4.4, It can be observed that after processing, the mask becomes more consistent and accomplishes correct classification. In Figure 4.5, we can observe that the beampatterns remain relatively stable across different frequencies, without noticeable fluctuations. In Figure 4.5a, the gain is noticeable low at the $\theta_d = 270°$ direction. In Figure 4.5b, there is an interruption in the middle but the gain is comparatively low at the low and high frequencies parts at the direction of $\theta_d = 180°$; these are all indicative the noise is canceled.
The results of spectrograms also indicate that, compared to previous results, the target signal obtained through the post-processing method retains its integrity with minimal distortion. In Figure 4.6b, there is some loss in the high-frequency part; however, the details in the low-frequency range are preserved exceptionally well.
For the 180° case, it can be observed from Figure 4.4c that the generated mask is incapable of distinguishing between noise and the speech signal. This is due to the linear



(a) $\theta_d = 90°$ (b) $\theta_d = 135°$ (c) $\theta_d = 180°$

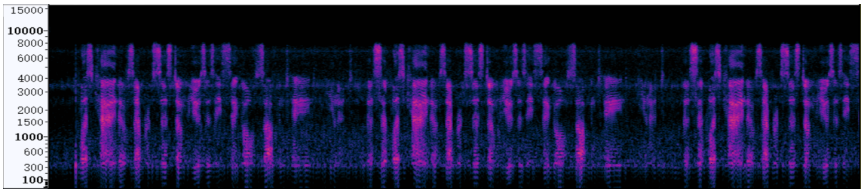
Figure 4.4: Noise masks after post-processing



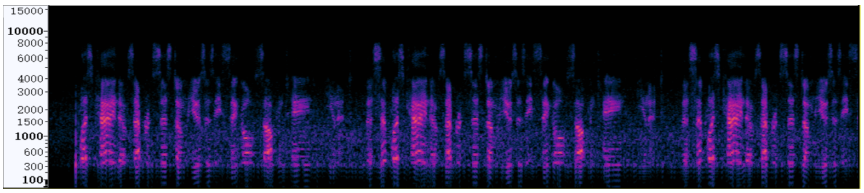
(a) $\theta_d = 90°$ (b) $\theta_d = 135°$ (c) $\theta_d = 180°$

Figure 4.5: Beampattern of post-processed CGMM-MVDR beamformer

array's inability to differentiate the signal directions, a limitation that cannot be overcome by post-processing method.



(a) $\theta_d = 90°$



(b) $\theta_d = 135°$

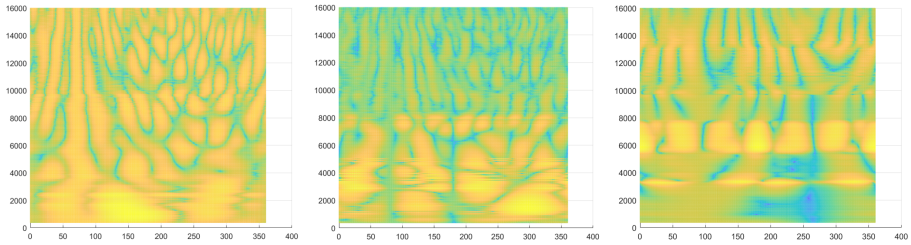


(c) $\theta_d = 180°$

Figure 4.6: Spectrograms of signal applied with post-processed CGMM-MVDR beamformer

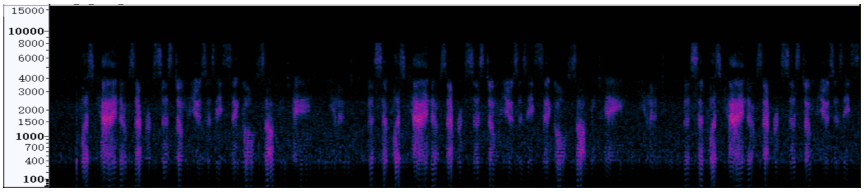
4.3.2 Results of circular microphone array

The performance of the circular array is significantly enhanced with the incorporation of post-processing. The resulting spectrogram indicates that it is approached to the pure speech signal. Furthermore, owing to the geometric properties of the circular array, it can resolve signals received from any direction, thus avoiding the issue encountered with linear arrays where signals from the perpendicular direction cannot be distinguished.

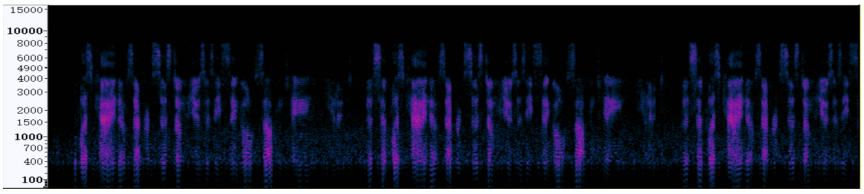


(a) $\theta_d = 70°$ (b) $\theta_d = 80°$ (c) $\theta_d = 90°$

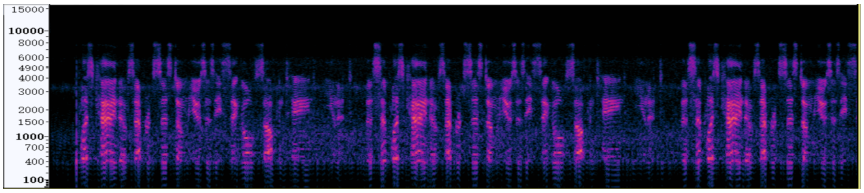
Figure 4.7: Beampattern of post-processed CGMM-MVDR beamformer



(a) $\theta_d = 70°$



(b) $\theta_d = 80°$



(c) $\theta_d = 90°$

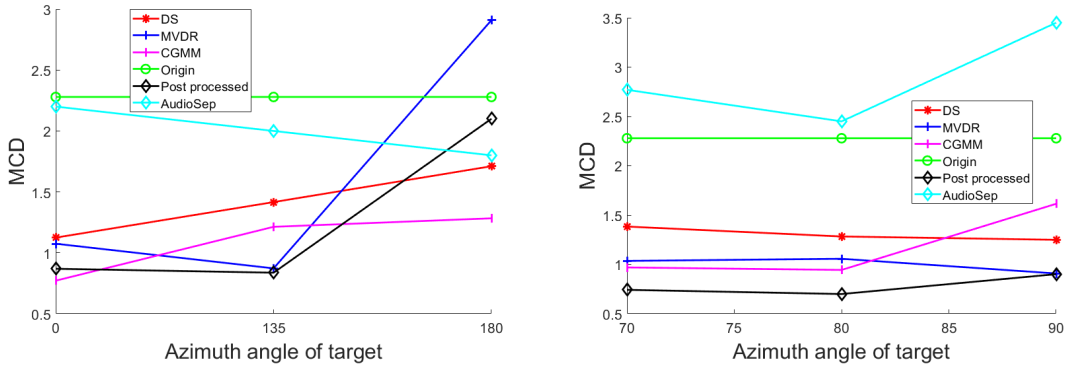
Figure 4.8: Spectrograms of signal applied with post-processed CGMM-MVDR beamformer

4.3.3 Speech Quality results of post-processed CGMM-MVDR beamformer

Similarly the Section 3.3.7, we do the subjective assessment here to confirm the efficacy of post-processing method. We can summarize the assessment into two points:

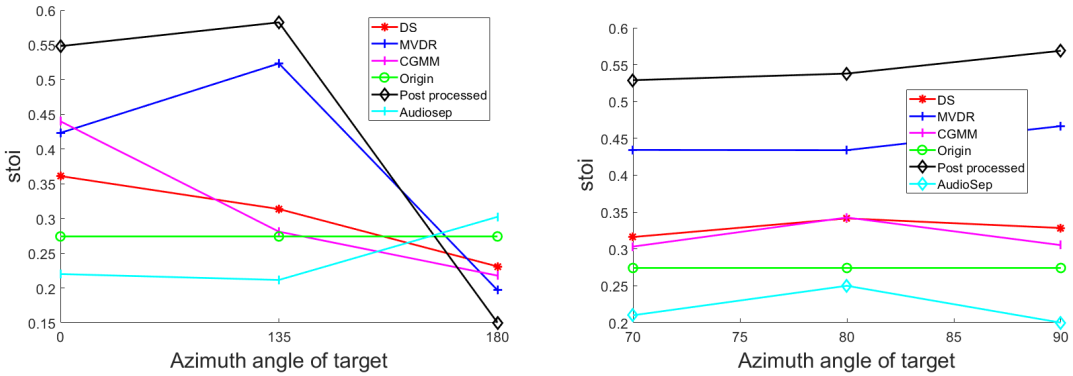
1. It is evident that among the arrays of both geometric shapes, STOI values obtained with post-processing are higher than other methods except at the direction of $\theta_d = 180°$. Similarly, MCD values are lower than almost all other approaches.This demonstrates the effectiveness and a certain level of stability of the post-processing method.

2. The performance of post-processing in the circular array is more stable and significantly superior to all other methods.



(a) with linear array (b) with circular array

Figure 4.9: MCD assessment results



(a) with linear array (b) with circular array

Figure 4.10: STOI assessment results

Chapter 5

System Design

5.1 Introduction

In this chapter, we will build a prototype and test the feasibility of the improved CGMM-MVDR Beamformer in real world. We will make a device that integrates microphone array, processor with input and output functions, and can be controlled by Android app on mobile phone. Due to the requirements of portability and reducing device dependence, the edge device should offload the high computational tasks to a server, so we placed the components with low latency requirements and low computational load on the edge device connected to the microphone array, and the parts with high computational demands and without low latency requirements on a remote server. The computational results are then transmitted back to the edge via a communication protocol MQTT so that the functions can be deployed on simple devices.

Regarding the control functionality of the Android app, our want to extend beyond merely sending start and stop instructions. We aim to implement a control method based on gesture interaction. In this scenario, we assume that the microphone array is near the mobile phone running the Android app. This assumption ensures that the direction in which the phone is pointed is consistent with that of the microphone array, so we can use the phone to point at our target signal to control the beamformer to steering it. By pointing at the target signal, we can use the orientation sensor on mobile phone to get the orientation information, then generate a steering vector and apply it on the beamformer. Consequently, users holding the phone can control the reception of sound from any desired direction based on the which direction it is pointing at.

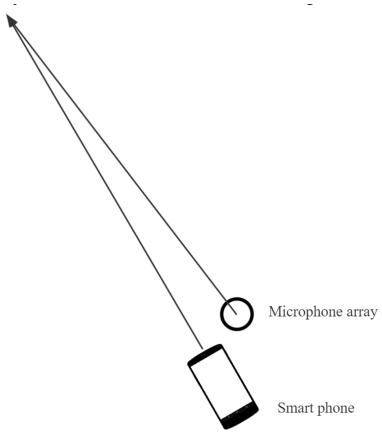


Figure 5.1: Beamformer orientation control schematic diagram

5.2 Hardware Design

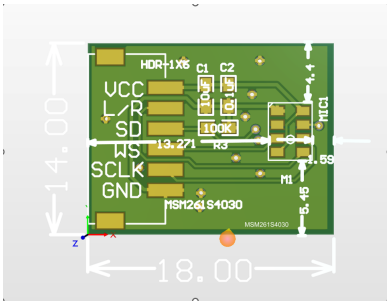
5.2.1 Microphone and Board

According to our comparison and discussion in Section 4.1, we will choose the circular microphone array to achieve a robust beamforming robust beamforming for signals coming from all directions. Specifically, we utilized six omnidirectional, top-ported I2S digital output MEMS microphones and a board capable of providing power and clock signal to the microphones, formatting the multi-channel data to comply with the UAC (USB Audio Class) standard. Then, the microphone array is connected to a Raspberry Pi, which serves as the edge device, via a USB 2.0 interface.

5.2.2 Edge device

The edge device is a Raspberry Pi 4 Model B with 2 GB of RAM.
This device supports Wi-Fi and Bluetooth, is equipped with multiple USB interfaces, and can perform simple computational tasks.

Finally, connect the microphone array to the Raspberry Pi, and the entire assembly is housed within a cylindrical enclosure to form our prototype, with the microphones placed at the top to better capture sound.



(a)

SPECIFICATIONS

All data taken at 25°C, Relative Humidity 45±5% L/R pin grounded unless otherwise specified
Vdd=1.8V, clock frequency=3.072MHz

	Limits			unit	condition
	Min.	Nom.	Max.		
Directivity		Omni directional			
Sensitivity	-27	-26	-25	dB	dBFS @1kHz 1Pa
Operation voltage	1.6		3.6	V	
Freq. range		Refer to the frequency response		Hz	
Sensitivity loss across supply voltage		No change across the voltage range		dB	
Signal to noise ratio	-	61	-	dB	20 kHz bandwidth, A-weighted
THD	-	0.2	-	%	94dB SPL @1kHz S =Nom, Rload > 2 k
AOP	-	124	-	dB SPL	10% THD @1kHz S =Nom, Rload > 2 k
Polarity		Increasing sound			Increasing density of 1's
PSR		-72		dBFS(A)	
Current consumption	-	750	1000	μA	Normal mode
	-	400	-	μA	Low power mode
Clock frequency	1.0	3	4.0	MHz	Normal mode
	150	-	800	KHz	Low power mode
Storage temperature	-40	-	100	°C	

(b)

Figure 5.2: Detailed specification of microphones

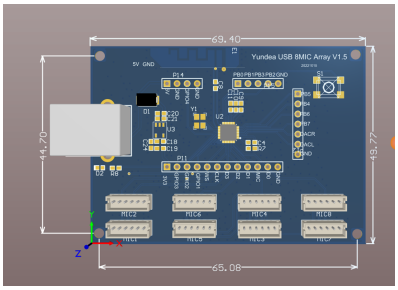


Figure 5.3: The layout of the board

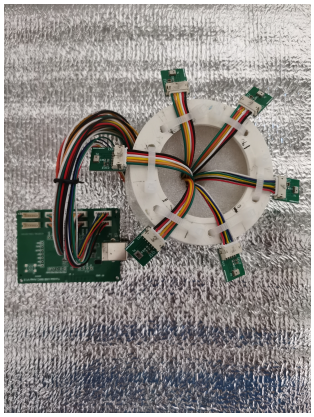


Figure 5.4: Six-microphone circular array

5.2.3 Software Design

As previously mentioned, due to considerations of portability and cost, we utilize a Raspberry Pi as the edge device. Consequently, the computationally intensive EM algorithm is executed on the server, after which the calculated R_n is transmitted back to the edge device. Then R_n is combined with the direction instruction from the mobile app to generate real-time beamformer weights. The edge device then employs these weights to process received audio signal.



Figure 5.5: The edge device



Figure 5.6: The prototype

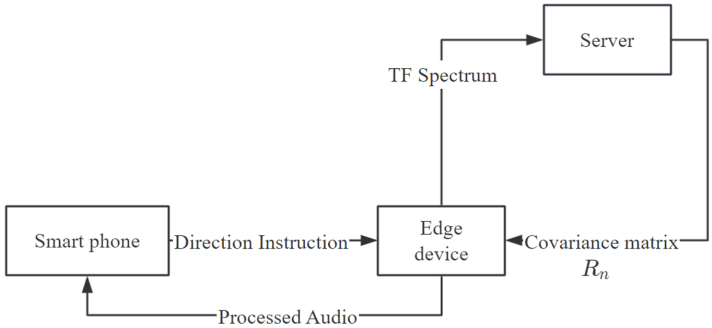


Figure 5.7: The flow chart of data processing procedures

5.2.4 Edge device and Server Software Design

Edge device

As depicted in Section 5.1, we will perform tasks requiring timely processing on the edge device, which can be divided into two parts. The first part involves audio signal beamforming processing. Initially, the received audio signals undergo Discrete Time Fourier Transform (DTFT) to convert into a spectrogram. Subsequently, as shown in Equation 1.3, beamforming is conducted, followed by an Inverse Discrete Time Fourier Transform

(IDTFT) to revert to the time-domain signal. The second part entails determining the steering vector based on the angles sent from the Android app, which is then used to update the beamformer's weights in real-time. These weights are subsequently used in the beamforming step mentioned in the first part.
All information exchange processes here, including the interactions between the Android app and the server, are based on the MQTT protocol, thus requiring the internet connection.

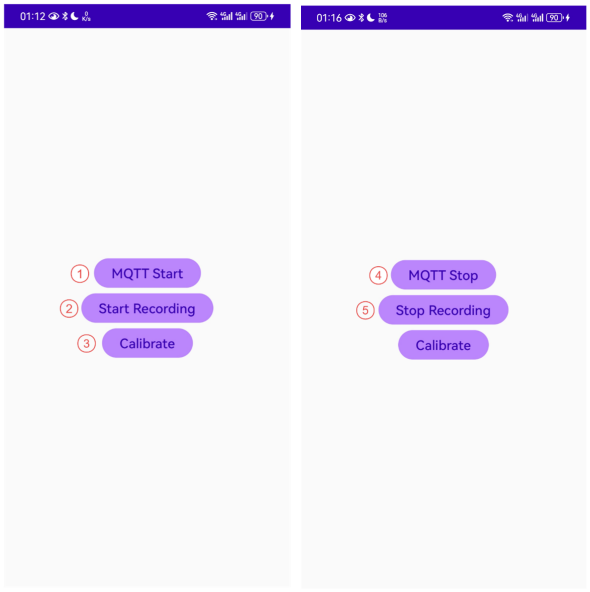
Server

For the server, it receives a time-frequency (TF) spectrum with a duration of 15 seconds. Based on this spectrum, the EM iterative algorithm will be executed. Since the variation in ambient noise is generally not excessive or rapid, the delay caused by this process, which includes the 15-second duration plus the time required for the EM algorithm to process, will not significantly impact the performance.

Android APP Design and Communication

In this section, we will introduce the UI design and operation logic of the Android app.

- Button ①: When MQTT is not connected, it will display this text; pressing this button will attempt to connect to the specified MQTT broker, with a Toast notification indicating success or failure.
- Button ②: Upon successful connection to the MQTT broker, this button can be used to initiate audio recording. Once activated, orientation instructions will be sent to the edge device, which will then begin to receive and process signals from the microphone array. Currently, the processed audio will be saved as a file.
- Button ③: As depicted in Figure 5.1 , we will use the mobile phone's orientation sensor to get the direction it is pointing at and control the beamformer. Therefore, prior to use, it is necessary to calibrate the orientation of the phone with the microphone array. A calibration marker is drawn on the prototype. By placing the phone parallel to the marker and pressing the button, the calibration can be completed.
- Button ④: Once the MQTT connection to the broker is established, the text of Button ① will display "MQTT Stop." To terminate the connection, use this button.



(a) (b)

Figure 5.8: The states of UI buttons

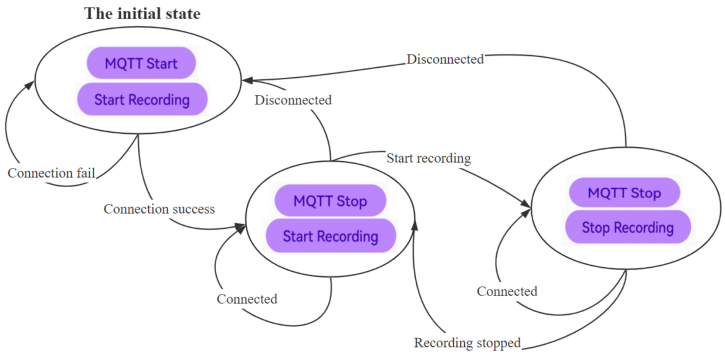


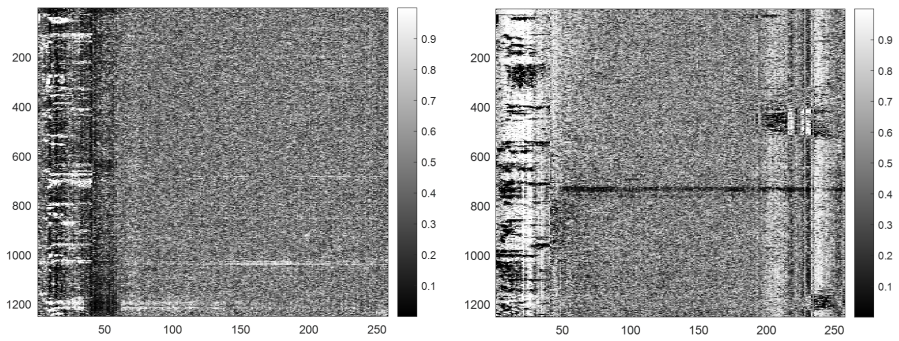
Figure 5.9: The states and flow of the app

- Button ⑤: Like Button ④, this button is used to cease recording.

As depicted in Figure 5.9, recording button can only be initiated after a successful MQTT connection has been established. Regarding the calibration feature, it can be used in all states.

5.3 Results and Conclusions

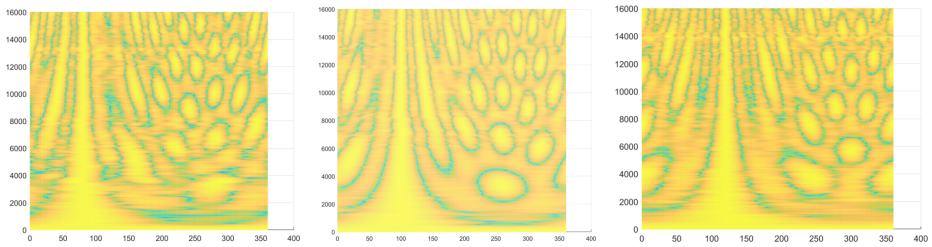
The Figure 5.10 presents the noise and noisy masks in real-world scenarios when receiving a noisy speech signal in an office. It can be observed that through the application of the CGMM model, the EM algorithm, and the reorganization process in post-processing, the two masks can be effectively generated. Additionally, the classification model based on CNN is crucial to distinguish between noise mask and noisy mask.



(a) Noisy mask (b) Noise mask

Figure 5.10: TF masks in real test

From the Figure 5.11, it can be observed that for received with high SNR, the beamformer generated is close to the Delay-Sum beamformer. However, in the low-frequency range, it is narrower and more effective than the conventional Delay-Sum beamformer.



(a) $\theta_d = 80°$ (b) $\theta_d = 100°$ (c) $\theta_d = 120°$

Figure 5.11: Beampatterns of prototype

Chapter 6

Final Conclusions and Future Work

6.1 Conclusions

The goal of this thesis and work is to further develop the application of beamforming using microphone arrays and to evaluate its performance. By creating a large amount of test data through recording and simulation techniques, overall trends were determined as to the improvement beamformers provide over noise isolation scenarios. Using the objective speech intelligibility metrics and spectrum distortion assessment allowed for the measurement of improvement by the beamforming technique. A series of experiments were presented and their results have led to the following:

- We conducted simulations to test four types of beamformering methods on both linear and circular microphone arrays to variety of direction combinations of target signal and noise. Then, we compared and analyzed the characteristics and performance of each method. Subsequently, the comparison concluded that the circle microphone array is more stable, and the CGMM-MVDR method has a higher upper-bound in performance.
- By observing the noisy mask generated by the CGMM-MVDR method, we propose post-processing steps to enhance the performance of it. Utilizing a substantial amount of data to validate and train, we developed a post-processing method that includes two steps: reorganization and classification. We also verified the effectiveness of this method through experiments.

After simulations, we set up a prototype with microphones and a Raspberry pi and we made the EM algorithm and beamforming process to be carried out separately on the edge

device and server, thereby better reducing the computational requirements of the devices and ensuring low latency. We have also developed an app for Android devices that allows users to control the direction of the beamformer with gesture interaction, enabling users to conveniently receive speech signal from a desired direction.

6.2 Limitations and Future Work

In this work, we employed the CGMM-MVDR and post-processing method on circle microphone array to generate beamformers targeted at specific directions, and we constructed a prototype to implement this method. However, during the experiments, we also identified issues with the current approach that deserve further exploration and study.

- The CGMM-MVDR method is adaptive, hence the gain is not stable under many circumstances. Consequently, it is need to incorporate an algorithm akin to wide dynamic range compression (WDRC) following beamforming to standardize the processed signal strength within a certain range.
- Existing time-frequency (TF) mask methods, including the CGMM-MVDR approach we employ, are capable of generating only two masks: one for noise and another for the target signal or the noisy signal. Therefore, they are incapable of handling more than one sound source, as it would aggregate multiple sound sources into a single mask.
- Due to limitations in experimental facilities and equipment, we have not been able to reproduce the various sound sources and noise scenarios from the simulation in our prototype.

References

[1] Kuperman W A and Ingenito F. *Spatial correlation of surface generated noise in a stratified ocean*. The Journal of the Acoustical Society of America, 67(6): 1988-1996, 1980.

[2] Wabnitz A, Epain N, Jin C, and et al. *Room Acoustics Simulation for Multichannel Microphone Arrays[J]*. Proceedings of the International Symposium on Room Acoustics, Melbourne, August 2010.

[3] Moonen M Andersen K T. *Robust Speech-Distortion Weighted Interframe Wiener Filters for Single-Channel Noise Reduction[J]*. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 26(1): 97-107, 2018.

[4] Uhl C. *Experiments with an extended adaptive SVD enhancement scheme for speech recognition in noise[J]*. ICASSP'01, 1: 281-284, 2001.

[5] Ngoc Q.K. Duong, Emmanuel Vincent, and Rémi Gribonval. *Spatial covariance models for under-determined reverberant audio source separation=*. 2009 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, 2009.

[6] Cron B F and Sherman C H. *Spatial-correlation functions for various noise models*. The Journal of the Acoustical Society of America, 34(11): 1732-1996, 1962.

[7] J. S. Garofolo, L. F. Lamel, and W. M. Fisher. *TIMIT Acoustic-Phonetic Continuous Speech Corpus (LDC93S1)*. Linguistic Data Consortium, 1993.

[8] John Garofolo, David Graff, and Doug Paul. *Csr-i (wsj0) complete ldc93s6a*. Linguistic Data Consortium, web download edition, 1993.

[9] Cox H, Zeskind R M, and Owen M M. *Robust adaptive beamforming*. IEEE Trans Audio Speech Signal Process, 35(10): 1365-1376, 1987.

[10] Taal C H, Hendriks R C, Heusdens R, and et al. *An Algorithm for Intelligibility Prediction of Time–Frequency Weighted Noisy Speech[J].* IEEE Transactions on Audio Speech & Language Processing, 19(7): 2125-2136, 2011.

[11] Eric W. Healy and et al. *An algorithm to improve speech recognition in noise for hearing-impaired listeners.* The Journal of the Acoustical Society of America, pp. 3029–3038, 2013.

[12] T. Higuchi, N. Ito, T. Yoshioka, and T. Nakatani. *Robust mvdr beamforming using time-frequency masks for online/offline asr in noise.* ICASSP. IEEE, pp. 5210–5214, 2016.

[13] O. Hoshuyama, A. Sugiyama, and A. Hirano. *A robust adaptive beamformer for microphone arrays with a blocking matrix using constrained adaptive filters.* IEEE Trans. Signal Process, vol. 47, no. 9, pp. 2677–2684, 1999.

[14] Cohen I, Benesty J, Gannot S, and et al. *Speech Processing in Modern Communication – Challenges and Perspectives[J].* The Journal of the Acoustical Society of America, 129(1): pp. 535, Tianjin, 2011.

[15] Capon J. *High-resolution frequency-wavenumber spectrum analysis.* Proceedings of the IEEE, 57(8): 1408-1418, 1969.

[16] Griffiths L J and Jim C W. *An alternative approach to linear constrained adaptive beamforming[J].* IEEE Transactions on Antennas and Propagation, 30(1): 27-34, 1982.

[17] Y. Jiang, H. Zhou, and Z. Feng. *Performance analysis of ideal binary masks in speech enhancement.* 2011 4th International Congress on Image and Signal Processing, Vol. 5, pp. 2422–2425, 2011.

[18] K. Kumatani, J. McDonough, B. Rauch, D. Klakow P. N. Garner, and W. Li. *"Beamforming with a maximum negentropy criterion.* IEEE trans. Audio, Speech, Language Process, vol. 17, no. 5, pp. 994–1008, 2009.

[19] Flanagan J L, Johnston J D, Zahn R, and et al. *Computer-steered microphone arrays for sound transduction in large rooms[J].* The Journal of the Acoustical Society of America, 78(5): 1508-1518, 1985.

[20] H. Li, Q. Zhao, G. Ren, and et al. *Speech Enhancement Algorithm Based on Independent Component Analysis[C].* Fifth International Conference on Natural Computation, pp. 598-602, Tianjin, 2009.

[21] Xubo Liu, Qiuqiang Kong, Yan Zhao, and et al. *Separate What You Describe: Language-Queried Audio Source Separation*. Proc. Interspeech 2022, 2022.

[22] Jianfen Ma, Yi Hu, and Philipos C. Loizou. *Objective measures for predicting speech intelligibility in noisy conditions based on new band-importance functions*. The Journal of the Acoustical Society of America, pp. 3387–3405, 2009.

[23] A. Mahmoodzadeh and et al. *"Binaural speech separation based on the time-frequency binary mask*. 6th International Symposium on Telecommunications (IST), pp. 848–853, 2012.

[24] Iain McCowan, Jason Pelecanos, and Sridha Sridharan. *Robust Speaker Recognition using Microphone Arrays*. IN PROCEEDINGS OF 2001: A SPEAKER ODYSSEY, 2001.

[25] MQTT.org. *MQTT Version 5.0 OASIS Standard*. OASIS, http://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.pdf, 2019.

[26] Nicoleta Roman and John Woodruf. *Intelligibility of reverberant noisy speech with ideal binary masking*. The Journal of the Acoustical Society of America, pp.2153–2161, 2011.

[27] Bharti S S, Gupta M, and Agarwal S. *A new spectral subtraction method for speech enhancement using adaptive noise estimation[C]*. International Conference on Recent Advances in Information Technology, India, 2016.

[28] M. Souden, J. Benesty, and S. Affes. *On optimal frequency domain multichannel linear filtering for noise reduction*. IEEE Trans. Audio, Speech, Language Process, vol. 18, no. 2, pp. 260–276, 2007.

[29] Soundararajan Srinivasan, Nicoleta Roman, and De Liang Wang. *Binary and ratio time-frequency masks for robust speech recognition[J]*. Speech Communication, 48(11): 1486-1501, 2006.

[30] S. Stenzel, J. Freudenberger, and G. Schmidt. *A minimum variance beamformer for spatially distributed microphones using a soft reference selection[C]*. 4th Joint Workshop on Hands-Free Speech Communication and Microphone Arrays, pp. 127–131, Nancy, ITALY, 2014.

[31] K. Tan, J. Chen, and D. Wang. *Gated Residual Networks with Dilated Convolutions for Monaural Speech Enhancement[J]*. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 27(1): 189-198, 2019.

[32] Emiya V, Vincent E, Harlander N, and et al. *Subjective and Objective Quality Assessment of Audio Source Separation[J].* IEEE Transactions on Audio, Speech, and Language Processing, 19(7): 2046-2057, 2011.

[33] D Wang. *Time–frequency masking for speech separation and its potential for hearing aid design.* Trends in Amplification, pp. 332–353, 2008.