Imperial College London

Department of Electrical and Electronic Engineering

Final Year Project Report 2018

Project Title:	**Audio Signal Zoom for Small Microphone Arrays**
Student:	**Chi Hang Leung**
CID:	**00978321**
Course:	**4T**
Project Supervisor:	**Dr Patrick Naylor**
Second Marker:	**Mr Mike Brookes**

Abstract

Nowadays, often do people use their smartphones or cameras to capture videos of the memorable moments. It is convenient to visually zoom in towards the subject of interest and suppress the unwanted interferences. A problem is then posed - whether the same could be achieved in audio terms. Despite demonstrating a high spatial selectivity, current approaches usually require large microphone arrays. This motivates the project to explore and develop alternative approaches that are suitable for small devices like smartphones.

This report firstly outlines the process of software simulation to capture audio containing two persons talking. Using the synthetic audio data, the report proposes several audio zooming algorithms, including time-frequency masking, beamforming and machine learning. The performance of the zooming algorithms under different reverberant conditions is then evaluated in terms of speech quality and intelligibility through both subjective and objective metrics. The report ultimately provides an insight on the practicality of implementing the algorithms on smartphones.

Acknowledgements

I would like to firstly express my gratitude towards my project supervisor, Dr. Patrick Naylor, for his insightful advice and guidance throughout the project.

I would also like to thank the Speech and Audio Processing Laboratory for providing me with the anechoic audio data needed in the project, and Mr. Aidan Hogg for his help with the evaluation section of the project.

Lastly, I would like to thank my parents for their unconditional support through the years. I also wish to express my great appreciation to Mr. Chak Man Tang on helping with the visualisation of the project concept, as well as whom took part in the listening test.

Contents

List of Figures

List of Tables

List of Abbreviations

BSS	Blind Source Separation
DNN	Deep Neural Network
DOA	Direction of Arrival
DRR	Direct-to-Reverberant Ratio
DSB	Delay-Sum Beamformer
FCM	Fuzzy c-means Clustering
GSC	Generalised Sidelobe Canceller
LTI	Linear Time-invariant
MOS-LQO	Mean Opinion Score - Listening Quality Objective
MUSHRA	MUltiple Stimuli with Hidden Reference and Anchor
MVDR	Minimum Variance Distortionless Response
PESQ	Perceptual Evaluation of Speech Quality
RIR	Room Impulse Response
RT_{60}	Reverberation Time
SAR	Sources-to-Artifacts Ratio
SDR	Signal-to-Distortion Ratio
SIR	Source-to-Interference Ratio
SNR	Signal-to-Noise Ratio
STFT	Short-time Fourier Transform
STOI	Short-time Objective Intelligibility
TDOA	Time Difference of Arrival
TF	Time-frequency
wCFCM	Weighted Contextual Fuzzy c-means Clustering
wFCM	Weighted Fuzzy c-means Clustering

1 Introduction

Often one would take videos using their smartphones to capture the precious moments, such as parties and ceremonies. However, inevitably there would be interference, where one would zoom the camera towards the point of interest to avoid the unwanted subjects. A similar question aroused - whether one can make the sound of the speaker of interest closer in the context of smartphones.

This problem resonates with the well-known 'Cocktail Party Problem' illustrated by Colin Cherry in 1953 [1]. Human brains have the ability to amplify the sound coming from a certain direction using their phase difference of arrival to two ears. Hence, under noisy environments such as cocktail party, with a lot of people speaking, the person one is chatting with is still audible. Over the years, different signal processing techniques have been developed to try replicating the processing by human ears and brains to microphones and computer. [2] Despite being an established technique to achieve audio zoom, classical beamforming technique requires large microphone arrays typically to provide a narrow and precise beam. For small devices like phones in this project's context, developing alternative approaches is required.

Hence, this project seeks to investigate both classical and novel or a combination of algorithms to achieve acoustic zooming in the context of smartphones. The purpose of the project is to capture audio samples containing multiple speakers, which then could be used to develop an algorithm that can zoom (in audio terms) towards one selected sound source. The nature of the project is software-based, with MATLAB being the main platform for development. The code and results of the project is available on `https://github.com/chl214/FYP`[1]

Figure 1: Concept of Audio Zoom on Smartphones

[1]Anechoic speech data used in the project is excluded from the repository due to confidentiality concern

1.1 Scope and Aims

The stated problem can be decomposed into two parts, one part as identifying the direction of the speaker with respect to the smartphone, with the next part being processing the received signals to zoom into a speaker of interest and suppress the other. In the context of this project, the focus is to develop the algorithm for the latter part. Hence, it is assumed throughout the project that the direction of arrival (DOA) of the speech from the speakers to the microphones, as illustrated in Figure 2b, are known.

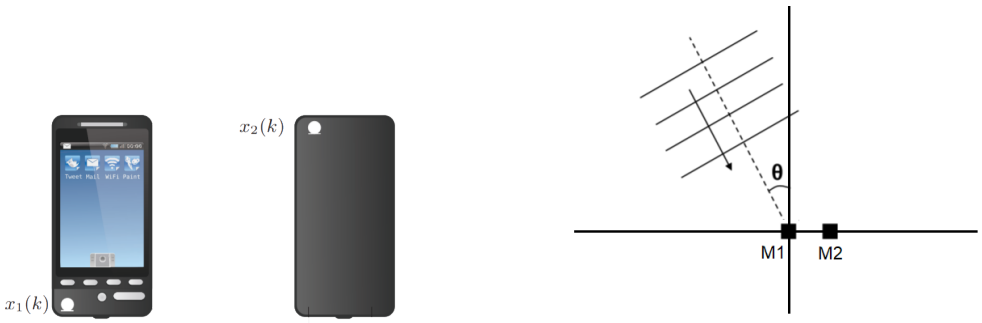
The aims of the project include the following:

- Capturing audio samples containing more than one person talking in real life with different reverberant conditions, and potentially with background noises
- Developing and Experimenting algorithms that can zoom (in audio terms) towards the speaker of interest
- Evaluating and comparing the performances of the algorithms (e.g. in terms of speech quality and intelligibility)

An important point to note here is regarding to the first aim to capture audio examples. Instead of real-life recording using smartphones, a room acoustics simulation environment using anechoic speeches as input is preferred. This would allow a more rapid verification environment for the algorithm, as one can easily modify parameters such as speakers and receivers position, degree of reverberation and size of the room, and generate a new set of test data. While real-life data would be the ultimate goal of the project, the flexibility and rapidness in prototyping leads to the selection of software simulation.

Also, in this project, processing are assumed to be done offline, i.e. after the audio capturing has been completed. This assumption allows the processing technique to not be in real-time, hence more freedom in developing the algorithm, where it can look into the 'future' data, not only the past.

Despite the ultimate aim to apply the acoustic zoom in scenarios containing multiple speakers, the scope of the project would be focusing on the cases involving two speakers, with the vision that the algorithm can be potentially extended to a multi-source environment. Besides, as the initiative of the project is to apply the acoustic zoom algorithm on mobile phones, the microphones set-up should comply the physical constraint of mobile phones. Typically, mobile phones have a dual-omnidirectional-microphone set-up [3], with its dimension being around 15cm x 8cm, as seen from Figure 2a. Hence, the number of omnidirectional microphones is chosen to be two, and the separation has to be confined within the mentioned dimensions.



(a) Example of microphones location in a smartphone. (From [3])

(b) Illustration of direction of arrival of a sound wave

Figure 2: Illustration of the set up adopted in the project

1.2 Project Breakdown and Requirements Capture

To achieve the above aims, the project can be broken down into three main tasks. Below outlines and captures the deliverables for each task.

The first task is about generating reverberant audio data with room acoustics. As mentioned, software simulation is preferred. Hence, the aim of the phase is to generate room impulse response (RIR), where a user can input room information, source and receiver locations, and return the RIR in the specified environment. To verify the simulation results, some baseline cases are also set up for discussion.

The second task, which is the core part of the project, is about the audio zooming algorithms. There are several approach to the zooming algorithms, such as time-frequency (TF) masking, beamforming and machine learning. The deliverable of this task is to establish several candidate zooming algorithms, with evidence like spectrogram proving their effectiveness or ineffectiveness.

The third task would be evaluation of the algorithm performance. After developing the algorithms, the requirement of this stage is to compare the effectiveness of the algorithms. Both subjective and objective tests would be taken into account, which covers measures such as the speech quality, intelligibility and distortion. The requirement of this stage is to provide a thorough analysis and comparison of advantages and disadvantages between different algorithms.

1.3 Report Structure

The report starts with the background research on the problem in **Section 2**, including the literature review on some key technical assumptions, simulation of room acoustics and insights into previous researches on the zooming algorithms and evaluation metrics.

Section 3 then illustrates the simulation environment to capture the audio data. **Section 4** collates the different zooming algorithms developed in the project and

their implementation results, while **Section 5** evaluates and compares their performances using subjective and objective tests.

Section 6 then evaluates the outcome of the project by comparing with the project plan in a management perspective. The report concludes with summary of findings and insights into potential further development of the project, as presented in **Section 7**.

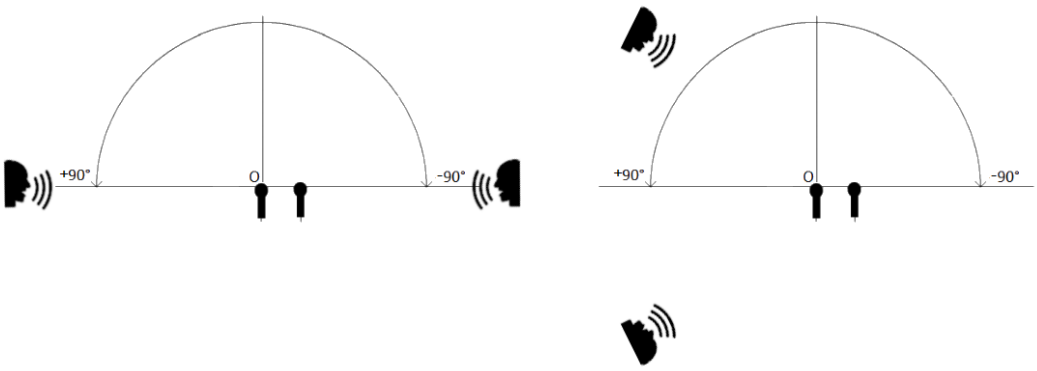
2 Theoretical Background

Despite such an interesting problem to be investigated, there are still some uncertainties to be resolved through undergoing background research. In this section, firstly, some essential technical assumptions for the project would be visited. The algorithm behind the simulation of room acoustics would also be introduced. Some established techniques such as beamforming and Blind-Source Separation (BSS) would also be looked into to assess their applicability in context of two closely-spaced microphones in smartphones. Lastly, the review concludes with introducing a few existing techniques for measuring speech quality and intelligibility.

2.1 Key Technical Assumptions

Before going into details of the core part of the project, there are some key assumptions that makes the project feasible. Below illustrates and explains why they are necessary for the project.

2.1.1 Physical Limitation of Microphone Arrays in Smartphones



(a) Range of resolvable directions

(b) Example with ambiguity on resolving directions

Figure 3: Illustration on the physical limitation of smartphones

First assumption is related to the physical limitation of smartphones. Smartphone commonly possess two microphones. However, in a dual-microphone set-up, in order to avoid ambiguity, the phase difference of arrival between two microphones can at most be used to resolve direction within 180° in a 2-D plane horizontal to the plane of two microphones, as depicted in Figure 3a. Figure 3b shows a case where speakers lie out of the range. Despite the two speakers speaking from different directions, the time difference of arrival is the same. Hence, with only two receivers, due to symmetry, it is not possible to resolve directions in such cases. Extending this deduction to three dimensional case, it can also be assumed that the elevation of the sound sources with respect to the microphone $\phi \approx 0$.

2.1.2 Sparsity Assumption

Considering established processing techniques like BSS, binary TF mask, they rely on the assumption that in time-frequency representations, a bin would only belong to one of the sources, alternatively called as single-source dominance. In other words, it is heavily linked to the assumption which is related to the sparsity of the sound source signal in time-frequency representations. Jourjine [4] introduced the concept of **W-disjoint orthogonality** stating that TF representations of the sources do not overlap. This was defined as

$$S_i^W(\omega,\tau)S_j^W(\omega,\tau) = 0, \forall i \neq j, \forall \omega, \tau. \tag{1}$$

where

$$S_k^W(\omega,\tau) = \mathcal{F}^W[s_k(\cdot)](\omega,\tau) = \int_{-\infty}^{\infty} s_k(t)W(t-\tau)e^{-j\omega t}\,dt \tag{2}$$

is the windowed Fourier transform, or commonly known as short-time Fourier transform (STFT) of signal $s_k(\cdot)$ from source k using windowing function $W(t)$.

However, this assumption is often violated by speech signals. Instead, speech signals exhibit a level of **approximate W-disjoint orthogonality**. Rickard [5] introduced a measure to W-disjoint orthogonality about the percentage of energy of source for time-frequency bins where it dominates the other sources by x dB. Figure 4 shows the result, which indicates that under the presence of two speech signals, they are 80% W-disjoint orthogonal if there are 15 dB of a single source dominance, which means 80% of the signal power can be recovered. This experiment opened up the potential of using time-frequency mask in the project, and will be further justified in Section 2.3.2.

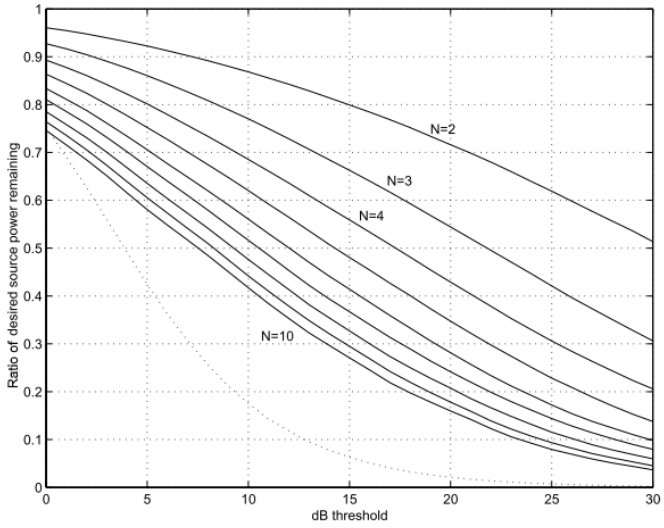


Figure 4: Ratio of remaining desired source energy for time-frequency bins where it dominates the other sources by x dB. (From [5])

2.1.3 Narrowband Assumption

Another assumption introduced by Jourjine [4] is the **narrowband assumption**, which is commonly used in array processing literature. It refers to the physical separation of the sensors is small enough relative to the carrier and bandwidth of the signal such that the relative delay between the sensors can be expressed as a phase shift of the signal, as seen in Equation (3).

$$\mathcal{F}^W[s(\cdot - \delta)](\tau, \omega) = e^{-j\omega\delta}\mathcal{F}^W[s(\cdot)](\tau, \omega) \tag{3}$$

where δ is the time delay corresponding to each path.

This gives rise to the property where the phase difference of each time-frequency bin between two microphones actually refers to the time difference of arrival (TDOA), which then can be used to distinguish whether that time-frequency bin belongs to the speaker of interest from a certain direction, hence enabling the use of binary time-frequency mask in this project.

2.2 Simulating Room Acoustics

2.2.1 Reverberation

To begin with simulating synthetic audio data, firstly, the effect of room properties towards a sound source has to be investigated. As widely known, waves experience reflection when encountering a change of medium. Hence, a sound wave experiences reflection at walls and floors of a room. This leads to the situation where the microphone or listener receives a signal that is the superposition of many delayed and attenuated copies of the original speech, as illustrated in Figure 5. This persistence of sound after the production of sound is named reverberation. A special case is where the walls and floors are purely absorbing. It means that there are no reflections, such that the received signal is purely through the direct path. This situation was defined as the "free field" in acoustics.

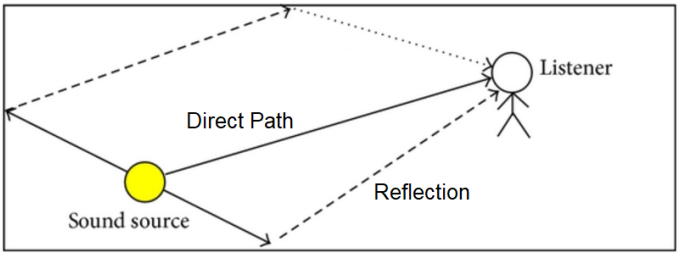


Figure 5: Illustration of room reverberation (After [6])

As the simulation aims to replicate the real world captured speech signals and enable the evaluation of performance of algorithm under different reverberation

condition, the degree of reverberation has to be quantified to allow a fair comparison between simulated and real-life environment. In acoustics, the reverberation is commonly characterised by the two parameters - Reverberation time (RT_{60}) and Direct-To-Reverberant Ratio (DRR).

2.2.1.1 Reverberation time (RT_{60})

RT_{60} was defined as the time taken for the reverberant energy to decay by 60dB once the sound source has been abruptly turned off. This parameter was defined by Sabine in 1898 [7], who discovered that reverberation time is closely related to the amount of absorption of the walls of the room and the room size. The relationship was then formally derived as

$$RT_{60} = 0.1611\frac{V}{A} \tag{4}$$

where V is the room volume in m^3, and A is the total absorption in m^2. With this formal definition, below investigates the physical meaning of each parameter and its impact on the reverberation time.

The first parameter is the total absorption A, which is the sum of products between absorption coefficients α and the corresponding surface area S. In acoustics, absorption refers to the process where sound energy is taken in by the materials when sound waves are encountered, contrary to reflecting. The amount of absorption is then quantified by the absorption coefficient α, which is ratio of absorbed sound intensity in an actual material to the incident sound intensity. Absorption coefficients are heavily dependent on the building material. When $\alpha = 0$, the material is purely reflective and loss-less. This means that the persistence of the sound would be infinitely long, and hence RT_{60} tends to infinity. On the other hand, when $\alpha = 1$, the material is purely absorbing. It means that the only propagation path from sound source to listener is the direct path, which is equivalent to a free field. By definition, RT_{60} is 0 seconds in free field. A point to note is that the absorption coefficients vary with frequencies. From Equation (4), it can be deduced that RT_{60} is also dependent on frequencies. In this report, RT_{60} generally refers to the mean RT_{60} value averaged across frequency bands.

The second parameter is the room volume, or less strictly referred as the room size. From Equation (4), it can be seen that the larger the room is, the longer the RT_{60} is. This is because with a larger room volume, the reflective paths would most likely to be longer, leading to a longer propagation time.

Putting these parameters into context, RT_{60} of a typical office room is around 300 ms, while that of an auditorium or an opera house is around 1.5 to 2.5 seconds [8].

2.2.1.2 Direct-To-Reverberant Ratio (DRR)

However, RT_{60} does not necessarily reflects the amount of reverberation in the received speech, as it is independent of the speaker and microphone locations. For instance, when the direct path is far shorter than the reflective paths, the amount of reverberation would be insignificant to the received speech. This gives rise to the second parameter, DRR. It refers to the ratio of the sound pressure level of a direct sound from a directional source to the reverberant sound pressure level simultaneously incident to the same location. In other words, it provides a measure of the energy from direct path compared to the reflected paths. Hence, contrary to reverberation time, this ratio depends highly on the position of sources and receivers. The closer the source is to the receiver, the higher the DRR, which also means that reverberation has less impact on the received signal. These two parameters together models the acoustic characteristics of a room with a specific source-receiver layout.

2.2.2 Modelling a Room

2.2.2.1 Room as a Linear Time-invariant (LTI) System

After understanding the theory of reverberation with some established quantitative characteristics, the next step is to model a room. Generally, assuming that the variation of temperature is a room is sufficiently slow, rooms can be modelled as passive **linear time-invariant** (LTI) systems.

Denoting the system output due to $x_k(t)$ is $y_k(t)$ for some k, the properties of a LTI system can be defined as

- **Linear**: When the new input is given by $\sum_k c_k\, x_k(t)$, output would be $\sum_k c_k\, y_k(t)$. for any arbitrary constant c_k.
- **Time-invariant**: Applying the same input signal at now or T seconds from now does not have an impact on the output signal, except for a time delay of T seconds. (i.e. the output due to input $x_k(t-T)$ is $y_k(t-T)$)

With these two properties, consider breaking input $x(t)$ down into a continuum of time-shifted impulse functions, the output would then be the corresponding continuum of impulse responses. This leads to the conclusion that the behaviour of a LTI system can be completely defined by its impulse response, where the output of the system $y(t)$ is the convolution of the input to the system $x(t)$ with the system's impulse response $h(t)$, defined as

$$y(t) = x(t) * h(t) = \int_{-\infty}^{\infty} x(t-\tau) \cdot h(\tau)\, \mathrm{d}\tau. \tag{5}$$

In the context of modelling a room, the impulse response $h(t)$ is named as the

Room Impulse Response (RIR), which defines the response of the room from a single source to a single receiver **at specific locations**. Hence, one can perform convolution between anechoic speeches and RIR to obtain the synthetic received speech signal.

2.2.2.2 Simulating Room Impulse Response

Knowing that RIR sufficiently defines a room as an acoustic system and hence allowing the generation of synthetic echoic speech data, the remaining step is to simulate or "generate" the RIR. Currently, there are a few methods including Digital Waveguide Mesh [9], which provides solutions for wave propagation problems, yet computationally expensive. In this project, as the focus is working on the audio zooming algorithm and evaluate its performance under different reverberant conditions, room design such as shapes is not the main subject to be investigated. Instead, for fast and simple stimulation, a "shoebox" rectangular room would be the choice. This allows the simple use of the **Image Method** to simulate room acoustics [10]. Image method is an algorithm to model the reflections by the walls into source images. This is analogous with the situation of placing a mirror in the context of light ray reflection. Hence, for a confined "shoebox" room, there will be infinitely many images of the sound source, as seen from Figure 6. Theoretically, RIR can then by generated by summing all the individual impulse response from the images to receivers. However, taking into account the absorption by the walls, the images are attenuated compared to the original source. The more reflections it experience, the more attenuated the response will be, and eventually becomes negligible. This reduces the image method from an infinite sum to a finite sum, and drastically improves the computational complexity, hence suitable for the project.

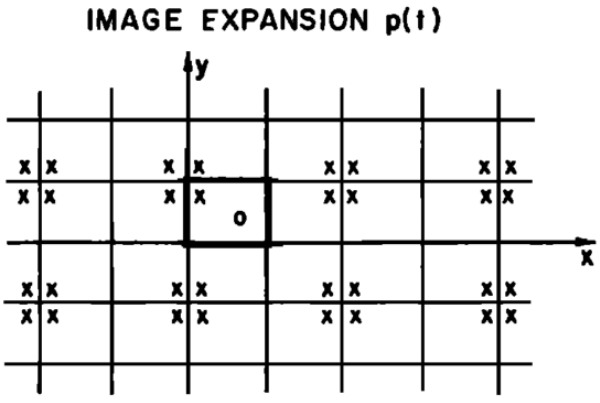


Figure 6: 2-D Bird-eye View of the Image Method (Solid box represent the original room). (From [10])

One interesting point to note is that although the image method achieves an exact, simple and computationally cheap impulse response generation for rectangular

rooms, non-rectangular or irregular room shapes would make the method computationally expensive and therefore undesirable. However, in the context of this project, as the room geometry is not a primary concern, it is assumed that regular rooms would be the environment of concern. Section 3.2.3 will further discuss the choice of established room impulse response generator using image method.

2.3 Processing Algorithms

In this section, several established techniques are analysed and commented on their suitability and challenge to be faced in the context of the project.

2.3.1 Beamforming

As one of the most renowned technique for array processing, beamforming, also known as spatial filtering, is a signal processing technique used in sensor arrays for directional signal transmission or reception. The simplest type of beamformer is a Delay Sum Beamformer (DSB), which is achieved by delaying and summing signals in such a way that signals at particular angles experience constructive interference while others experience destructive interference.

To visualise this algorithm, consider Figure 7, where two element array whose sensors are d m apart, and the desired signal is coming from a DOA of θ. To provide a constructive interference towards θ, a delay δ can be applied to the signal received by Microphone 1 and summing the resultant signal with Microphone 2. The required delay can be calculated by

$$\delta = \frac{d}{c} \sin\theta \tag{6}$$

with c being the speed of sound. An important point to note is that the maximum delay is constrained by the physical separation of microphones $\delta_{\max} = \frac{d}{c}$.

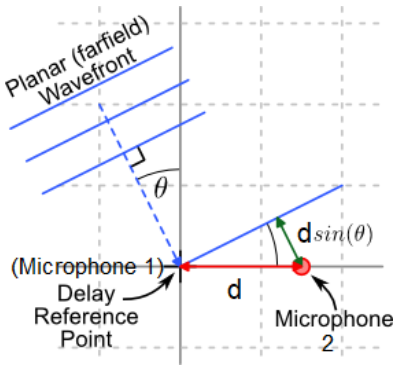


Figure 7: Illustration of delay calculation. (After [11])

This approach, however, has usually been used in audio zooming with a large microphone array. The first reason is that this approach suffers from **spatial**

aliasing. Introduced in [12], when the microphones are too far apart, delay-sum beamforming actually provide constructive interference for signals coming from other angles denoted by $\hat{\theta}$. Below provides a mathematical explanation.

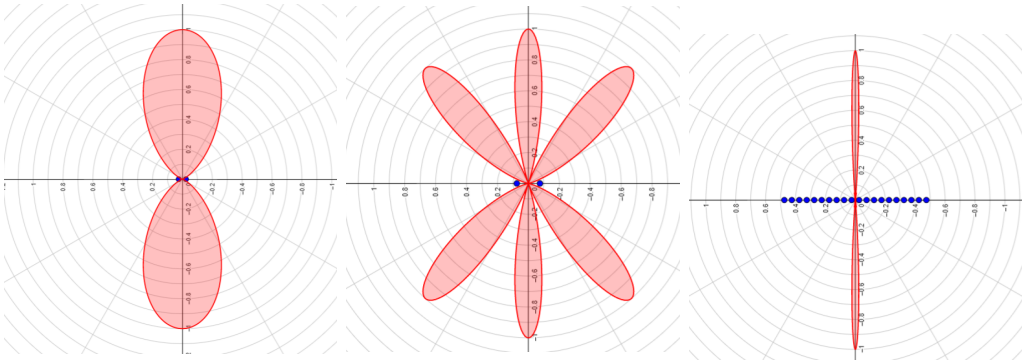
Under the assumption that the wave is stationary, due to the periodicity of a sinusoidal wave, delaying a signal by δ would be the same as delaying a signal by $\delta + nT$, where nT is any integer multiples of the period of the incident wave. When sensors are placed too far apart, i.e. $-\delta_{\max} < \delta + nT < \delta_{\max}$ is satisfied for some n, then the beams would also be directing to directions with bearings $\hat{\theta}$ defined by

$$\delta + nT = \frac{d}{c}\sin\hat{\theta} \implies \hat{\theta} = \arcsin\left(\frac{c}{d}(\delta + nT)\right) = \arcsin\left(\sin\theta + \frac{n\lambda}{d}\right). \tag{7}$$

This evolution of unwanted beams is named spatial aliasing. Note that since $\hat{\theta}$ is a function of wavelength, the unwanted beams occurs at different angles for different frequencies, which makes the suppression of interferences from a certain angle of arrival difficult. In order to have a **unique** beam for constructive interference, where solution of Equation (7) only exists for $n = 0$, one can restrict the separation as

$$T \geq 2\delta_{\max} = \frac{2d}{c} \implies d \leq \frac{\lambda}{2}. \tag{8}$$

This gives the relationship that the separation between microphones should be restricted to be $d \leq \frac{\lambda}{2}$, a similar result to that of sampling theory in frequency domain. An example of spatial aliasing is illustrated in Figure 8, where $d = \frac{3\lambda}{2}$ and $\theta = 0°$. Derived in Equation (7), the actual beam angles with constructive interferences is given by $\hat{\theta} = \arcsin\left(\sin\theta + \frac{2n}{3}\right) = 0°, \pm 41.81°$, which explains the evolution of two extra beams.



(a) 2 receivers, $d = \frac{\lambda}{2}$ (Wide Beamwidth)

(b) 2 receivers, $d = \frac{3\lambda}{2}$ (Spatial Aliasing)

(c) 20 receivers, $d = \frac{\lambda}{2}$

Figure 8: Array Pattern in different setups with $\theta = 0°$. (After [13])

Another problem to adopt beamforming in the project is the **wide angular**

beamwidth. As reflected by [14], the 3dB beamwidth is given approximately by

$$\theta_{3\text{dB}} = 2 \arcsin \frac{\lambda}{dN} \tag{9}$$

which means is approximately inversely proportional to the number of microphones N, as well as the microphone separation d. In other words, the beamformer will become an "allpass" spatial filter when the microphone separation is very small. Together with the half-wavelength condition imposed by spatial aliasing, the best compromise would be $d = \frac{\lambda}{2}$. However, with only two microphones, the main lobe of the beam formed would be having a 3dB beamwidth of around 180°, which still possesses a low spatial selectivity for constructive interference.

Despite a wide beamwidth and spatial aliasing, beamforming under dual-microphone setup still has the potential to achieve an effective interference cancellation. As seen from Figure 8, there still exists a sharp "null" in the case of having two microphones. This gave the potential of sidelobe cancelling beamformer, which performs similarly as a delay-sum beamformer yet with an extra aim to minimize the variance. Contrary to the conventional delay-sum beamformer, this type of beamformer would not only preserve output at the wanted angle, but also attempts to adaptively steer a strong null onto the interfering source. This class of **adaptive constrained beamformer** possesses the potential to track the source of interferences and provides a kinematic model to the interfering source, and would be looked into in Section 4.1.3.

2.3.2 Blind Source Separation using Time-frequency Masks

Another common approach is blind source separation, the separation of a set of source signals from a set of mixed signals, with very little information about the source signals or the mixing process. An estiblished approach is through estimation of a binary TF mask. In [15], it was proposed that under W-disjoint orthogonality assumption, there exists ideal binary time-frequency masks that can separate several anechoic speech signals from one mixture, which motivates this project to explore the usage of binary mask for audio zoom.

The problem remains on the method to **estimate** the ideal binary mask, with the knowledge on direction of arrival of the speech signals, as well as the phase difference of each bin in time-frequency space, which correspond to time difference of arrival under narrowband assumption, the natural approach is to estimate the ideal mask using phase difference of arrival. However, this raw binary mask estimation induces distortion to the speech signals. [16] Upon further research, according to studies carried out in [17], dominant parts of speech signals also form patches and are not randomly scattered across the time-frequency plane. In other words, spectrogram normally exhibit a high correlation between neighbour-

ing time-frequency bins. This gives rise the opportunity of clustering. There are currently a wide variety of research available on time-frequency clustering, including using expectation-maximization (EM) algorithm [18], fuzzy c-means (FCM) clustering [19, 20]. These research makes use of the high correlation between neighbouring time-frequency bins to develop their novel clustering algorithm, providing satisfactory performance is source separation even under reverberant environment, which coincides with the context of the project.

On top of that, there are also other research which involves a combination of techniques such as combining time-frequency mask and mixing matrix estimation. [16]

2.3.3 Subband Estimation of Time Difference of Arrival (TDOA) using Generalized Cross-Correlation with Phase Transform (GCC-PHAT)

An alternate approach would be to estimate directly the TDOA using Generalized Cross-Correlation with Phase Transform (GCC-PHAT) [21]. Instead of typical cross-correlation which operates in time-domain that finds the delay between signals, GCC employs the equivalent expression in the frequency domain, and include an extra factor for general frequency weighting. GCC is defined as

$$R_{12}(\tau) = \int_{-\infty}^{+\infty} \psi(\omega) G_{12}(\omega) e^{j\omega\tau} d\omega \tag{10}$$

where $G_{12}(\omega) = X_1^*(\omega)X_2(\omega)$ is the cross power spectral density function of $x_1(t)$ and $x_2(t)$, and $\psi(\omega)$ represents the frequency weighting.

Now, consider the case where $x_2(t)$ is equivalent to $x_1(t)$ delayed by δ seconds. Using Phase transform as the frequency weighting $\psi(\omega)$ yields

$$\psi(\omega) = \frac{1}{|G_{12}(\omega)|} \implies \psi(\omega)G_{12}(\omega) = \frac{G_{12}(\omega)}{|G_{12}(\omega)|} = \frac{X_1^*(\omega)X_2(\omega)}{|X_1(\omega)||X_2(\omega)|} = e^{j\omega\delta} \tag{11}$$

which means that only the phase difference between two spectra has been retained in the integral. Computing the integral would yield a unit dirac function at $\tau = \delta$, which is the delay of $x_2(t)$ with respect to $x_1(t)$.

However, GCC-PHAT only gives a single delay estimate across all frequencies. In other words, it is only applicable for scenarios where $x_1(t)$ and $x_2(t)$ is of single source. This is undesirable in the context of the project, as the received signal is a mixture of two sources coming from two different directions. From the W-disjoint orthogonality assumption, each frequency bin in each time-frame may be dominated by either of the source, hence meaning that each frequency bin may have different TDOA. To extend this application, the idea of frequency dependent delay estimate and subband processing evolved.

Hence, instead of obtaining a single time delay estimate for all frequencies, the idea of subband processing was introduced in [22], where bandpass filters are applied on the received signal mixture to break it down into subbands, followed by GCC-PHAT to provide the delay estimate for each sub-band. By identifying the frequency-dependent TDOA, each subband within each time frame can be labelled as Speaker 1 or Speaker 2. However, a key question aroused from the research is the way of forming subbands, e.g. the number of subbands. A small number of subbands will lead to a wide frequency range within a subband, which means that each subband might still contain similar amount of influences from two simultaneous speakers, making the delay estimate inaccurate. However, a finer subband will lead to less data available, hence a more erratic TDOA estimation. Therefore, this approach would not be implemented in this report.

2.4 Evaluation Metrics

The problem of audio zooming is highly related to the subjective perceptions of human. Therefore, the most straightforward evaluation metrics would be to conduct listening tests. However, this is usually time-consuming and costly as human listeners are involved. Instead, this could be estimated by some objective measures, which are highly correlated to subjective measures. Below introduces the concept of speech intelligibility and quality, as well as some parameters for performance estimation.

2.4.1 Speech Intelligibility

2.4.1.1 Estimation using Objective Tests

The first question that a speech will be judged on is "How many words could you hear from the speech?", which translates to the intelligibility of the speech. To estimate the speech intelligibility, Taal [23] introduced an Short-time Objective Intelligibility Measure (STOI).

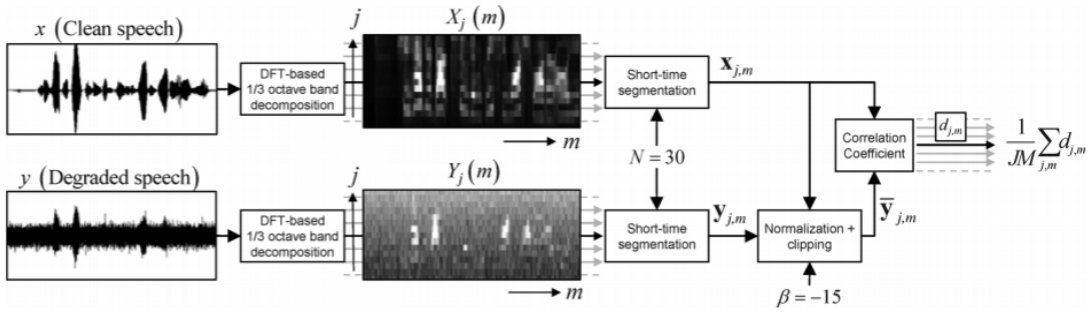


Figure 9: Structure of STOI (From [23])

STOI compares the temporal envelopes of the clean and degraded speech in time segments using the correlation coefficient between them. Collecting the coefficients

and averaging them yields the STOI, which ranges from 0 and 1, where 1 represents a 100% intelligible speech. Results have shown that STOI has a high correlation to actual subjective human rating.

2.4.2 Speech Quality

The second doubt is:"Did you like the speech?". This is purely a subjective opinion by human, yet there exists physical parameters to estimate the subjective rating. Below illustrates the process of estimating using objective measures and conducting a listening test.

2.4.2.1 Estimation using Objective Tests

In a source separation problem, there are three key sources of distortions - background noise, interference from unwanted sources and artifacts such as musical noise. To quantify these effects, Vincent [24] proposes evaluation metrics based on the decomposition of the retrieved signal. Denoting s_j as the retrieved signal of the j-th source, then decomposition is defined as

$$s_j = s_{\text{target}} + e_{\text{noise}} + e_{\text{interf}} + e_{\text{artif}} \tag{12}$$

which is a sum of the target signal and errors due to noise, interfering sources and artifacts. This gives rise to parameters to describe these aspects respectively - Signal-to-Noise Ratio (SNR), Source-to-Interference Ratio (SIR) and Sources-to-Artifacts Ratio (SAR). Finally, collecting all the errors, a metric named Signal-to-Distortion Ratios (SDR) is also proposed for measuring the overall distortion. These are defined as

$$\text{SIR} = 10\log\frac{\|s_{\text{target}}\|^2}{\|e_{\text{interf}}\|^2} \tag{13}$$

$$\text{SNR} = 10\log\frac{\|s_{\text{target}} + e_{\text{interf}}\|^2}{\|e_{\text{noise}}\|^2} \tag{14}$$

$$\text{SAR} = 10\log\frac{\|s_{\text{target}} + e_{\text{interf}} + e_{\text{noise}}\|^2}{\|e_{\text{artif}}\|^2} \tag{15}$$

$$\text{SDR} = 10\log\frac{\|s_{\text{target}}\|^2}{\|e_{\text{noise}} + e_{\text{interf}} + e_{\text{artif}}\|^2} \tag{16}$$

in decibels (dB). These measures the ratio of desired signal energy over unwanted signal energy. The higher the ratios are, the more like that the speech would be rated "good". The MATLAB implementation is available in the BSS_Eval Toolbox [25].

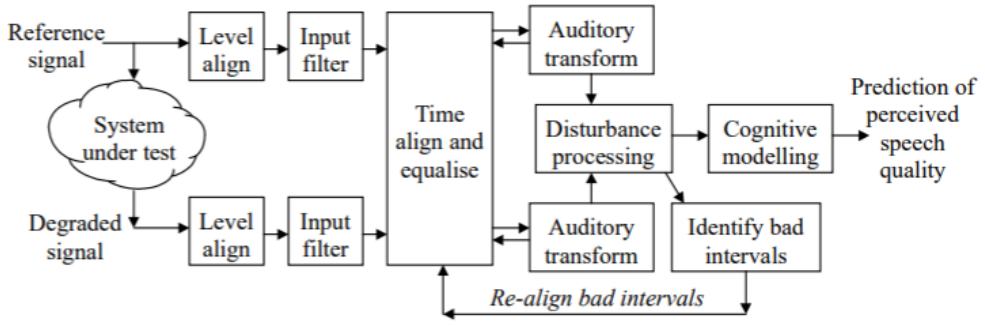


Figure 10: Structure of PESQ (From [26])

Alternatively, Perceptual Evaluation of Speech Quality (PESQ) [26] is a **perceptually motivated** metric. Results in [26] indicate that it gives accurate predictions of subjective quality in a wide variety of conditions, including those with background noise, analogue filtering and variable delay. Besides, often do PESQ get translated into the standardised speech quality expressions, named as Mean Opinion Score - Listening Quality Objective (MOS-LQO) [27, 28]. The score ranges from 1 (lowest perceived quality) to 5 (highest perceived quality), with its meaning defined in Table 1 [29]. The mapping implementation is available in `VOICEBOX` [30] as `pesq2mos.m`.

MOS-LQO	**Label**
5	Excellent
4	Good
3	Fair
2	Poor
1	Bad

Table 1: Absolute Category Rating of MOS-LQO imposed by [29]

2.4.2.2 Measurement using Subjective Tests

There are a lot of listening tests method available to evaluate the speech quality. Multiple Stimuli with Hidden Reference and Anchor (MUSHRA) [31] is one of the common methodology for conducting a listening test to evaluate the perceived speech quality. It adopts a double-blind multi-stimulus test method with hidden reference and hidden anchors, which means that assessors and testers both do not know about the sequence of the speech samples. Reference refers to the perfect speech signal, in this case, the "perfectly separated" speech signal from one of the speakers, while anchors means a standard speech signal with a known performance. In an experiment, the subject is required to give the best-performing audio a score of 100. As the reference is hidden amongst test audios and most likely given the score of 100, this effectively normalise the scores from every subjects, hence yielding a reliable evaluation score, even with a small number of assessors. According to [31], where the conditions of a listening test are tightly controlled

on both the technical and behavioural side, experience has shown that data from no more than 20 assessors are often sufficient for drawing appropriate conclusions from the test.

3 Simulating Audio Data

As mentioned, the first aim of the project is to capture audio data containing multiple speakers, preferably through software simulation. This consists of two steps - capturing anechoic speech data and filtering the anechoic speech with the Room Impulse Response (RIR).

3.1 Capturing Anechoic Audio Data

The first step is capturing the anechoic audio data containing only one speaker. From the database of Speech and Audio Processing Laboratory `sapfs`, databases *ABC Sentences* and *IEEE Sentences* are chosen. Each of the databases contain over 700 anechoic speech at 16kHz by male and female respectively. Each of the recordings comprise short sentences lasting approximately 3 seconds. Besides, all the recordings has been equalized in loudness according to International Telecommunication Union p.56 [32]. These databases provides a sufficient variety of example test speeches to be used in later stages.

3.2 Generating Room Impulse Response (RIR)

As discussed, the simulated audio data can be obtained by performing convolution between anechoic speech and RIR. Hence, the next step is to develop a software to generate RIR.

3.2.1 Requirements

The basic requirements of the software would be the ability to specify the parameters of the room, such as room dimensions, RT_60. The speaker and microphone locations should also be modifiable. Extra features such as modelling the directivity of microphones and human speakers would be preferred as well.

3.2.2 Software Candidates

Complying with the requirements, there are two established software available - *RIR Generator* in AudioLabs by Habets [33] and *MCRoomSim* [34, 35], which both founded on the image method to generate the impulse response. Below compares their features and outlines their respective pros and cons.

RIR Generator is a relatively simple software, where user can specify source and receiver locations, room dimensions and reverberation time, directivity of microphones. The advantage of this software is its simple and user-friendly interface and the ability to specify the reverberation time of the room. However, the software does not have the capability to specify multiple sources at one time.

In comparison, *MCRoomSim* provides a more sophisticated interface, yet a higher degree of freedom for users to modify the room characteristics. Despite the inability to specify RT_{60}, it allows the specification of absorption coefficients at certain

frequencies. This effectively provides a finer control over reverberation of the room as compared with specifying RT_{60}. It also includes routines like 3D room layout plot, RT_{60} against frequency plot, which helps visualise the user input. An extra feature compared to the *RIR Generator* is the modelling of directivity of male and female speakers [36], which provides a better emulation of the real-life environment. Hence, *MCRoomSim* is deemed to be the better choice. One point to note is that *MCRoomSim* requires a minimum sampling frequency of 44.1kHz. Therefore, the sampling frequency is set to be 48kHz, triple of the anechoic audio data. The output RIR would then be downsampled back to 16kHz to match that of the audio data for filtering.

3.2.3 Baseline Testing

To verify and validate if software generates RIR that replicates the real-life settings, baseline cases have to be created. Below provides a comparison between the reality and the simulation using the results obtained from the baseline cases.

3.2.3.1 Reverberation Time (RT_{60})

The first key parameter to be validated is the reverberation time RT_{60}, that is, to check if the RT_{60} of the simulated room adheres the real life results. In *MCRoomSim*, prebuilt room models are provided to define the absorption characteristics of different venues, such as cathedrals, concert halls and office rooms. For validation purpose, the room model of an office room has been chosen, as shown in Table 2.

	Frequency (Hz)					
	125	250	500	1000	2000	4000
Walls	0.30	0.30	0.30	0.30	0.30	0.30
Floor	0.44	0.42	0.40	0.40	0.40	0.40
Ceiling	0.45	0.46	0.47	0.48	0.49	0.50

Table 2: Absorption Coefficients of Predefined Room Model of an Office Room

With the room model established, the next step is to attempt to vary the room size and observe if the reverberation time increases with it. Three different choices are experimented, namely small, medium and large. The room dimensions of each size are designed with reference to the common sizes of office rooms for a fair comparison, as listed in Table 3.

Room Dimension (m)	Width	Length	Height
Small	3	4	3
Medium	6	8	3
Large	9	12	3

Table 3: Dimensions of Various Room Sizes adopted in the Testbench

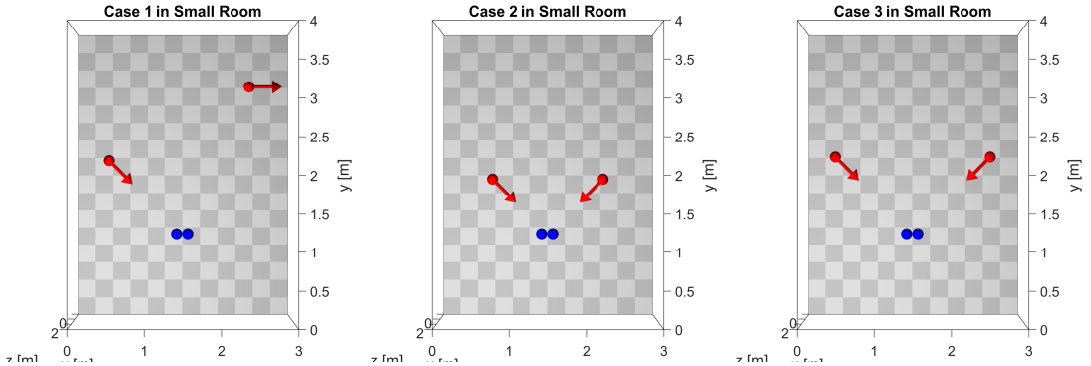
With both room sizes and absorption coefficients now specified, RT_{60} is then calculated using the Sabine's formula stated in Equation (4) to compare against the RT_{60} measured in real life. The results are plotted in Table 4, which shows that the simulated small office room has a RT_{60} of approximately 250 ms. In contrast, in real life, RT_{60} of a small office room in the Department of Electrical and Electronic Engineering Building at Imperial College London is measured to be 320 ms, while a typical RT_{60} of a small office room would be around 300 ms to 400 ms. Comparing both sets of values, despite the slight discrepancy between the measured and simulated value, the result is still deemed acceptable.

	Frequency (Hz)					
RT_{60} (s)	125	250	500	1000	2000	4000
Small	0.249	0.250	0.252	0.250	0.249	0.248
Medium	0.341	0.344	0.346	0.344	0.341	0.339
Large	0.390	0.393	0.396	0.393	0.390	0.386

Table 4: RT_{60} obtained using Predefined Room Model of an Office Room in *MCRoomSim*

3.2.3.2 Speaker-Microphone Layout

With the room acoustics correctly simulated, the remaining doubt is whether the simulation accurately produces RIR for the specified locations of the speakers and microphones. Figure 11 has depicted three baseline scenarios.



(a) Case 1 (b) Case 2 (c) Case 3

Figure 11: Source-Receiver Layout of Different Baseline Cases (Arrow represents the Direction of Speech)

As a convention throughout the project, sources and receivers are numbered from left to right, starting from number 1. Note that speakers are labelled in red, while microphones are labelled in blue. The rationale behind setting up each baseline case is then listed.

- Case 1 represents scenarios where two speakers has an obvious difference in distance to the microphones. This should be reflected by the amplitude difference in the generated RIR.

- Case 2 is a simple symmetrical case. Utilising the symmetry, the generated RIR from speaker 1 to microphone 1 would be expected be the same as the RIR from speaker 2 to microphone 2.

- Case 3 is similar to Case 2, with the only change being the distance of the direct path from the speaker to the microphone. This should be reflected by the decrease in amplitude at the first peak as compared to Case 2.

Below collates and explains some key results obtained from the baseline cases using the acoustic model of a small room. The full results for each of the 3 baseline cases in the 3 choices of rooms are included in Appendix A.

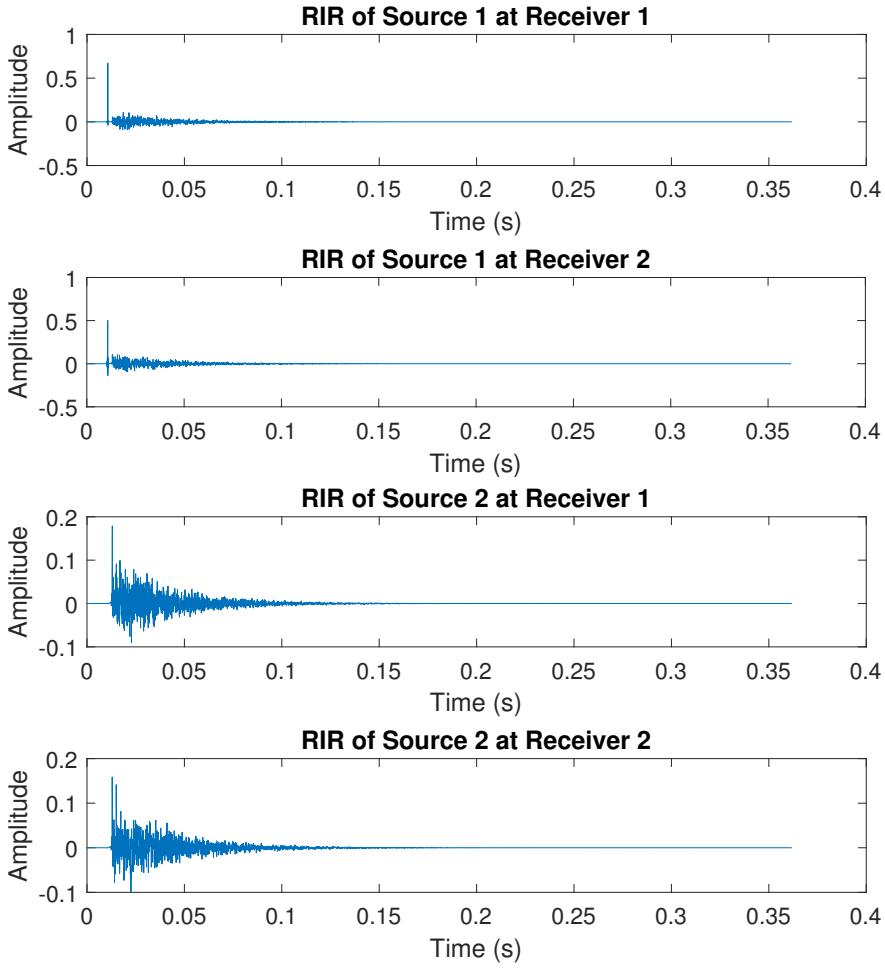


Figure 12: Room Impulse Responses for Case 1

Figure 12 shows the resulting RIR of case 1 in a small room. In the RIR, the earliest peak represents the direct path, while the later peaks corresponds to the reverberation produced by the reflected paths. It can be observed that the earliest peak of the RIR of source 1 is much higher than that of source 2, as expectedly.

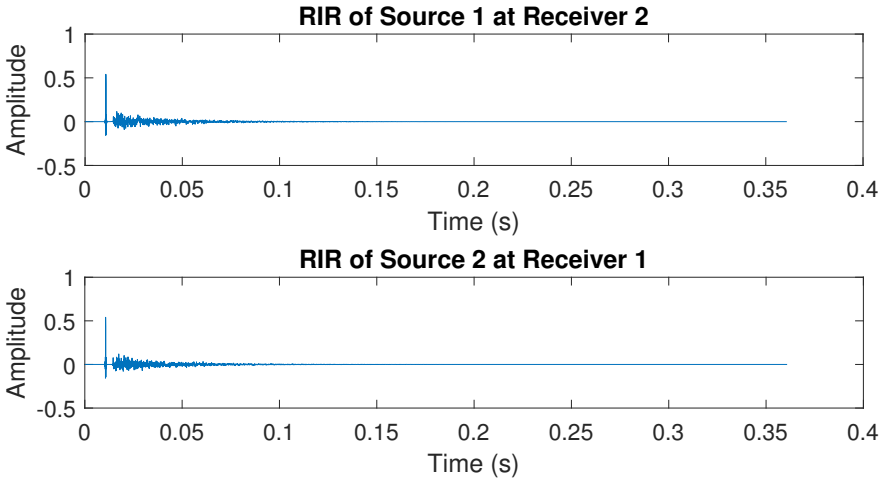


Figure 13: Room Impulse Responses for Case 2

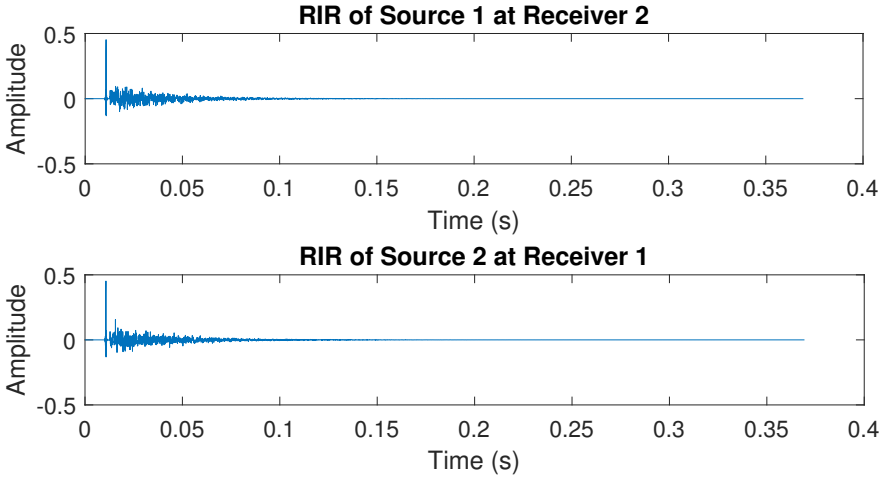


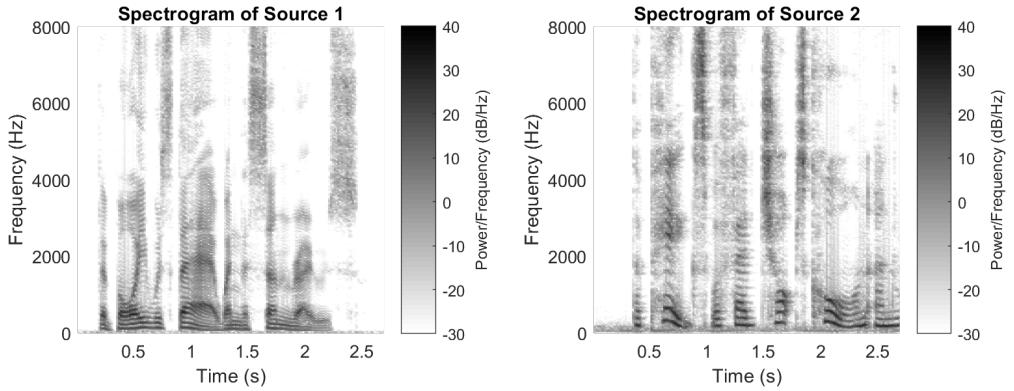
Figure 14: Room Impulse Responses for Case 3

Figure 13 shows the RIRs from case 2. Contrary to case 1, it can be seen that the RIR from Source 1 to Receiver 2 is the same as the RIR from Source 2 to Receiver 1. This verifies the hypothesis where the symmetry of the room leads to the same impulse response. Figure 14 shows the resulting RIRs for case 3. Same as case 2, due to the symmetry of the room layout, the RIR is the same from Source 1 to Receiver 2 and from Source 2 to Receiver 1. The difference between the two cases is once again the direct-path peak. As designed, the distance from source to receiver are made larger in case 3. Hence, it can be observed that the direct path peak is lower as compared to case 2.

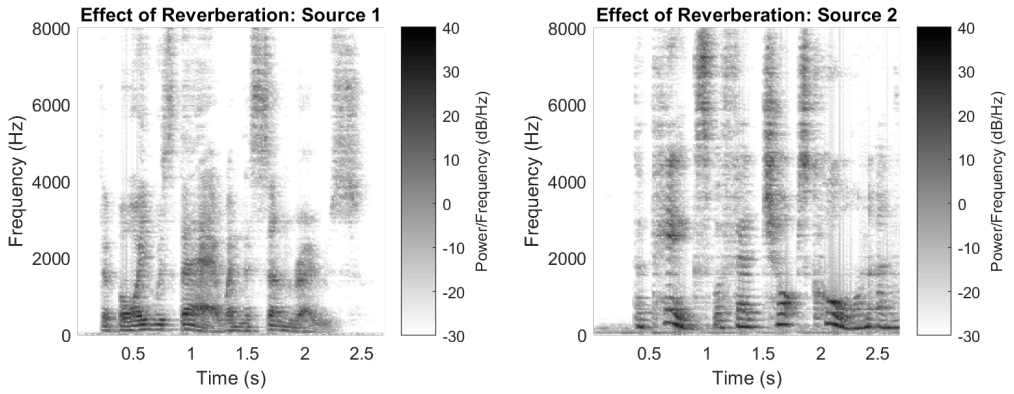
Overall, by comparing different baseline cases, it has been shown that the software generates the correct RIR for specific microphone and speaker locations within a room with an arbitrary size.

3.3 Results

With both anechoic speech data and the software platform for generating RIR available, the final step is to combine them to simulate the synthetic echoic speeches. With the aid of spectrogram, below discusses the effect of the room acoustics towards the anechoic speeches, and displays the results of the received signal using baseline case 2 in a small room.



(a) Original Anechoic Speech



(b) Echoic Speech

Figure 15: Spectrograms of the Anechoic Speech and the Echoic Speech

Figure 15 shows the spectrogram of the anechoic and echoic speech, generated using Hamming window of $N_w = 512$, with an overlapping factor of 2 under a sampling frequency of 16kHz. It can be observed from the spectrogram that the echoic speech gives rise to a more "blurry" spectrogram. This is particularly evident looking at the dark areas, which represents the formant frequencies. In anechoic speech, the formant frequencies can easily be observed, while in echoic speech, they looks submerged by the surroundings. This is mainly due to the reverberation from the reflection of the walls, which leads to a temporal "dispersion" of the dark areas. It hence verifies the software simulation to generate synthetic echoic audio data. Figure 16 shows the resultant echoic mixture received by the microphones.

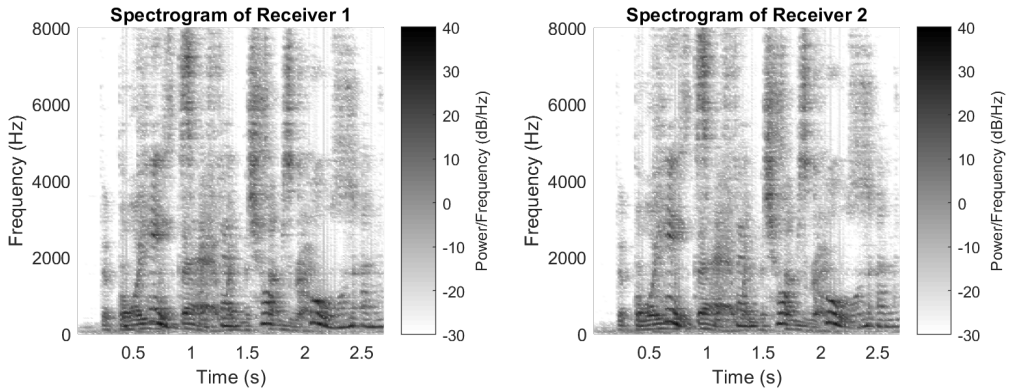


Figure 16: Spectrogram of Received Mixture of Echoic Speech

4 Algorithmic Development

Having built a robust software simulation platform, the next step is to develop the processing algorithm. Using the only information obtained, which is the DOA of the **desired** speaker, below proposes a few approaches, namely beamforming, time-frequency Masking through clustering and machine learning.

4.1 Beamforming

4.1.1 Motivation

As introduced in Section 2.3.1, beamforming is an established audio zooming technique typically used in large microphone arrays. Despite facing with problems such as spatial aliasing and wide beamwidth, it was still worthwhile to observe the effect of beamforming in a dual-microphone set-up, and investigate its effectiveness in the context of the project.

4.1.2 Delay-Sum Beamformer (DSB)

The first basic type of beamformer experimented was the delay sum beamformer. As its name suggests, it was done by simply delaying and summing. The theory is illustrated in Figure 17. It can be seen that for a signal coming from a specific DOA θ, it possesses a unique TDOA δ. By delaying the signal by its corresponding δ, constructive interference would be steered to the desired DOA. In contrast, signal coming from other DOA will be attenuated.

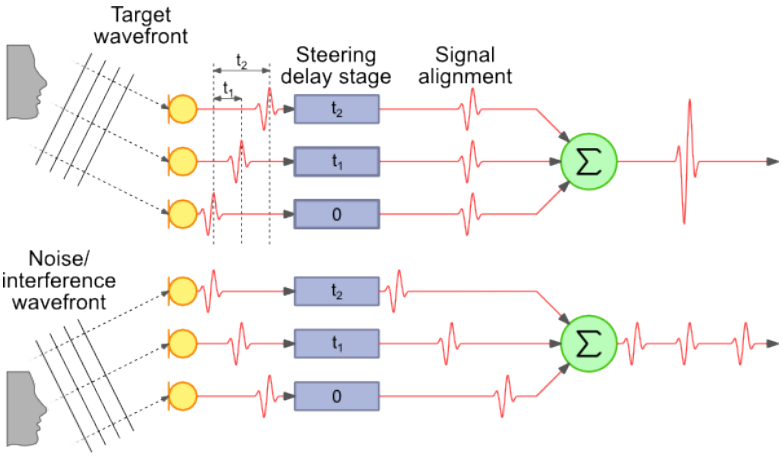


Figure 17: Illustration of theory of delay-sum beamformer formed of 3 microphones. (From [11])

4.1.2.1 Array Pattern

To quantify the performance provided by a beamformer, the gain across different DOA has to be found. The key in deriving the array pattern was to operate in frequency domain.

Firstly, consider Figure 7, to steer the beam towards a DOA of $\hat{\theta}$, the signal received at microphone 1 has to be delayed by $\hat{\delta} = \frac{d}{c}\sin\hat{\theta}$, as derived in Equation (6). In frequency domain, this is equivalent to multiplying an exponential term of $e^{-j\omega\frac{d\sin\hat{\theta}}{c}}$ to the received signal at microphone 1.

Now, consider a signal arriving with an angle θ. As microphone 1 receive the same signal as microphone 2 earlier by $\delta = \frac{d}{c}\sin\theta$. This introduces a factor of $e^{j\omega\frac{d\sin\theta}{c}}$ to the received signal at microphone 1 with respect to that of microphone 2 in frequency domain.

Incorporating two factors, the gain at the microphone 1 can be expressed as

$$H_1(\omega,\theta) = e^{-j\omega\frac{d\sin\hat{\theta}}{c}} e^{j\omega\frac{d\sin\theta}{c}} = e^{\frac{j\omega d(\sin\theta-\sin\hat{\theta})}{c}} \tag{17}$$

where $\hat{\theta}$ is the steering angle of the beamformer, and θ is the actual DOA of the received signal. On the other hand, as microphone 2 is the reference microphone, the gain at microphone 2 would be unity. Hence, the array pattern is given by

$$H(\omega,\theta) = \frac{1}{2}(H_1(\omega,\theta) + H_2(\omega,\theta)) = \frac{1}{2}\left(e^{\frac{j\omega d(\sin\theta-\sin\hat{\theta})}{c}} + 1\right) \tag{18}$$

where the factor of $\frac{1}{2}$ is to normalise the array pattern to have a maximum gain of unity. This deduction has been summarised in Figure 18.

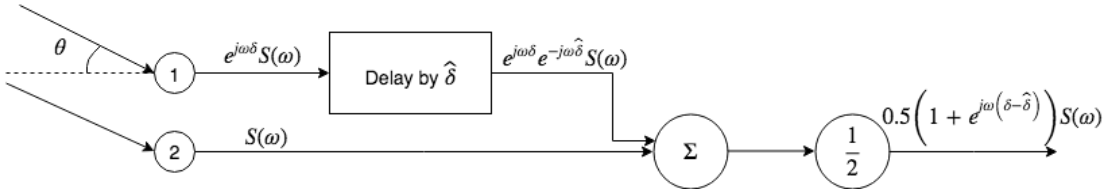


Figure 18: Block Diagram of a dual-microphone beamformer in frequency domain

Observing the expression for array pattern, it can be seen that constructive interference ($|H(\omega,\theta)| = 1$) occurs when $\theta = \hat{\theta}$, which means when the steering angle is the same as the DOA of the received signal.

4.1.2.2 Results

A simple example would be using **baseline case 2** in Figure 11, where the DOA of desired speaker is $\theta_1 = 45°$, and the DOA of the interfering speaker is $\theta_2 = -45°$. To begin with, the microphone separation was set to be 10cm, complying with the usual microphone set-up in mobile phones.

For visualisation, the array pattern are plotted for different frequencies in the voice band, which ranges from 300Hz to 3400Hz [37]. In telephony, the voice band represents the harmonic series of human fundamental frequency that are required

for the speech to be audible. Hence, by observing the array pattern within the voice band, the behaviour of the beamformer towards speech could be known.

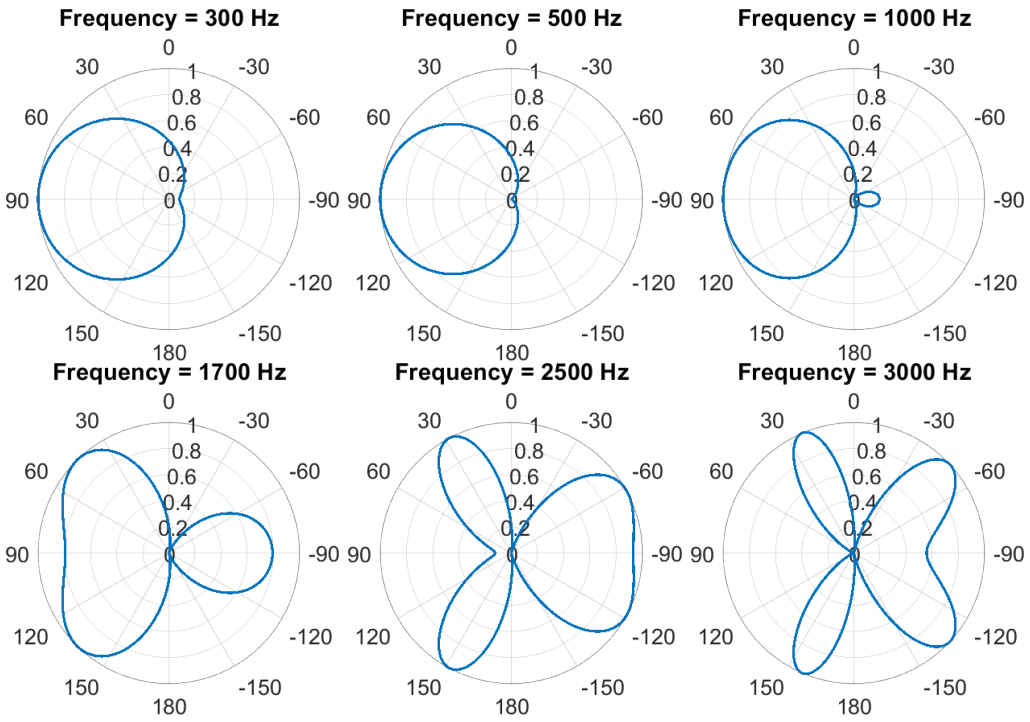


Figure 19: Array Pattern of Delay Sum Beamformer for frequency range 300-3000 Hz

Figure 19 shows the array pattern when the beam angle is 45° and the separation of microphones d is 10cm. From the figure, all array patterns contain unity response at 45°, which verifies the software implementation. Following the discussion in Section 2.3.1, a trade off between spatial selectivity and spatial aliasing has also been shown. At low frequencies, the beam width of the beamformer is very large, due to the fact that the $\frac{d}{\lambda}$ is small, as given in Equation (9). At $f = 1700$Hz, $d = \frac{\lambda}{2}$, which is the maximum separation without spatial aliasing. For higher frequencies, as $d > \frac{\lambda}{2}$, despite a narrow beam width, spatial aliasing was observed, where multiple beams evolved.

This result shows that the choice of the microphone separation directly affects the filtering effect across frequencies. Below attempts to vary the microphone separations and evaluate using the Source-to-Interference Ratio (SIR) under free field and reverberant environment. As the change of microphone position leading to slight differences in the SIR value, to allow a fair ground for comparison, instead of the absolute value, the gain in SIR was instead shown. This was defined as

$$\text{SIR}_{\text{gain}} = \text{SIR}_{\text{zoomed}} - \text{SIR}_{\text{unprocessed}} \tag{19}$$

in dB, which shows the extent of improvement that beamforming performed as an audio zooming technique.

SIR_{gain} (dB)	d =2cm	d =5cm	d =10cm	d =15cm
RT_{60}=0ms	0.4839	0.5981	1.3111	2.3992
RT_{60}=200ms	0.1203	0.5342	1.5438	2.1183
RT_{60}=400ms	0.1310	0.8107	1.2954	1.0766

Table 5: Performance of Delay Sum Beamformer under different degree of reverberation

Table 5 shows the results by varying the separation in range of 2cm to 15cm, fulfilling the dimension restriction of a mobile phone. From the SIRs, it can be seen that the zooming effect is most significant when $d = 15$cm, achieving a gain in SIR of 2.4 dB. Also, the results verified that the degrading of performance of DSB encountering reverberation. This was expected as reflected paths meant that some of the reflected paths from interfering signal may have the same DOA as the direct path of the desired source.

4.1.3 Linear Constraint Minimum Variance (LCMV) Beamformer

However, observing from the beam patterns, the only aim of DSB was to keep the signal of interest. However, in the context of audio zooming, suppression of interfering source is also very important. This gave rise to adaptive beamforming, where the beamformer utilizes the information from the received signal to maximize the received signal power while minimizing interfering noise. One example is the LCMV beamformer, which aims to minimize the total output power of an array subject to some constraints. Consider the case where the constraints are to maintain the signals coming from specific DOAs, then LCMV beamformer effectively becomes an interference canceller. Below formulates the problem mathematically.

4.1.3.1 Problem Formulation

Extending the usage of DSB in frequency domain, beamforming can be done by applying a complex weight to each sensor and summing across all sensor [38].

$$y = \mathbf{w}^H \mathbf{x} \tag{20}$$

where y is the output signal, and $\mathbf{x} = [x_1, ..., x_N]^T$ is the column vector containing the signals received at all N sensors. The output power of the beamformer is then given by

$$P = \mathcal{E}\{y^H y\} = \mathcal{E}\{\mathbf{w}^H \mathbf{x}^H \mathbf{x}\mathbf{w}\} = \mathbf{w}^H R_{xx} \mathbf{w} \tag{21}$$

where $\mathcal{E}\{\cdot\}$ denotes expectation, and $R_{xx} = \mathcal{E}\{\mathbf{x}\mathbf{x}^H\}$ is the covariance matrix of $\mathbf{x}$.

For the constraint, every direction (θ, ϕ) has a corresponding steering vector, $\mathbf{c}(\theta, \phi)$, which performs the same effect as delaying in time domain. The response of the array to the steering vector is given by $\mathbf{c}(\theta, \phi)^H \mathbf{w}$. Therefore, the constraints can now be written as

$$\mathbf{c}(\theta, \phi)^H \mathbf{w} = q \tag{22}$$

where q is the desired response value.

The LCMV problem can then be formulated as

$$\min_{\mathbf{w}} \mathbf{w}^H R_{xx} \mathbf{w} \text{ subject to } C^H \mathbf{w} = \mathbf{q} \tag{23}$$

where C and $\mathbf{q}$ are formed by collating the constraints into columns.

The solution of Equation (23) is then given in [39] as

$$\mathbf{w}_{\text{LCMV}} = R_{xx}^{-1} C (C^H R_{xx}^{-1} C)^{-1} \mathbf{q} \tag{24}$$

An interesting point is that the maximum number of constraints. As seen from Equation (24), solution only exists when $C^H R_{xx}^{-1} C$ is invertible. Hence, it can be seen that the number of constraints should not exceed the number of microphones. This solved a constrained optimisation problem, and allow the beamformer to specify its desired response at a certain angle.

4.1.3.2 Minimum Variance Distortionless Response (MVDR) Beamformer

Minimum Variance Distortionless Response (MVDR) Beamformer is a special type of the LCMV beamformer with only one constraint [38] that the signal coming from a certain angle (θ, ϕ) is distortionless. By distortionless, it means that the desired response $q = 1$. With only one constraint, one could easily rewrite the LCMV solution as

$$\mathbf{w}_{\text{MVDR}} = \frac{R_{xx}^{-1} \mathbf{c}(\theta, \phi)}{\mathbf{c}^H(\theta, \phi) R_{xx}^{-1} \mathbf{c}(\theta, \phi)} \tag{25}$$

This result was particularly useful in this project, as the solution only requires the spatial information of the desired user. Using this solution, the same experiment as DSB was done again with $\theta_1 = 45°, \theta_2 = -45°, d = 10$cm. The constraint imposed was $\mathbf{c}^H(45°, 0°)\mathbf{w} = 1$, and the results are shown in Figure 20.

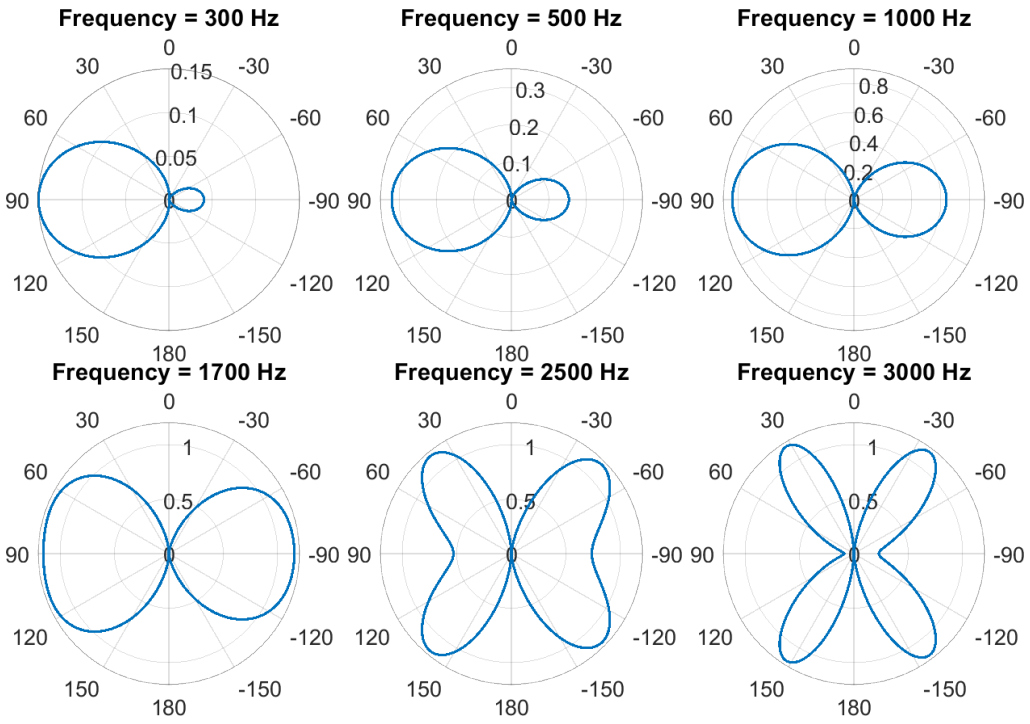


Figure 20: Array Pattern of MVDR Beamformer for frequency range 300-3000 Hz

It can be seen that the array pattern at $\theta_1 = 45°$ is always kept at 1 as expected by the imposed constraint. However, one drawback of the algorithm is that initialisation is required to estimate the noise signal. This could be seen by observing from the pattern without initialisation in Figure 20, where a null was inserted across all frequencies at azimuth of 0° at the wrong DOA. Unfornately, due to time constraint, this problem was not solved. Despite erroneous localisation, MVDR beamformer has demonstrated its power to cancel interferences.

4.1.3.3 Generalised Sidelobe Canceller (GSC)

As a commonly used beamformer, generalized sidelobe canceller (GSC) [40] is another example of an adaptive beamformer. As seen from Figure 21, it consists of a delay-sum beamformer (top branch) and a tapped delay line (bottom branch). The top branch performs the same as a conventional beamformer to provide constructive interference to the wanted signals at a certain DOA. Contrarily, the bottom branch aims to model the interference by passing the signal firstly into the blocking matrix, which is orthogonal and removes the signal of interest, and then filtered by a bank of FIR filters. The adaptive weights of the FIR filters are updated using least mean-square (LMS) algorithm. The difference between two branches would then form the output of the GSC.

Besides, GSC is actually an efficient implementation of a LCMV beamformer. As seen from Equation (24), the solution requires matrix inversion, which is computationally expensive. GSC effectively transformed this constrained problem to an unconstrained problem, which simplifies the implementation. However, as the

scope of the project focuses on the use of small microphone arrays, matrix inversion would be of a small size. Hence, LCMV beamformer was still preferred over GSC for its accuracy in interference suppression. The implementation of GSC has been included in Section B.1.

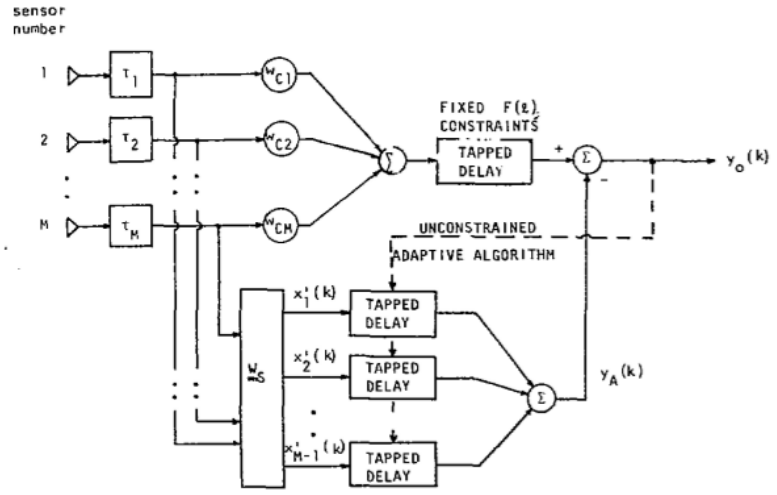


Figure 21: Block Diagram of the Generalised Sidelobe Canceller (From [40])

4.1.4 Summary

In this section, different techniques of beamforming was attempted. Despite demonstrating a weak audio zooming effect, it demonstrated the ability to localise and nullify interferences. Hence, it could be served as an enhancement technique, coupled with other audio zooming techniques to produce a better result.

4.2 Time-frequency Mask Estimation through Clustering of Phase Differences

4.2.1 Motivation

With [4] stating that there exist a perfect binary TF mask to separate speech signals, an estimation technique to form the TF mask is thought to be a way forward. Hence, the first approach attempted is forming TF mask to "extract" the desired speech signal. Throughout the project, it is emphasised that the only information in hand is the DOA of the desired human speaker. The problem now remains to translate this information into parameters that can be obtained in TF domain.

Revisiting the Section 2.3.1 about delay calculation in beamforming, it is shown that signal coming from different DOA θ results in different delays δ between the received signal by the microphones. Using the relationship given by Equation (6), DOA can be translated into TDOA, which effectively transformed a spatial parameter into a temporal parameter.

With the TDOA of the desired speaker now available, the next step is to relate it with the TF representation of the received signals. This can be done using the **narrowband assumption** given by Equation (3), which states that when the separation of the microphones is sufficiently small, time delay δ of the signal can be translated into a phase shift of $-\omega\delta$ in the TF domain. Further to that, from the **W-disjoint orthogonality assumption**, each TF bin will only be occupied by one of the speakers. In other words, the phase difference between received signals at microphone 1 and 2 at each TF bin can be assigned to an unique speaker.

To formally represent this, firstly define the short-time cross power spectral density $G_{12}(\tau, \omega)$ of received signal $x_1(t)$ and $x_2(t)$ as

$$G_{12}(\tau, \omega) = (X_1^W(\tau, \omega))^* X_2^W(\tau, \omega) \tag{26}$$

where $X_k^W(\tau, \omega)$ refers to the STFT of received signal at k-th microphone. The phase difference between received signals at microphone 1 and 2 can then be defined as the phase of the cross power spectral density using

$$\phi_{12}(\tau, \omega) = \angle G_{12}(\tau, \omega). \tag{27}$$

From the above deduction using the **narrowband assumption** and **W-disjoint orthogonality assumption**, ϕ_{12} can be represented by

$$\phi_{12}(\tau, \omega) = -\omega\delta_j \tag{28}$$

where δ_j is TDOA of the speech signal $s_j(t)$ that occupies the TF bin. Therefore, using the phase difference between received signals of microphones, one can effectively decide if the TF bin belongs to the desired speech signal coming from a known DOA. This classification or assignment of membership is called TF masking. By applying this TF mask onto the mixture of signals, the desired speech signal can then be retrieved.

4.2.2 Baseline Testing using Naive Binary Mask

To begin with, the time representation of the signal has to be transformed into a TF representation. Under the help of `VOICEBOX` [30], particularly with the routines `enframe.m`, the signal is split into overlapping frames. In this time-frequency analysis, the simulated audio signal received by the two microphones is first split into time frames using Hamming window of $N_w = 512$, with an overlapping factor of 2 under sampling frequency of 16kHz. In other words, the signals has been split into overlapping time-frame of $t = \frac{512}{16000} = 32$ms. After TF Masking, the TF representation would be transformed back to the time-domain signal using overlap-add method using `overlapadd.m`. As the signal is now represented in discrete time

and discrete frequency, the phase difference is now instead represented by

$$\varphi(k,l) = \phi_{12}(k\tau_0, l\omega_0) \tag{29}$$

where τ_0 and ω_0 are the corresponding time and frequency resolution. In this case, $\tau_0 = \frac{256}{16000} = 16\text{ms}$, and $\omega_0 = 2\pi\frac{16000}{512} = 62.5\pi\text{rad/s}$

To verify the feasibility of the proposed approach, as well as the validity of the derivation above, the first step is to develop a "naive" binary mask using phase difference in **free field**.

A simple example would be again using **baseline case 2** in Figure 11, where the DOA of desired speaker is $\theta_1 = 45°$, and the DOA of the interfering speaker is $\theta_2 = -45°$. With this information, a crude "naive" decision could be made based on the sign of the phase difference as stated below.

$$M(k,l) = \begin{cases} 1, & \text{if } \varphi(k,l) < 0 \\ 0, & \text{otherwise} \end{cases} \tag{30}$$

4.2.2.1 Separation of Microphone d

One problem faced by this approach is the phase wrapping. Revisiting Equation (28), the phase difference can be represented as $-\omega\delta_j$. However, when $|\omega\delta_j| > \pi$, the received phase difference $\phi_{12}(\tau,\omega)$ would be a wrapped version of $-\omega\delta_j$, lying in the principal range of $(-\pi,\pi]$. This wrapping can be defined as

$$\varphi(k,l) = (-l\omega_0\delta_j + \pi) \bmod 2\pi - \pi \tag{31}$$

To avoid this ambiguity, $|l\omega_0\delta_j| < \pi$ has to be satisfied for all TF points (k,l). Hence, the equation can be rewritten as

$$\omega_{\max}\delta_{\max} < \pi \tag{32}$$

where $\omega_{\max}$ is the maximum frequency of interest and $\delta_{\max}$ is the maximum possible TDOA. As discussed in Equation (6), $\delta_{\max} = \frac{d}{c}$. Hence, this relation can again be rewritten as

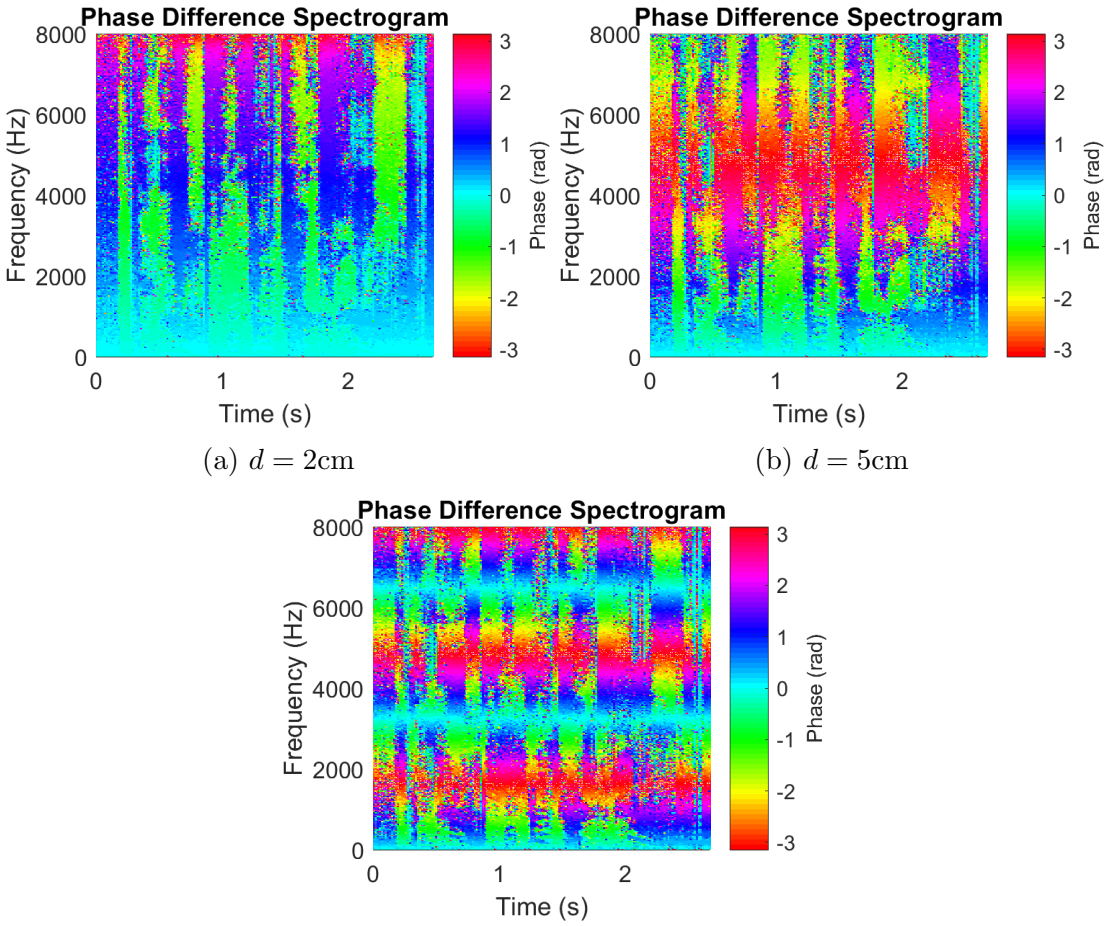
$$d < \frac{c\pi}{\omega_{\max}}. \tag{33}$$

Here, the design choice became to choose $\omega_{\max}$. For a full band implementation, $\omega_{\max} = \pi f_s$, where f_s is the sampling frequency of the speech signal. Therefore, with a sampling frequency of 16kHz, the maximum distance between the microphones would be $d \approx \frac{340}{16000} = 2.125\text{cm}$. This restriction could potentially be further

reduced if $\omega_{\max} < \pi f_s$. In the context of this project, the human voice frequency is considered. In telephony, the voice band ranges from 300Hz to 3400Hz [37]. Using this as a reference, the maximum microphone separation without a significant impairment to the speech signal is $d \approx \frac{340\pi}{2\pi 3400} = 5\text{cm}$.

Figure 22 then visualises the discussion. At $d = 2\text{cm}$, no phase wrapping occurs. Hence, it can be easily observed that the green regions belongs to the desired user. However, at $d = 5\text{cm}$, phase wrapping happens at high frequency regions beyond 3400Hz. When the separation is too large, at $d = 15\text{cm}$, it is obvious that phase wrapping has a significant impact by observing the strong red and cyan horizontal regions in the spectrogram.

In this algorithmic development stage, $d = 2\text{cm}$ is used for a complete evaluation of the performance of the algorithm. However, it is well noted that in smartphones, a separation of 5cm would a more feasible solution to the problem.



(a) $d = 2\text{cm}$ (b) $d = 5\text{cm}$

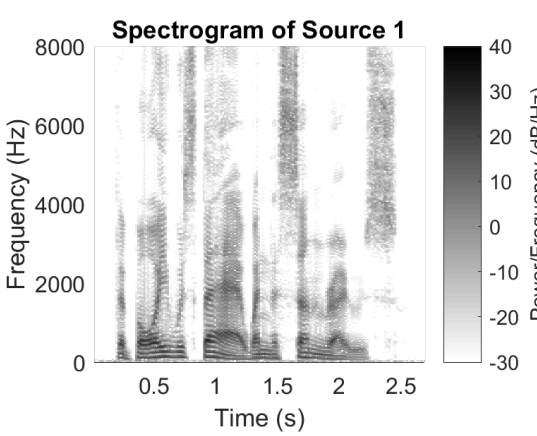
(c) $d = 15\text{cm}$

Figure 22: Phase Difference $\varphi(k, l)$ under different microphone separation

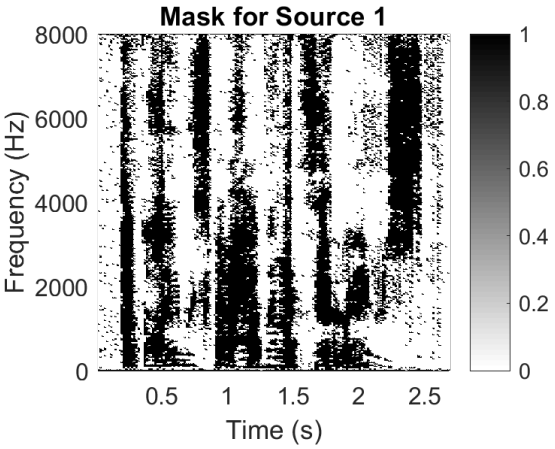
4.2.2.2 Results

Results of the formed binary mask is then shown in Figure 23. It can be observed that the proposed naive TF binary mask successfully labels most TF bins from

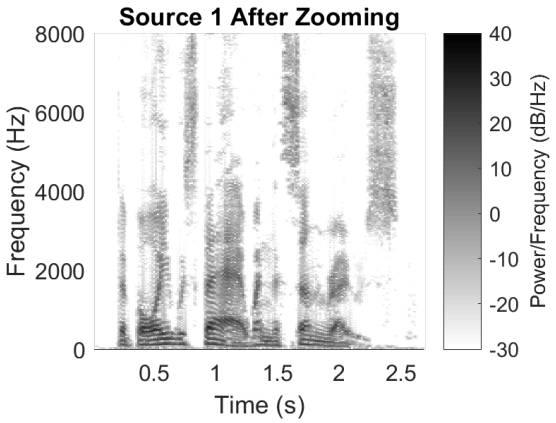
speaker 1 and retrieves the speech signal. It verifies the feasibility of using TF masks to perform audio zooming.



(a) Anechoic Speech



(b) Naive Binary Mask



(c) Zoomed Speech

Figure 23: Evaluating the performance of TF mask in free field

Despite a successful demonstration of theory, the naive binary mask still had some major questions to answer. The major question was that in the development of the binary mask, the DOA of the interfering source was assumed to be known. However, in the context of this project, only the DOA of the desired speaker was assumed to be known. Therefore, it gave rise to the need of localising the interfering sources and getting the DOA of the interfering sources.

4.2.3 k-means Clustering

To achieve that, clustering techniques are introduced. The first clustering technique is an iterative process named k-means clustering [41]. The objective of k-means clustering is to partition the n samples into k sets as to minimize the within-cluster euclidean distances, defined as

$$J_{k\text{-means}} = \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2 \tag{34}$$

where $\mathbf{x}$ is the data point, S_i is the set representing i-th cluster, and $\boldsymbol{\mu}_i$ is the centroid of the i-th cluster. In the context of this report, k refers to the number of speakers in total, which is 2.

4.2.3.1 Data Formulation

Before clustering, a consideration should be made on how to formulate the data in a way that TF bins belonging to the same speaker has a similar data value. Again, the TDOA value could be used. This could be derived from Equation (28) as

$$\hat{\delta}(k,l) = \frac{-\varphi(k,l)}{l\omega_0}. \tag{35}$$

By representing this each TF bin in this form and collating this feature into a vector, the data is now ready for clustering.

4.2.3.2 Forming the clusters

The next step was to form the clusters. k-means clustering adopts an iterative process, consisting of initialisation, assignment and updating.

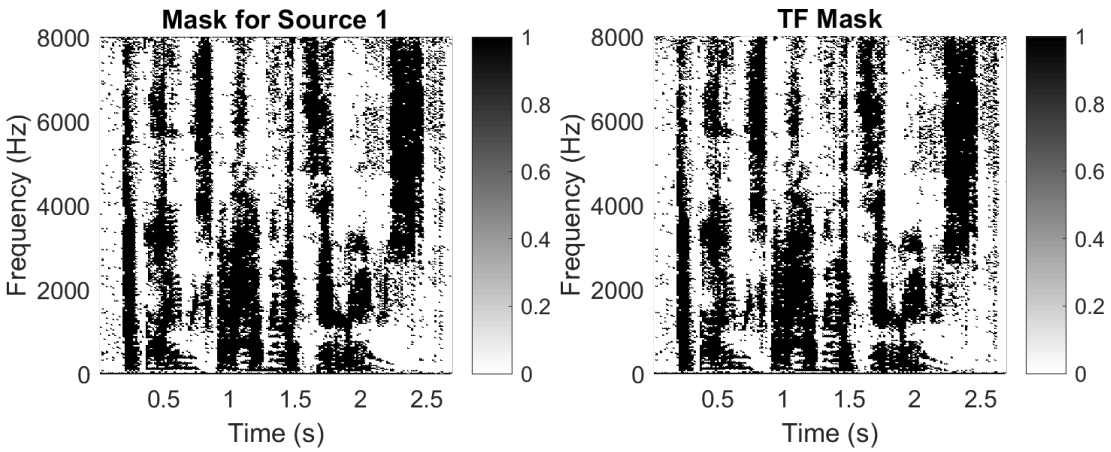
Firstly, the data were partitioned randomly as an initialisation step. Centroids were then computed for each cluster. Comparing the Euclidean distances to each centroid, the data point was then assigned to the nearest cluster. This assignment and update process repeated until the change of objective function stated in Equation (34) was smaller than an arbitrarily small value ϵ, which concludes the clustering process.

4.2.3.3 Cluster Labelling

Forming the clusters, the remaining step was to find out which clusters belong to the desired speaker. This was achieved by again comparing the distance from the known TDOA δ of the desired speaker to that of each centroid and choosing the minimum.

4.2.3.4 Result

With the features formulated and number of clusters decided, the resultant mask is then displayed in Figure 24.



(a) Naive Binary Mask (b) k-means clustering

Figure 24: Comparison of "Naive" Binary Mask and k-means Clustering

Comparing the two binary masks, it can be seen that both methods produced similar results, which shows that k-means clustering successfully localised the interfering source as hypothesised.

4.2.4 Soft mask using Fuzzy c-means Clustering (FCM)

In the development above, the W-disjoint orthogonality assumption was considered true, which means that a TF bin can only be occupied by one source at a time. However, as depicted in Figure 4, the behaviour of speech signals was found to be **approximately** W-disjoint orthogonal. In other words, a TF bin may be occupied simultaneously by multiple sources. This led to the thought of developing a soft mask using fuzzy c-means Clustering (FCM) [42].

Compared with k-means clustering, the main difference is that a data point could simultaneously belong to multiple clusters in FCM. Instead of assigning each point a cluster, a membership index $0 \leq u_n \leq 1$ is given to each TF point to indicate the degree of it belong to the n-th cluster, defined as

$$u_n(\mathbf{x}) = \frac{1}{\sum_{k=1}^{c} \left(\frac{\|\mathbf{x}-\boldsymbol{\mu}_n\|}{\|\mathbf{x}-\boldsymbol{\mu}_k\|} \right)}. \tag{36}$$

It can be seen that, the closer the data point is towards the centroid of the n-th cluster, the higher membership value is given as expected. Then, the formal definition of FCM is defined as an optimisation problem that aims to minimize

$$J_{\text{FCM}} = \sum_{k=1}^{c} \sum_{\forall \mathbf{x}} u_k^2(\mathbf{x}) \left\| \mathbf{x} - \boldsymbol{\mu}_k \right\|^2 \tag{37}$$

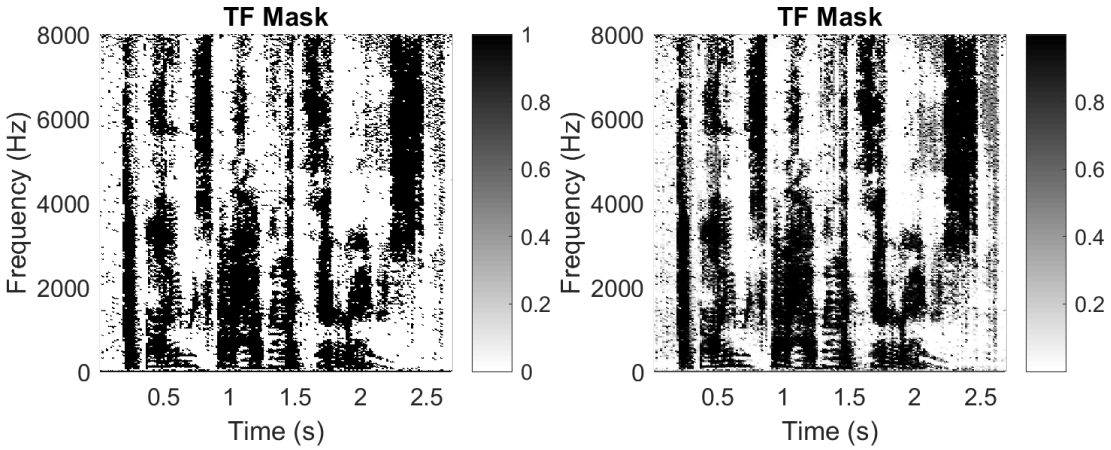
subject to $\sum_{k=1}^{c} u_k(\mathbf{x}) = 1 \, \forall \mathbf{x}$. Again, $\mathbf{x}$ represents the data point, c represents the number of clusters and $\boldsymbol{\mu}_k$ represents the centroid of the k-th cluster. However,

as now a single point can belong to various clusters, the centroid of the cluster is redefined as a weighted mean of all data points given by

$$\boldsymbol{\mu}_n = \frac{\sum_{\forall \mathbf{x}} u_n(\mathbf{x})\mathbf{x}}{\sum_{\forall \mathbf{x}} u_n(\mathbf{x})} \tag{38}$$

Similar to k-means clustering, by setting a threshold on the change of the objective function J_{FCM} and iterates, the optimal clusters would be found.

4.2.4.1 Results



(a) k-means clustering (b) FCM

Figure 25: Comparison of k-means and fuzzy c-means clustering

Experiment was again performed in free field, and the resultant mask is shown in Figure 25. However, by judging on the appearance of the TF mask, no obvious conclusion could be drawn. Hence, using the evaluation metrics discussed in Section 2.4, the performance was compared and listed in Table 6.

Aspect	Metrics	Mixture	Naive	k-means	FCM
Speech Quality	SIR (dB)	-2.0410	21.3404	21.2588	21.2719
	MOS-LQO	1.4601	2.6858	2.6905	**3.1584**
Speech Intelligibility	STOI	0.6104	0.8217	0.8237	0.8356

Table 6: Performance of different clustering techniques in free field

From the objective metrics, it can be seen that the clustering algorithms has successfully achieved the target of improving speech quality and intelligibility. Overall, SIR improved by around 23dB, which was an exciting result. However, focusing on the MOS-LQO score obtained from PESQ evaluation, FCM outperformed k-means clustering by around 0.45. This hence showed that a soft TF mask had the potential to produce high speech quality, hence, further investigation was required.

With the promising result of FCM in free field, the next step is to experiment with reverberant environment. FCM is then implemented in a simulated small room in Table 3. The RT_{60} value was set to be 200ms. The resulting time-frequency mask is then shown in Figure 26.

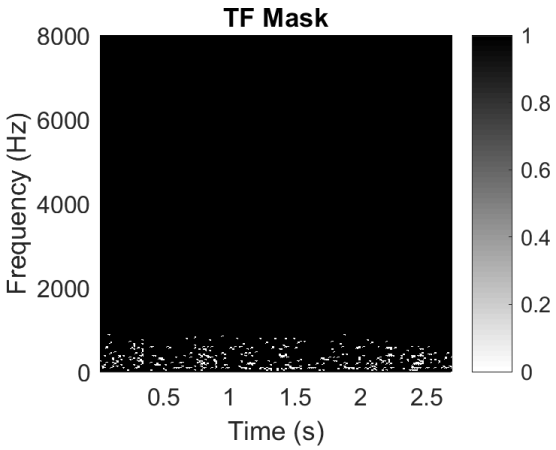
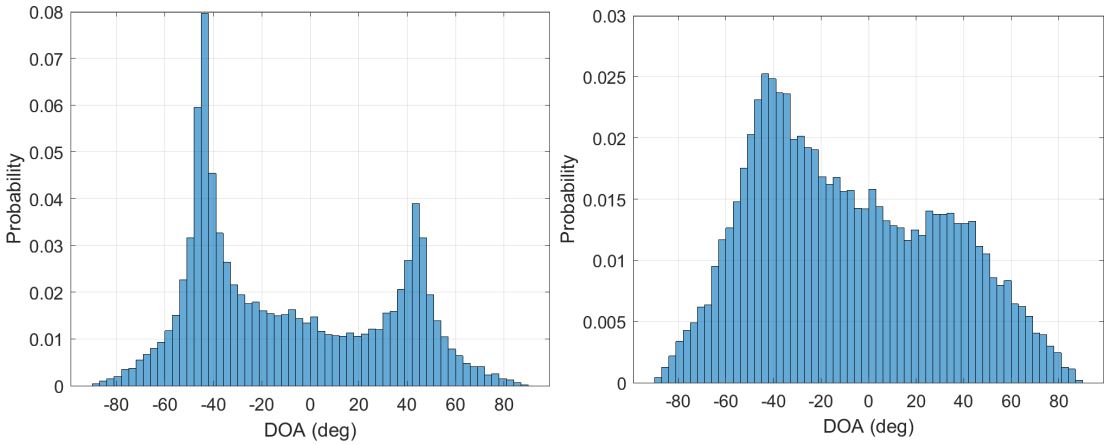


Figure 26: Time-frequency Mask by FCM with $RT_{60} = 200$ms

As seen from the figure, the TF mask was erroneous, which means that FCM was not robust in reverberant environment. To explain this situation, the distribution of DOA φ has been plotted in Figure 27. It can be seen that with no reverberation, there exists two clear peaks at the source DOA. However, when $RT_{60} = 200$ms, the peaks are hardly observed. The reason is that the with reverberation, there exists reflective paths, which leads to different DOA values. This effectively "flattens" the distribution, which made clustering a more difficult task.



(a) $RT_{60} = 0$ms (Free field)

(b) $RT_{60} = 200$ms

Figure 27: Histogram showing distribution of DOA with $\theta_1 = -45°, \theta_2 = 45°$ under different reverberant conditions

4.2.5 Weighted Fuzzy c-means Clustering (wFCM)

With the inability of FCM to deal with reverberation, one of the proposed methods for improvement was to provide a confidence weighting to each TF bin. It meant

that the clustering process would cope more towards those confident TF bins. The new objective function could be written as

$$J_{\text{wFCM}} = \sum_{k=1}^{c} \sum_{\forall \mathbf{x}} u_k^2(\mathbf{x}) w(\mathbf{x}) \left\| \mathbf{x} - \boldsymbol{\mu}_k \right\|^2 . \tag{39}$$

The centroid $\boldsymbol{\mu}_k$ is calculated by

$$\boldsymbol{\mu}_n = \frac{\sum_{\forall \mathbf{x}} u_n(\mathbf{x}) w(\mathbf{x}) \mathbf{x}}{\sum_{\forall \mathbf{x}} u_n(\mathbf{x}) w(\mathbf{x})}. \tag{40}$$

The update of $u_n(\mathbf{x})$ is the same as that of FCM, as shown in Equation (38). Note that when $w(\mathbf{x}) = 1 \, \forall \mathbf{x}$, wFCM defaults to FCM. Below investigates the candidate of weighting factor.

4.2.5.1 Variance

One of the weighting measure was introduced by [43], which assumed that TF regions with low fluctuations of $\hat{\delta}(k, l)$ are not affected by reflections, possess high SNR, and only belongs to one source. To quantify this fluctuations, local variance was introduced as

$$\sigma^2(k, l) = \frac{1}{|N| - 1} \sum_{\forall (k', l') \in N} [\hat{\delta}(k', l') - \mu_{\hat{\delta}}(k, l)]^2 \tag{41}$$

where $\mu_{\hat{\delta}}(k, l)$ is the local mean defined by

$$\mu_{\hat{\delta}}(k, l) = \frac{1}{|N|} \sum_{\forall (k', l') \in N} \hat{\delta}(k', l'). \tag{42}$$

Here, N represents the time-frequency window selected. Using this result, the weight at TF bin (k, l) can be written as

$$w(k, l) = 1 + \frac{1}{\max(\sigma^2(k, l), \kappa)} \tag{43}$$

where κ controls the upper limit of the weight. From this relationship, it can be seen that when the variance is low, the weights assigned would be high, indicating the TF bin is most likely reliable. As suggested by [43], the value of κ was set to be 10^{-3}, and N was set to be a 11-point window of neighbouring TF bins.

4.2.5.2 Signal-to-Noise Ratio (SNR)

An alternative measure would be using the Signal-to-Noise Ratio at each TF bin. This was proposed with the rationale that, when the SNR is high at a particular TF bin, it would be more likely for the TF bin to provide a reliable phase value.

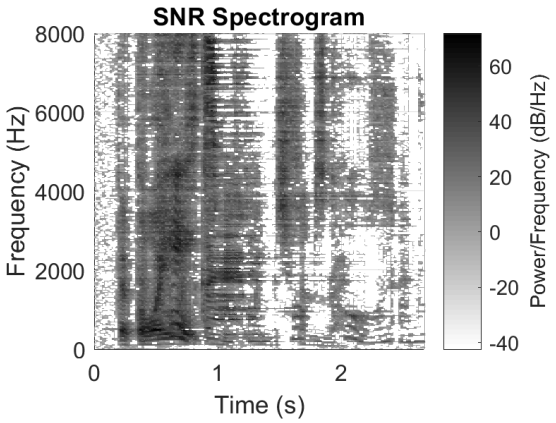


Figure 28: SNR at each Time-frequency point

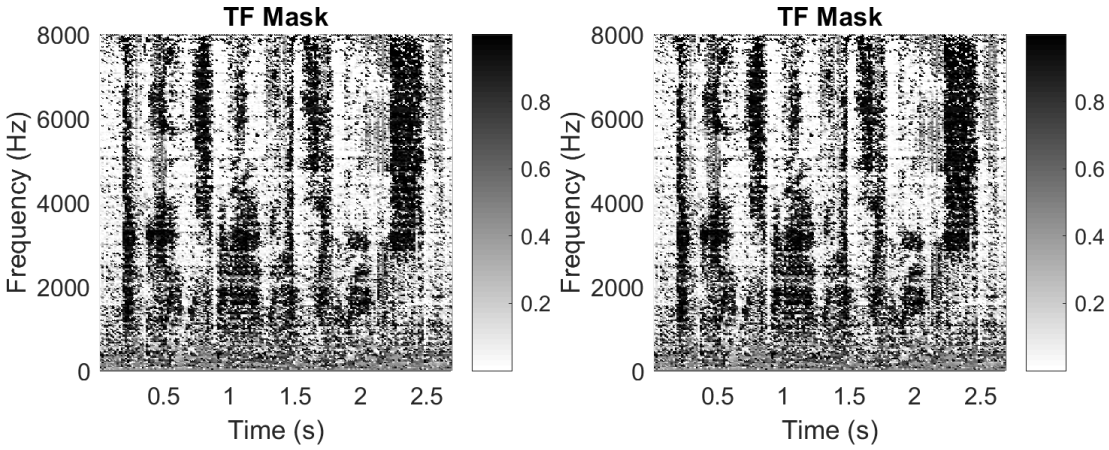
The weights are then give by

$$w(k,l) = \min\{\mathrm{SNR}(k,l), \kappa\} \tag{44}$$

where κ again controls the upper limit. In this report, the value used is $\kappa =$ 100dB.

4.2.5.3 Results

Using both candidate weighting scheme to form the weights, the results are then shown below in Figure 29.



(a) Variance weighted (b) SNR weighted

Figure 29: Time-frequency Mask generated using wFCM

As there was no obvious difference in the TF mask, the performance is then evaluated in Table 7. This shows that both of the proposed weightings provided similar performance, while SNR provides a slightly better interference suppression.

Aspect	Metrics	Mixture	Variance	SNR
Speech Quality	SIR (dB)	-2.2494	0.8102	**0.8923**
	MOS-LQO	1.4260	1.5175	1.5214
Speech Intelligibility	STOI	0.5728	0.6176	0.6198

Table 7: Performance of wFCM in $\mathrm{RT}_{60} = 200\mathrm{ms}$

4.2.6 Weighted Contextual Fuzzy c-means Clustering (wCFCM)

However, comparing Table 6 and Table 7, there still existed a discrepancy in performance in environments with and without reverberation. Therefore, this approach was introduced in [20] as a further improvement in consideration of reverberant environment. Previously, clustering had been done without consideration of how speech signals would appear in clusters in TF representations. As seen from the spectrograms, there exists a strong correlation between neighbouring time-frequency bins. It can be verified by the studies in [17], which states that the dominant parts of speech signals form patches and are not randomly scattered across the time-frequency plane. Instead, speech sounds should display a smooth and continuous appearance.

To include this in the clustering problem, a regularisation term has been added to the cost function so that TF points from neighbourhood will likely to have similar membership values. Firstly, define

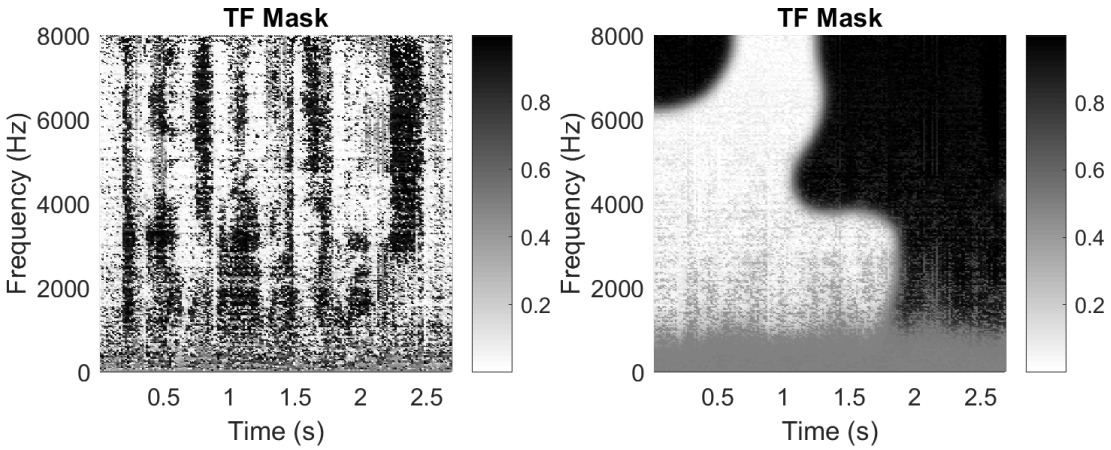
$$C_n(k,l) = \sum_{\forall(k',l')\in N} \sum_{\forall n'\neq n}^{c} u_{n'}(k',l')^2. \tag{45}$$

This is a measure of the degree to how the neighbourhood "belongs" to any cluster other than the n-th cluster. The overall objective function can be written as

$$J_{\mathrm{wCFCM}} = J_{\mathrm{wFCM}} + \frac{\beta}{2} \sum_{n=1}^{c} \sum_{\forall(k,l)} u_n(k,l)^2 C_n(k,l). \tag{46}$$

To minimize this added regularisation term, $C_n(k,l)$ had to be small when $u_n(k,l)$ is large. Effectively, when the membership values of a TF point to a cluster is high, the term attempts to force the neighbourhood to be in the same cluster. Here, β is a trade-off parameter between minimising the objective of wFCM and ensuring the homogeneity of the TF mask, which would usually be determined from cross-validation.

4.2.6.1 Results



(a) Good Initialisation (b) Erroneous Initialisation

Figure 30: Time-frequency Mask generated using wCFCM

Figure 30 shows the result. After experiments, it was found that the algorithm heavily dependent on the initialising partition. With a good initialisation, as seen from Figure 30a, the TF mask successfully capture the TF bins of the speech from desired speaker. However, when the initialisation partition is not as expected in Figure 30b, clustering produces erroneous result. Due to its over-sensitivity towards initialisation of the clusters, wCFCM was not used in the project.

4.2.7 Limitations

Overall, developing TF maskings through clustering showed promising results, however, there were a few limitations and challenges to be solved with this approach using clustering. Below outlines them and proposes corresponding solutions.

4.2.7.1 Estimating the Number of Speakers

In the context of the project, it was assumed that the number of speakers was two, which enables a simple and neat result presentation in the report. However, in real life, the information about the number of speakers or number or clusters is not known from the smartphones. Yet, clustering algorithms introduced requires the knowledge of the number of clusters.

This gives rise to the class of problems named as cluster validity, which is equivalent to the question about if the clusters built are valid. A brute force way to tackle this problem would be to iterate through a range of number of clusters and evaluate the cluster validity using the metrics such as partition coefficients and partition entropy discussed in [44]. In the context of audio zooming or source separation, an alternative method based on the histogram of DOA was proposed

in [45]. As seen from Figure 27, the number of peaks of DOA histogram can provide an estimate on the number of sources. Making use of this characteristics, by selecting the **reliable** TF bins and finding the DOA distribution, an estimation can be acquired.

The **reliability** of the DOA at a TF bin can be defined using the following

- The value of phase difference $\varphi(k, l)$: A relatively large phase difference would be more reliable, as it is less vulnerable to noise. As shown in Equation (28), the phase difference is the product between frequency and TDOA. This means that low-frequency regions tends to have a lower reliability

- Dominance of a single speaker: This originates again from Equation (28) derived from the W-disjoint orthogonality assumption. With the assumption violated at TF bins in reverberant environment, the phase difference does not reflect the TDOA of a speaker, hence unreliable.

Using these two criteria, the DOA distributions at the **reliable** TF bins can be found, hence estimating the number of sources. Despite not being implemented in the project due to time constraint, this method has shown its potential to be coupled with the clustering algorithm.

4.2.7.2 Vulnerability to Initialisation

Another problem faced by clustering was its inherited vulnerability to the erroneous initial partition. As discussed by [46], the convergence speed of clustering algorithms heavily depends on the initialisation. Besides, with a bad initialisation partition, there would also be chances that the clusters converge to a local minimum, yet not the global minimum, as shown in the case of wCFCM in Figure 30.

4.2.8 Summary

In this section, audio zooming by developing TF mask using the phase difference at every TF bin was introduced. With reference to past research and literatures, the approach of clustering was further developed and implemented. The performance of the clustering algorithms was encouraging in anechoic conditions, with the binary mask successfully capturing the TF bins from the desired speaker. On the other hand, the performance deteriorated under reverberant conditions, due to the reflected images of the desired speaker.

4.3 Time-frequency Mask Estimation through Machine Learning

4.3.1 Motivation

Promising signs of using TF mask has been shown in Section 4.2. However, the performance dropped when reverberant environment is introduced, where phase difference became a "weak" feature due to the reflected propagation paths of the room and the violation of W-disjoint orthogonality. Besides, in the project, with the availability of a software platform to simulate reverberant speeches, large amount of training data could be generated. This led to the idea of machine learning, which has the ability to exploit "weak" features using a large amount of data. Due to the time constraint of the project, a model was not built. Nonetheless, below attempts to provide an insight into the potential of using machine learning in estimating TF mask.

4.3.2 Neural Networks

In terms of speech processing, one of the most common machine learning techniques is neural network. The structure of a neural network is shown in Figure 31.

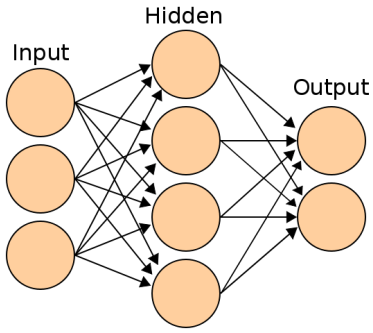


Figure 31: Structure of Neural Network

The structure was designed to copy the behaviour of a human brain, hence, each circle was named as an artificial neurone. In the simplest linear case, each neurone accepts multiple inputs, and produces output by doing a weighted sum. By passing in the training data sequentially and comparing the output from the neural network to the true output, errors are then propagated back through the system, which would then adjust the weights at each neurone. This allows a neural network to approximate continuous functions.

In recent years, the concept of deep learning has been the core of research in neural networks [47]. This gave rise to the evolution of Deep Neural Network (DNN), where only difference is that DNN possesses multiple hidden layers. With the increase in depth of the neural network, one of the most encouraging achievement

by DNN is feature learning [48]. In Section 4.2, under reverberant environment, phase difference or TDOA became a "weak" or less discriminative feature. This would cause problems in clustering, as clustering does not have the ability to develop a discriminative feature out of numerous weakly discriminative features. Yet, with the use of back propagation and the existence of multiple hidden layers, it provides freedom for the system to learn a new and discriminative feature.

4.3.3 Features Extraction

Another important step in performing classification in machine learning is the input features. Below collates different type of features and explains their importance in the audio zooming problem.

4.3.3.1 Monaural Features

As its name suggests, monaural refers to the situation where only single channel data is available. Hence, the information obtained would be limited, and would not include any spatial information. The most straight forward feature would be the short-time magnitude spectrum of the signal. This features provides information about the frequency variations of the speakers. Being studied and used in [49], a demonstration of separation results can be viewed at `https://sites.google.com/site/deeplearningsourceseparation/`, which has shown a satisfactory result of source separation using the magnitude of STFT as its only input to the DNN.

4.3.3.2 Binaural Features

In the context of the project, two microphones are available. Hence, information could be utilised to further improve the monaural zooming.

As described in [50], human ears are sensitive to two primary features for source localisation. The first one, which was discussed in Section 4.2, is the inter-aural time differences (ITD), or a similar expression, the TDOA of the signals. The second feature is the interaural level difference (ILD), defined by the energy ratio of the received signal at the left ear to that of right ear at a specific TF bin. [51] This aroused due to the fact that human ears are are shadowed by heads. Therefore, if a sound wave is coming from the left, the right ear would receive an significantly attenuated version.

In the context of the project, level differences may also be applicable in smartphones in real life. Despite being omnidirectional, the microphones are also shadowed by the body of mobile phone, creating an observable level difference across two microphones. Hence, utilising the similarities or capturing audio using smartphones and human ears, two features are introduced to provide spatial information about the speaker.

4.3.4 Proposed Framework

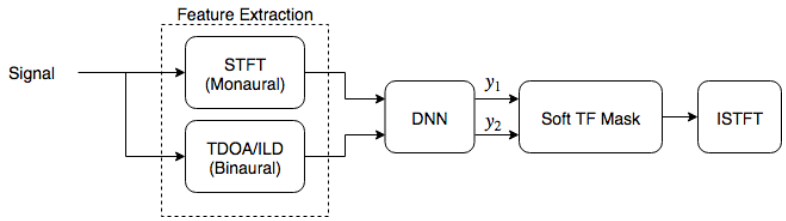


Figure 32: Proposed Framework

Figure 32 illustrates the proposed framework of this TF mask estimation approach. Firstly, the signals received by both microphones are broken down into frames. For monaural features, the magnitude spectrum would be used. On the other hand,

By feeding the features into DNN to train, the output y_1 and y_2 would be the output predictions of whether the particular frequency bin in that frame belongs to speaker 1 or speaker 2. Using this information, one could simply build a hard TF mask by selecting the maximum out of y_1 and y_2, or a soft TF mask as

$$M_{\text{soft},1}(k,l) = \frac{y_1(k,l)}{y_1(k,l) + y_2(k,l)}. \tag{47}$$

The framework then concludes with the inverse STFT to retrieve the signal of the speaker of interest. Despite not being tested due to time constraint, the proposed framework utilizes most information from received signals of two microphones, and has the potential to achieve audio zooming even in reverberant environment.

5 Evaluation

With the conclusion of algorithmic development, the final step was to compare their merits and drawbacks of each approach quantitatively. Particularly, the relationship between the performance of the algorithms and the amount of reverberation would be investigated. In the sections below, speech quality and intelligibility are estimated under different reverberant environments and analysed.

5.1 Estimating Speech Intelligibility using Objective Test

To estimate the speech intelligibility, STOI was used as an indicator. The score ranges from 0 to 1, where 1 represents perfect intelligibility i.e. all the words can be detected accurately. The results are shown in Figure 33.

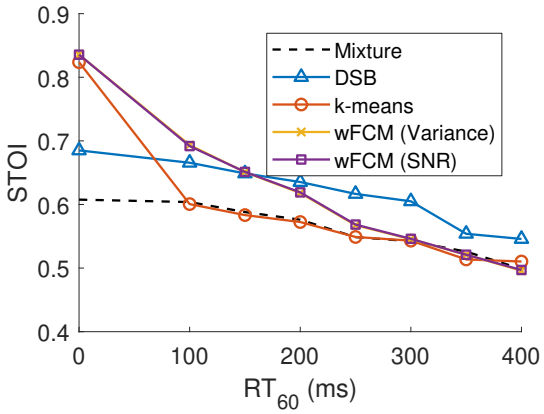


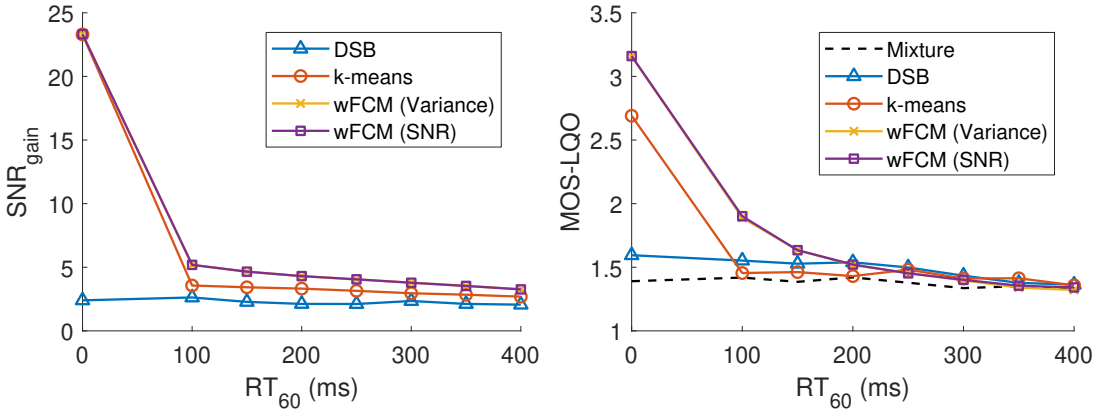
Figure 33: STOI values obtained with different algorithms

From Figure 33, a few observations can be made. The first one is that all algorithms had a decreasing intelligibility with increasing reverberation time, which was expected as echo would make speeches less intelligible. In free-field, TF masking possesses a high STOI value of around 0.85, while DSB stays at around 0.7. However, for a larger RT_{60}, all algorithms for TF mask have a STOI value of around 0.5, while DSB still maintain at around 0.6. One hypothesised reason was the linearity of beamforming and the non-linearity of TF masking. Despite introducing weighting in wFCM clustering, TF mask estimation does not necessarily result in a "continuous" mask. This may lead to artifacts like musical noise that directly affect the intelligibility of the zoomed signal.

5.2 Estimating Speech Quality using Objective Tests

To estimate the speech quality, two parameters are used. The first one is the Source-to-Interference Ratio (SIR), which was computed using BSS_Eval Toolbox [25]. The SIR gain defined in Equation (19) captures the difference in ratio before

and after audio zooming, and indicate how much interference were successfully suppressed. The second measure is Mean Opinion Score - Listening Quality Objective (MOS-LQO). It provides a score from 1 (Bad) to 5 (Excellent) for listening quality, with reference to Table 1. Here, the scores were obtained by mapping from Perceptual Evaluation of Speech Quality (PESQ) using `pesq2mos.m` from `VOICEBOX`.



(a) SIR_{gain} (b) MOS-LQO

Figure 34: Assessing Speech Quality with varying reverberation time

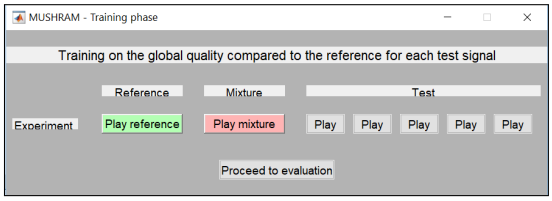
Both graphs in Figure 34 has shown similar trend that TF mask performed well with little or none reverberation, achieving a SIR_{gain} of 23dB and a MOS-LQO of 3.2. However, with the effect of reverberation, the performance significantly has degraded. Therefore, it remains an important task to develop an audio zooming algorithm to combat with highly reverberant environments.

5.3 Measuring Speech Quality through Subjective Listening Test

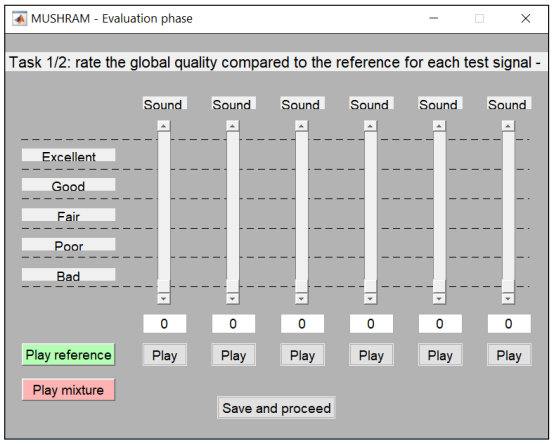
To support the estimation from objective quantitative metrics, an **informal** listening test has been carried out. As mentioned in Section 2.4.2.2, one of most commonly used method is Multiple Stimuli with Hidden Reference and Anchor (MUSHRA). Below illustrates the method of conducting the test. It was well noted that the test was not served for any statistical significance. Instead, the response obtained would only be served as a reference to support the analysis of objective metrics.

The test was conducted using the Graphical user interface in MATLAB, provided by Perceptual Evaluation methods for Audio Source Separation (PEASS) Listening Test Toolkit [52]. The test consists of a training phase and an evaluation phase. The training phase was designed to instruct about the procedures of the test and allow them to adjust the volume to a comfortable level. The evaluation phase is where the subject listen to the test audio and score each audio regard-

ing each question in a scale of 0 to 100. Figure 35 shows the interface used in MATLAB.



(a) Training Phase



(b) Evaluation Phase

Figure 35: Graphical User Interface used in MUSHRA test

As an experimental informal test, only four test audio were included. They were generated by audio zooming with wFCM and DSB. Each method were simulated at two different reverberant environment, where RT_{60} =0ms and 400ms. Together with the anechoic speech (**reference**) and the echoic speech mixture (**anchor**), there were six audio signals for the test. Three participants were invited to the MUSHRA test. The test were surrounding two core questions. The first one was "rate the global quality compared to the reference for each test signal". The results are shown in Table 8.

			RT_{60} = 0ms		RT_{60} = 400ms	
Participant	Reference	Anchor	DSB	wFCM	DSB	wFCM
A	100	18	50	90	21	21
B	100	27	50	95	28	30
C	100	20	50	90	24	25
Mean	**100**	**21.67**	**50**	**91.67**	**24.33**	**25.33**

Table 8: MUSHRA results on the global quality of zoomed speeches

From Table 8, it can be seen that the results adhered with the conclusion from the objective tests. The results supported the conclusion that wFCM outperformed DSB under free field. However, with reverberant situations, performance of both algorithms degraded, hence yielding low marks for both algorithms.

The second task was to "rate the closeness of the sound". This was aimed to ask about the perception of audio zooming - whether the techniques actually make the speaker sound closer. Results are shown in Table 9.

			RT_{60} = 0ms		RT_{60} = 300ms	
Participant	Reference	Anchor	DSB	wFCM	DSB	wFCM
A	100	30	60	100	30	30
B	100	27	63	100	29	27
C	100	33	61	97	33	31
Mean	**100**	**30**	**61.33**	**99**	**30.67**	**29.33**

Table 9: MUSHRA results on the closeness of zoomed speeches

The results, despite having a similar trend as the one in the objective tests, was found surprising. The score from this experiment had an significant increase as compared to rating of global quality. A hypothesis would be that "closeness of the sound" is a very subjective opinion and not defined clearly. For instance, in this project, a lot of the effort was done on source separation or interference suppression. However, there is no clear evidence that extracting a source out of a mixture correlates with the improvement in the closeness of the sound. This would be an interesting issue in progressing this project further forward.

6 Project Management

Apart from evaluating from a technical aspect, it is also important to monitor the project progress by evaluating the targets set in the project plan in the beginning of the project. Figure 36 shows the Gantt Chart laying out the project plan and the deliverables.

The first milestone in the plan was capturing audio data containing multiple speakers, with the deliverable being a robust simulation software to generate echoic speech data with specified microphone and speakers location. This was finally achieved by using *MCRoomSim* as a RIR generator, and performing convolution with anechoic speeches. This deliverable is therefore deemed successful.

The second planned milestone was to develop an audio zooming algorithm. In the Gantt chart, it was expected that developing the time-frequency mask via clustering would be the focus throughout the project. Nevertheless, following the discussion with the project supervisor, it was deemed that developing several zooming algorithms would allow a more complete and thorough understanding towards the audio zooming problem. This deviated from the project plan, hence tightened the time constraint. The total time spent on algorithmic development was, therefore, extended by two weeks as compared with the project plan.

The last milestone was about evaluating the performance of the developed algorithms. In the project plan, it was expected that both objective and subjective measures would be conducted. Despite completing this deliverable, due to time constraints, the subjective listening test was conducted informally, hence statistically insignificant. This fallback led to the discussion in Section 7.1.3 about the future directions of the project.

Overall, the project has followed the project plan and achieved most of the aims set up before the start of the project. Nonetheless, there are still a huge potential to improve the project, as laid out in Section 7.1.

MEng Final Year Project: Audio Signal Zoom for Small Microphone Arrays

Project Supervisor: Dr. Patrick Naylor
Project Lead: Chi Hang Leung
Project Start Date: 10/16/2017 (Monday)

WBS	Task	Start	End	Weeks
1	**Capturing Audio Data and Establishing Testbench**	**Mon 16/10/17**	**Sun 17/12/17**	**9**
1.1	Research on Room Acoustics and Generating Room Impulse Response (RIR)	Mon 16/10/17	Sun 05/11/17	3
1.2	Capturing/Fetching Anechoic Speech Examples from available Databases	Mon 23/10/17	Sun 29/10/17	1
1.3	Developing the RIR Generator Software	Mon 30/10/17	Sun 26/11/17	4
1.4	Creating Test Cases with Scenarios with Different Microphones and Speaker Location	Mon 20/11/17	Sun 03/12/17	2
1.5	Generating the Room Impulse Response and Filter the Anechoic Audio Data	Mon 04/12/17	Sun 17/12/17	2
2	**Developing Processing Algorithm**	**Mon 18/12/17**	**Sun 06/05/18**	**20**
2.1	Background Research on Time-Frequency Analysis	Mon 18/12/17	Sun 14/01/18	4
2.2	Familiarising with Time-Frequency Analysis using Spectrogram and Overlap-add Method	Mon 08/01/18	Sun 21/01/18	2
2.3	Experimenting Binary Oracle Mask based on Phase Difference of Arrival	Mon 15/01/18	Sun 04/02/18	3
2.4	Background Reading on Clustering	Mon 29/01/18	Sun 25/02/18	4
2.5	Developing the Algorithm with Clustering Time-frequency bins on Phase-Spectrogram	Mon 19/02/18	Sun 15/04/18	8
2.6	Exploring the possibility of enhancements by integrating techniques, such as Beamforming, Blind Source Separation	Mon 16/04/18	Sun 06/05/18	3
3	**Evaluating the Performance of the Algorithm**	**Mon 07/05/18**	**Sun 24/06/18**	**7**
3.1	Setting the Standard to Evaluate the Performance of Audio Zooming	Mon 07/05/18	Sun 20/05/18	2
3.2	Refining the Algorithm Based on the evaluation according to the standard	Mon 22/05/17	Sun 04/06/17	2
3.3	Demonstrating the final results (potentially with a real-time demo) in Presentation	Mon 28/05/18	Sun 24/06/18	4

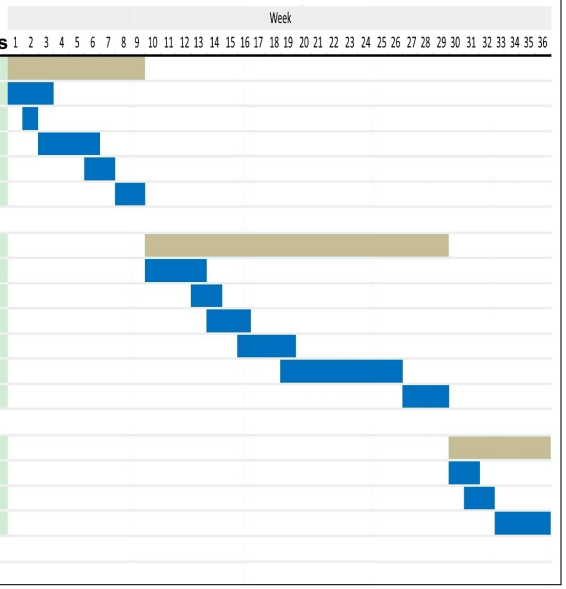


Figure 36: Gantt Chart

7 Conclusion

In this project, the aim was to develop offline audio zooming algorithms that would be applicable in the context of smartphones. With the assistance of synthetic audio data generated by convolution between simulated room impulse responses and anechoic speeches, three different audio zooming techniques were investigated.

Firstly, beamforming was experimented. As an established technique for large microphone arrays, when used with a small microphone array with two microphones, this approach suffers from the challenge of wide beamwidth (low spatial selectivity) and spatial aliasing. Despite facing such challenges, beamforming was able to produce a moderate amount of audio zooming and a significant amount of interference cancellation, which opened up the opportunity of being an enhancement technique.

The second experiment was time-frequency mask estimation by clustering the time difference of arrival of every time-frequency point. Binary (hard) mask and fuzzy (soft) mask were discussed and experimented. Experiment showed that soft mask produced promising audio zooming effect in terms of both speech quality and intelligibility. However, the performance drops when facing reverberant environment, due to the evolution of reflected paths which degrades the approximate W-disjoint orthogonality condition.

The third approach was time-frequency mask estimation using deep neural networks. Due to time constraints, the report was only able to provide a qualitative analysis. Despite the lack of experimentation, it was believed that deep neural networks have the potential of learning discriminative features from composition of weak features, such as the phase difference between two channels under reverberant conditions used in the second approach.

To conclude, soft time-frequency masks has shown promising results in performing audio zoom. While estimation using fuzzy c-means clustering on phase difference yielded satisfactory results under anechoic or lightly reverberant environment, the performance significantly deteriorates Hence, the future vision of the project would be to develop an audio zooming algorithm that perform robustly in reverberant environment.

7.1 Future Works

Below outlines some further improvements that could have been achieved if more time has been given.

7.1.1 Project Scope

Throughout the project, audio zooming algorithms are based and tested against synthetic data using image method, which limits to audio data captured in a

rectangular room. It would be advantageous to test the algorithms against real-life data to understand their performance and limitations. Hence, the next step of the project would be to capture real-life audio data using smartphones under different reverberation conditions and room dimensions.

Another step forward would be increasing the number of speakers. In the project, an assumption of having only two speakers was made. Yet, in real-life, the number of speakers in a video could be far more than two. Further investigation and validation has to be made on the algorithms under multi-speaker environment.

7.1.2 Processing Algorithms

In this project, the approach of using machine learning for audio zoom has been discussed using literatures and previous researches. One of the aims in the future would be to experiment with deep neural networks using monaural and binaural features according to the designed framework in Figure 32. This would be the primary target in the future.

The secondary target would be to try combining different algorithms. For example, TF mask could potentially be coupled with beamforming for a better outcome. The reason is that beamforming has shown a promising effort in nullifying sources from a certain angle, while clustering has demonstrated its ability to localise the interfering sources. By combining both methods, one could potentially accurately steer the nulls onto the interfering signals, yielding a better audio zooming effect.

7.1.3 Evaluation Metrics

Besides, regarding evaluation section of the project, in this project, an informal MUSHRA listening test was conducted. It was, however, statistically insignificant due to insufficient subjects in the test. To develop a statistically significant test, there is a new trend of web-based auditory experiments, with its advantage being its inherited ability to engage more participants. This hence gives a potentially even more reliable result, and could potentially make the test more statistically significant. Hence, one of the future works would be the migrate the MUSHRA test used in the project onto an open web platform. An online listening test prototype based on webMUSHRA [53] has been built, and is available on `https://chl214.github.io/webMUSHRA/`.

References

[1] E. C. Cherry, "Some Experiments on the Recognition of Speech, with One and with Two Ears," *The Journal of the Acoustical Society of America*, vol. 25, no. 5, pp. 975–979, 9 1953.

[2] O. M. Mitchell, C. A. Ross, and G. H. Yates, "Signal Processing For A Cocktail Party Effect," *The Journal of the Acoustical Society of America*, vol. 50, no. 2, pp. 656–660, 1971.

[3] M. Jeub, C. Herglotz, C. Nelke, C. Beaugeant, and P. Vary, "Noise reduction for dual-microphone mobile phones exploiting power level differences," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*. IEEE, 3 2012, pp. 1693–1696.

[4] A. Jourjine, S. Rickard, and O. Yilmaz, "Blind separation of disjoint orthogonal signals: demixing N sources from 2 mixtures," in *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.00CH37100)*, vol. 5. IEEE, pp. 2985–2988.

[5] S. Rickard and O. Yilmaz, "On the approximate W-disjoint orthogonality of speech," *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. 1, pp. 529–532, 5 2002.

[6] Y. L. Yu, Y. T. Chao, L. C. Lee, J. Y. Yen, and Y. C. Fan, "A Novel Soundproof Ventilation Plant Design with High Performance and No Energy Consumption," *Mathematical Problems in Engineering*, vol. 2015, pp. 1–10, 2015.

[7] W. C. Sabine and C. H. Dwight, "Collected Papers on Acoustics," *American Journal of Physics*, vol. 34, no. 4, pp. 370–371, 4 1966.

[8] H. E. White and D. H. White, *Physics and music : the science of musical sound*. Saunders College Pub, 1980.

[9] D. Murphy, M. Beeson, S. Shelley, A. Moore, and A. Southern, "Hybrid Room Impulse Response Synthesis in Digital Waveguide Mesh Based Room Acoustics Simulation," *Proceedings of the 11th International Conference on Digital Audio Effects*, pp. 2–9, 2008.

[10] J. B. Allen and D. A. Berkley, "Image method for efficiently simulating small-room acoustics," *The Journal of the Acoustical Society of America*, vol. 65, no. 4, pp. 943–950, 1979.

[11] The Lab Book Pages, "Delay Calculation." [Online]. Available: http://www.labbookpages.co.uk/audio/beamforming/delayCalc.html

[12] M. J. Hinich, "Processing spatially aliased-arrays," *The Journal of the Acoustical Society of America*, vol. 64, no. 3, pp. 792–794, 1978.

[13] Guido Dietl, "Beamforming with Linear Antenna Array." [Online]. Available: https://www.geogebra.org/m/ArF3sKpW

[14] V. Rabinovich and N. Alexandrov, "Typical Array Geometries and Basic Beam Steering Methods," in *Antenna Arrays and Automotive Applications*. New York, NY: Springer New York, 2013, pp. 23–54.

[15] O. Yilmaz and S. Rickard, "Blind Separation of Speech Mixtures via Time-Frequency Masking," *IEEE Transactions on Signal Processing*, vol. 52, no. 7, pp. 1830–1847, 7 2004.

[16] A. Blin, S. Araki, and S. Makino, "Underdetermined blind separation of convolutive mixtures of speech using time-frequency mask and mixing matrix estimation," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E88-A, no. 7, pp. 1693–1699, 2005.

[17] C. Faller and J. Merimaa, "Source localization in complex listening situations: Selection of binaural cues based on interaural coherence," *The Journal of the Acoustical Society of America*, vol. 116, no. 5, pp. 3075–3089, 2004.

[18] H. Sawada, S. Araki, and S. Makino, "Underdetermined convolutive blind source separation via frequency bin-wise clustering and permutation alignment," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 19, no. 3, pp. 516–527, 3 2011.

[19] I. Jafari, M. Atcheson, R. Togneri, and S. Nordholm, "Time-frequency clustering with weighted and contextual information for convolutive blind source separation," in *IEEE Workshop on Statistical Signal Processing Proceedings*. IEEE, 6 2014, pp. 157–160.

[20] M. Kühne, R. Togneri, and S. Nordholm, "A novel fuzzy clustering algorithm using observation weighting and context information for reverberant blind speech separation," *Signal Processing*, vol. 90, no. 2, pp. 653–669, 2010.

[21] C. Knapp and G. Carter, "The generalized correlation method for estimation of time delay," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 24, no. 4, pp. 320–327, 8 1976.

[22] A. D. Firoozabadi and H. R. Abutalebi, "Subband processing-based approach for the localisation of two simultaneous speakers," *IET Signal Processing*, vol. 8, no. August 2013, pp. 996–1008, 12 2014.

[23] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "An algorithm for intelligibility prediction of time-frequency weighted noisy speech," *IEEE*

Transactions on Audio, Speech and Language Processing, vol. 19, no. 7, pp. 2125–2136, 2011.

[24] E. Vincent, R. Gribonval, and C. Fevotte, "Performance measurement in blind audio source separation," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, no. 4, pp. 1462–1469, 2006.

[25] C. Févotte, R. Gribonval, and E. Vincent, "BSS_EVAL Toolbox User Guide Revision 2.0," 2005. [Online]. Available: https://hal.inria.fr/inria-00564760/document

[26] A. Rix, J. Beerends, M. Hollier, and A. Hekstra, "Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs," in *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.01CH37221)*, vol. 2. IEEE, pp. 749–752.

[27] ITU-T, "P.800.1: Mean Opinion Score (MOS) terminology," *ITU-T Recommendation*, p. 12, 2006.

[28] ITU-T, "Rec. P.862.1, Mapping function for transforming P.862 raw result scores to MOS-LQO," *International Telecommunication Union*, 2003.

[29] ITU-T, "P.800: Methods for subjective determination of transmission quality," *ITU-T Recommendation*, vol. 800, 1996.

[30] M. D. Brookes, "VOICEBOX: A speech processing toolbox for MATLAB." 1997. [Online]. Available: http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html

[31] International Telecommunication Union, "ITU-R BS.1534-3, Method for the subjective assessment of intermediate quality level of audio systems," *ITU-R Recommendation*, vol. 1534-3, pp. 1534–3, 2015.

[32] ITU-T, "ITU-T P.56 Objective measurement of active speech level," *ITU-T Recommendation*, vol. 5, 2011.

[33] E. Habets, "Room impulse response generator for MATLAB," 2017. [Online]. Available: https://www.audiolabs-erlangen.de/fau/professor/habets/software/rir-generator

[34] S. M. Schimmel, M. F. Muller, and N. Dillier, "A fast and accurate shoebox room acoustics simulator," in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 4 2009, pp. 241–244.

[35] A. Wabnitz, N. Epain, C. Jin, and A. Van Schaik, "Room acoustics simulation for multichannel microphone arrays," in *the International Symposium on Room Acoustics, ISRA 2010, Melbourne, Australia*, 2010.

[36] W. T. Chu and A. C. C. Warnock, "Detailed Directivity of Sound Fields Around Human Talkers," *Research Report (National Research Council Canada. Institute for Research in Construction); no. RR-104*, 2002.

[37] R. L. Freeman, *Fundamentals of telecommunications*. John Wiley & Sons., 2005.

[38] E. A. Habets, J. Benesty, S. Gannot, P. A. Naylor, and I. Cohen, "On the application of the LCMV beamformer to speech enhancement," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*. IEEE, 10 2009, pp. 141–144.

[39] J. Benesty, J. Chen, and Y. Huang, *Microphone array signal processing*. Springer Science & Business Media, 2008, vol. 1.

[40] L. Griffiths and C. Jim, "An alternative approach to linearly constrained adaptive beamforming," *IEEE Transactions on Antennas and Propagation*, vol. 30, no. 1, pp. 27–34, 1 1982.

[41] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: A K-Means Clustering Algorithm," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. 100–108, 1979.

[42] J. C. Bezdek, "Objective Function Clustering," in *Pattern Recognition with Fuzzy Objective Function Algorithms*. Boston, MA: Springer US, 1981, pp. 43–93.

[43] M. Kuhne, R. Togneri, and S. Nordholm, "Robust Source Localization in Reverberant Environments Based on Weighted Fuzzy Clustering," *IEEE Signal Processing Letters*, vol. 16, no. 2, p. 85, 2 2009.

[44] N. R. Pal and J. C. Bezdek, "On Cluster Validity for the Fuzzy c-Means Model," *IEEE Transactions on Fuzzy Systems*, vol. 3, no. 3, pp. 370–379, 1995.

[45] B. Loesch and B. Yang, "Source number estimation and clustering for underdetermined blind source separation," *Proceedings of the 11th International Workshop on Acoustic Echo and Noise Control*, no. 5, pp. 943–952, 2008.

[46] D. Arthur and S. Vassilvitskii, "How slow is the k-means method?" *Proceedings of the twenty-second annual symposium on Computational geometry - SCG '06*, p. 144, 2006.

[47] J. Schmidhuber, "Deep Learning in neural networks: An overview," pp. 85–117, 4 2015.

[48] D. Yu, M. L. Seltzer, J. Li, J.-T. Huang, and F. Seide, "Feature learning in deep neural networks-studies on speech recognition tasks," *arXiv preprint arXiv:1301.3605*, 2013.

[49] P. S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, "Joint Optimization of Masks and Deep Recurrent Neural Networks for Monaural Source Separation," *IEEE/ACM Transactions on Audio Speech and Language Processing*, vol. 23, no. 12, pp. 2136–2147, 2015.

[50] C. J. Darwin, "Listening to speech in the presence of other sounds," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 363, no. 1493, pp. 1011–1021, 2008.

[51] Y. Jiang, D. L. Wang, R. S. Liu, and Z. M. Feng, "Binaural classification for reverberant speech segregation using deep neural networks," *IEEE/ACM Transactions on Audio Speech and Language Processing*, vol. 22, no. 12, pp. 2112–2121, 12 2014.

[52] V. Emiya, E. Vincent, N. Harlander, and V. Hohmann, "Subjective and objective quality assessment of audio source separation," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 19, no. 7, pp. 2046–2057, 9 2011.

[53] M. Schoeffler, F.-R. Stöter, B. Edler, and J. Herre, "Towards the Next Generation of Web-based Experiments: A Case Study Assessing Basic Audio Quality Following the ITU-R Recommendation BS.1534 (MUSHRA)," in *1st Web Audio Conference*, Paris, France, 2015.

Appendices

A Results of Baseline Testing of Synthetic Audio Data

This section displays the results from 9 baseline cases simulated in different room geometries and different source-receiver locations. Speakers are labelled in red, while microphones are labelled in blue. Arrows represent the directivity of the source, i.e. the direction that the speaker is facing.

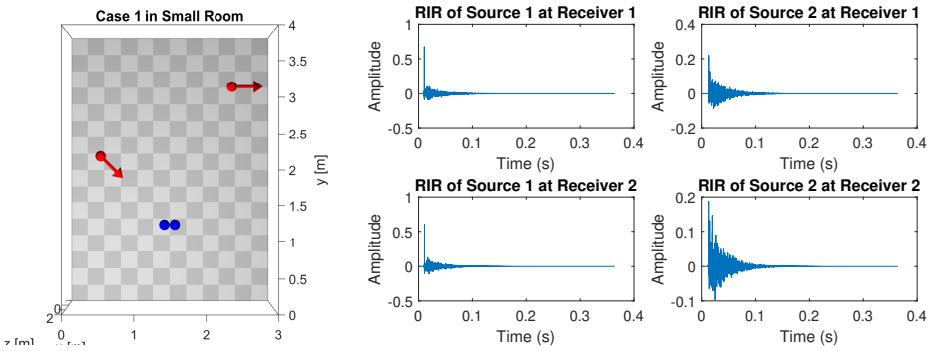


Figure 37: Results of Baseline Case 1 in a Small Room

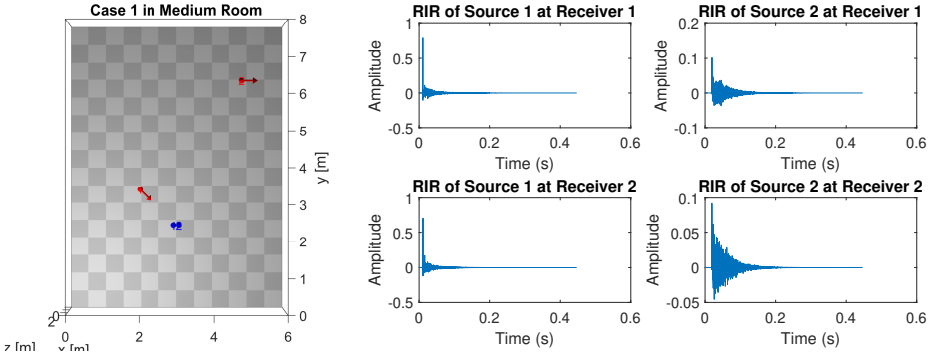


Figure 38: Results of Baseline Case 1 in a Medium Room

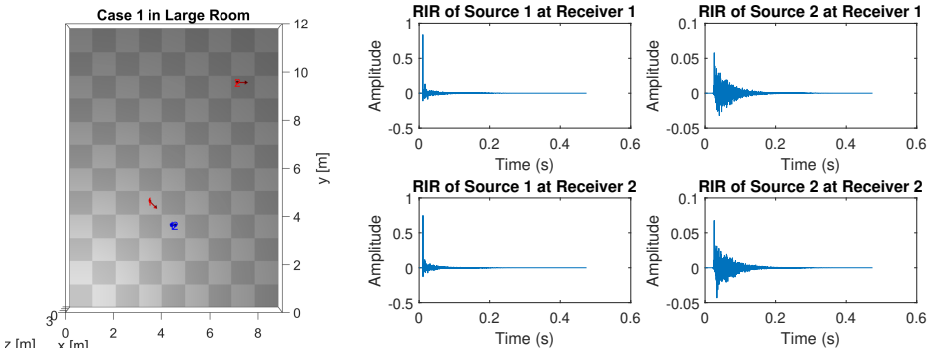


Figure 39: Results of Baseline Case 1 in a Large Room

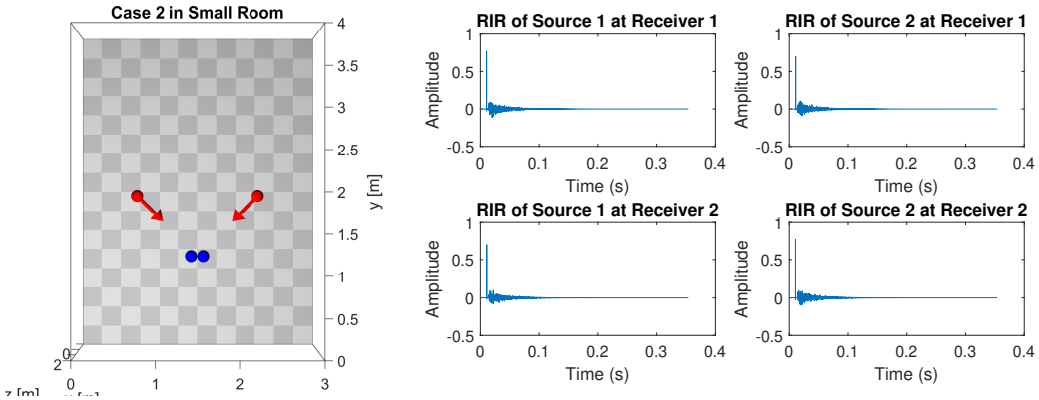


Figure 40: Results of Baseline Case 2 in a Small Room

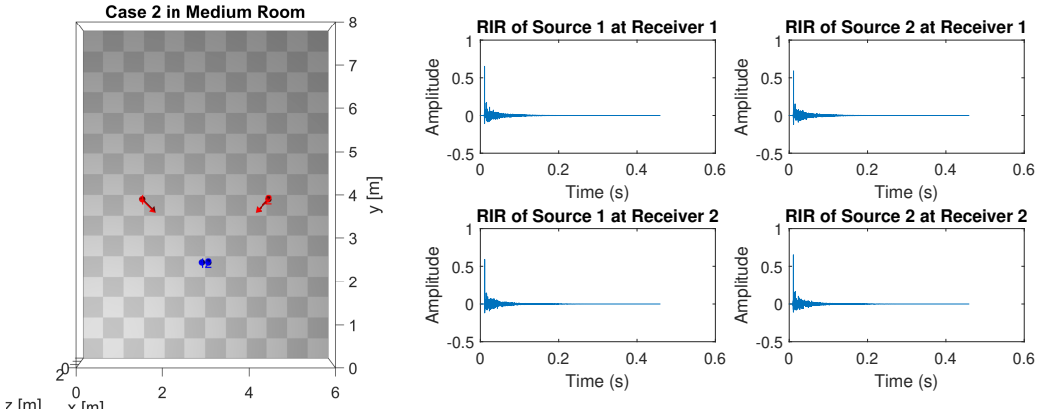


Figure 41: Results of Baseline Case 2 in a Medium Room

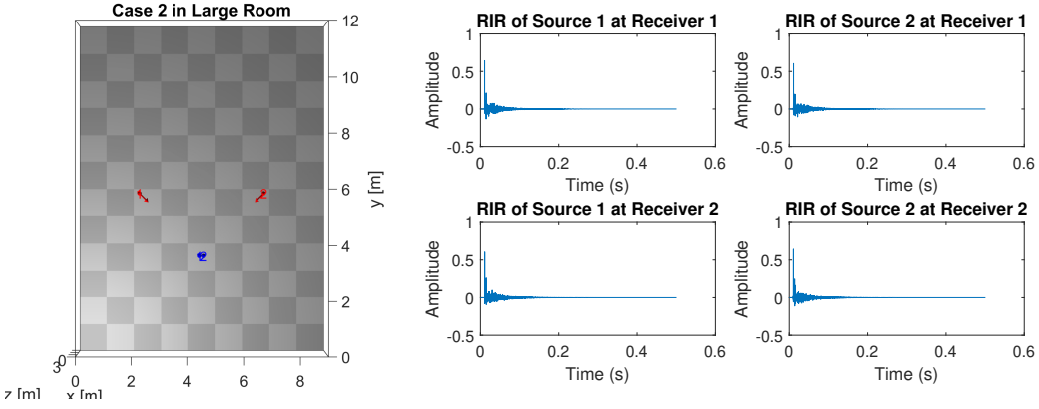


Figure 42: Results of Baseline Case 2 in a Large Room

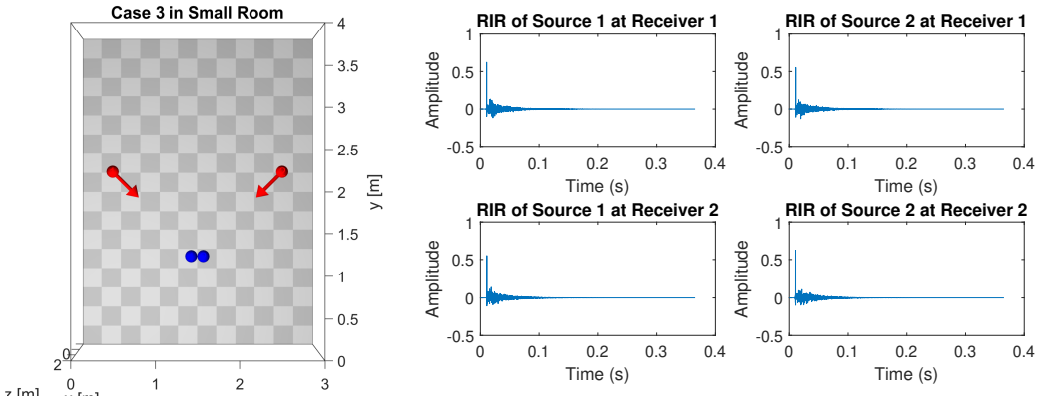


Figure 43: Results of Baseline Case 3 in a Small Room

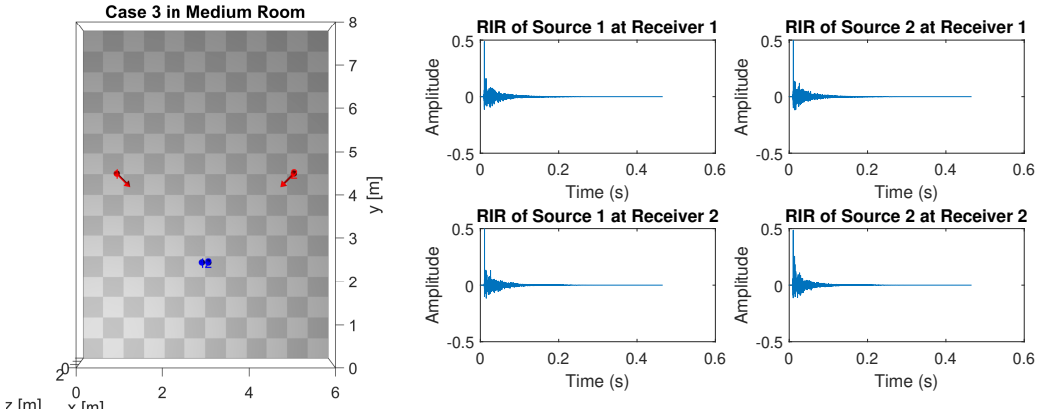


Figure 44: Results of Baseline Case 3 in a Medium Room

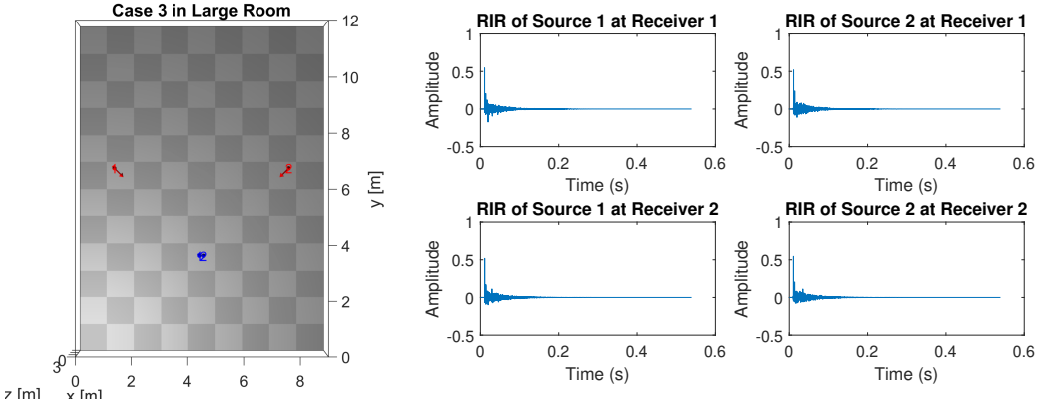


Figure 45: Results of Baseline Case 3 in a Large Room

B Code Listing

B.1 Beamforming

```
% Chi Hang Leung, EE4, 2018, Imperial College.
% 18/6/2018
%%%%%%%%%%%%%%%%%%%%%%%%%
% Perform beamforming on the received signals
%%%%%%%%%%%%%%%%%%%%%%%%%
% Inputs
% y_rec (2x1 cell array) = the received signals at two microphones stored
% in cell array
% method = beamforming method
% Receivers = Structure from MCRoomSim containing information about the
%       microphpones
% fs = sampling frequency of the signal
% DOA = the DOA of the desired speaker
%%%%%%%%%%%%%%%%%%%%%%%%%
% Outputs
% y = Beamformed Signal in time domain
%%%%%%%%%%%%%%%%%%%%%%%%%

function y = Beamforming(y_rec,DOA, Receivers,fs,method,fc)
c = 340;
d = norm(Receivers(1).Location-Receivers(2).Location);
numRec = size(Receivers,1);
switch lower(method)
  case 'delaysum' %Delay Sum Beamformer
    tau = d/c*sind(DOA);
    nsamp = round(tau*fs);
    y_rec1_delayed = circshift(y_rec{1},nsamp);
    if nsamp >0
      y_rec1_delayed(1:nsamp)=0;
    else
      y_rec1_delayed(end-nsamp+1:end)=0;
    end
    y = y_rec1_delayed+y_rec{2};
    y = y./numRec;


  case 'delaysum_matlab' %Delay Sum Beamformer - MATLAB version
    array = phased.ULA('NumElements',2,'ElementSpacing',d);
    beamformer = phased.TimeDelayBeamformer('SensorArray',array,...
      'SampleRate',fs,...
      'PropagationSpeed',c,...
      'Direction',[-DOA; 0],...
      'WeightsOutputPort',true);
    [y,] = beamformer(cell2mat(y_rec)');
    y=y';

    tau = d/c*sind(DOA);
    w = [exp(-1j*2*pi*fc*tau);1];
    figure;
    [PAT,AZ_ANG,~]=pattern(array,fc,[-180:180],0,...
      'Type','power',...
      'PropagationSpeed',c,...
      'Weights',w);
    polarplot(deg2rad(AZ_ANG),PAT,'LineWidth',1.2)
    ax = gca;
    ax.ThetaZeroLocation='Top';
    ax.ThetaLim = [-180 180];

  case 'gsc_matlab' %Generalised Sidelobe Canceller
    array = phased.ULA('NumElements',2,'ElementSpacing',d);

    gscbeamformer = phased.GSCBeamformer('SensorArray',array, ...
      'PropagationSpeed',c,...
      'SampleRate',fs,...
      'Direction',[-DOA; 0], ...
      'FilterLength',10);
    y = gscbeamformer(cell2mat(y_rec)');
    y = real(y)';

  case 'mvdr_matlab' %Minimum Variance Distortionless Response
    array = phased.ULA('NumElements',2,'ElementSpacing',d);

    % Define the MVDR beamformer
    mvdrbeamformer = phased.MVDRBeamformer('SensorArray',array,...
      'Direction',[-DOA; 0],...
      'PropagationSpeed',c,...
      'OperatingFrequency',fc,...
      'WeightsOutputPort',true);
    [y, w] = mvdrbeamformer(cell2mat(y_rec)');
    y = real(y)';

    %Plot the array pattern
    figure;
    [PAT,AZ_ANG,~]=pattern(array,fc,[-180:180],0,...
      'Type','power',...
      'PropagationSpeed',c,...
      'Weights',conj(w),...
      'Normalize',false);
    polarplot(deg2rad(AZ_ANG),PAT,'LineWidth',1.2)
```

```
90      ax = gca;
91      ax.ThetaZeroLocation='Top';
92      ax.ThetaLim = [-180 180];
93
94    case 'lcmv_matlab' %Linear
        Constrained Minimum Variance
95      array = phased.ULA('NumElements
        ',2,'ElementSpacing',d);
96      steer_vec = phased.
        SteeringVector('SensorArray',
        array,...
97        'PropagationSpeed',c);
98      LCMVbeamformer = phased.
        LCMVBeamformer('Constraint',
        steer_vec(fc,[-DOA; 0]),...
99        'DesiredResponse',1,...
100       'WeightsOutputPort',true);
101     [y,w] = LCMVbeamformer(cell2mat
        (y_rec)');
102     y = real(y)';
103
104     %Plot the array pattern
105     figure;
106     [PAT,AZ_ANG,~]=pattern(array,fc
        ,[-180:180],0,...
107       'Type','power',...
108       'PropagationSpeed',c,...
109       'Weights',w);
110     polarplot(deg2rad(AZ_ANG),PAT,'
        LineWidth',1.2)
111     ax = gca;
112     ax.ThetaZeroLocation='Top';
113     ax.ThetaLim = [-180 180];
114
115   case 'null-steering' %Null
        Steering Beamforming
116     array = phased.ULA('NumElements
        ',2,'ElementSpacing',d);
117     lambda = c/fc; % wavelength
118
119     thetaad = -30:5:30;   % look
        directions
120     thetaan = 45;       %
        interference direction
121
122     elementPos = getElementPosition
        (array);
123     % Calculate the steering vector
         for null directions
124     wn = steervec(elementPos/lambda
        ,thetaan);
125
126     % Calculate the steering
        vectors for lookout directions
127     wd = steervec(elementPos/lambda
        ,thetaad);
128
129     % Compute the response of
        desired steering at null
        direction
130     rn = wn'*wd/(wn'*wn);
131
132     % Sidelobe canceller - remove
        the response at null direction
133     w = wd-wn*rn;
134
135     % Plot the pattern
136     figure;
137     pattern(array,fc,-180:180,0,'
        PropagationSpeed',c,'Type','
        powerdb',...
138       'CoordinateSystem','
        rectangular','Weights',w);
139     hold on; legend off;
140     plot([40 40],[-100 0],'r--','
        LineWidth',2)
141     text(40.5,-5,'\leftarrow
        Interference Direction','
        Interpreter','tex',...
142       'Color','r','FontSize',10)
143
144     y = real(y_rec{1}-w'*cell2mat(
        y_rec));
145   end
146 end
```

Listing 1: Beamforming

B.2 Time-frequency Masking through Clustering

```
% Chi Hang Leung, EE4, 2018, Imperial College.
% 18/6/2018
%%%%%%%%%%%%%%%%%%%%%%%%%
% Perform Clustering on TDOA found by Phase Difference of the input signals
%%%%%%%%%%%%%%%%%%%%%%%%%
% Inputs
% F (2x1 cell array) = STFT (using VOICEBOX) of the received signals at two microphones
% clusterMethod = 'naive'/'kmeans'/'FCM'/'wFCM'/'wcFCM'
% Receivers = Structure from MCRoomSim containing information about the
%       microphpones
% numSrc = number of Speakers
% fs = sampling frequency of the signal
% INC = windows increment
% desiredDOA = the DOA of the desired speaker
%%%%%%%%%%%%%%%%%%%%%%%%%
% Outputs
% F_output = the STFT representation after TF mask
%%%%%%%%%%%%%%%%%%%%%%%%%

function F_output = PhaseClustering(F,clusterMethod,Receivers,numSrc,fs,INC,desiredDOA)

%% Developing Binary Mask
%Find the Cross power spectral density for each time segment
G12 = F{2}.*conj(F{1});

%Plot the Spectrogram for Phase Difference
figure('pos',[150 300 400 300]);
t = (0:size(G12,1)-1)/fs*INC;
f = (0:size(G12,2)-1)/(size(G12,2)-1)*fs/2;
phaseDiff = angle(G12);%./repmat(2*pi*f,size(G12,1),1);
PlotPhaseSpectrogram(phaseDiff,f,t,'Phase Difference Spectrogram');

%Plot SNR Spectrogram
figure('pos',[150 300 400 300]);
Fpower=F{1}.*conj(F{1});
noiseSpect=estnoiseg(Fpower,INC/fs); % estimate the noise power spectrum
signalSpect=Fpower-noiseSpect;
signalSpect(signalSpect<0)=0;
SNR = signalSpect./noiseSpect;
PlotSpectrogram(SNR,f,t,'SNR Spectrogram');

%Find the TDOA of the signals for clustering
dmax = norm(Receivers(1).Location-Receivers(2).Location);
c = 340;
alpha = 2*pi;
timeDiff_complex = -angle(G12)./(alpha*repmat(f,size(G12,1),1));
timeDiff_complex(:,1)= 0; %Avoid error in clustering
timeDiff_reshaped = reshape(timeDiff_complex,[],1);
timeDiff_reshaped_norm = timeDiff_reshaped/norm(timeDiff_reshaped);
timeDiff_complex_norm = timeDiff_complex/norm(timeDiff_reshaped);

timeWin = 4;
freqWin = 3;

%Plot Histogram for DOA
timeDiff_complex(:,1)= NaN;
DOA = asind(timeDiff_complex/dmax*c);
DOA(imag(DOA)~=0)=NaN;
DOA = real(DOA);
figure;
histogram(DOA,60,'Normalization','probability');
grid on;
xlabel('DOA (deg)');
ylabel('Probability');

delay = dmax/c*sind(desiredDOA);
%% Clustering algorithm
switch lower(clusterMethod)
  case 'naive'
  %% Naive Binary Mask
    attenuationRatio = 0;
    Mask = double(phaseDiff<=0);
    Mask(Mask==0)=attenuationRatio; %Attenuate the unwanted TF bins

  case 'kmeans'
    [U,centers] = kmeans(timeDiff_reshaped_norm,numSrc,'MaxIter',1000,'Replicates',10);
    [~,index]= min(abs(centers*norm(timeDiff_reshaped)-delay));

    %Binary Mask
    Mask = reshape(double(U==index),size(G12));

  case 'fcm'
  %% Fuzzy c-means clustering
```

```
84      options = [NaN 100 1e-10 1];
85      [centers,U] = fcm(
        timeDiff_reshaped_norm,numSrc,
        options);
86
87      %Soft Mask
88      [~,index]= min(abs(centers*norm
        (timeDiff_reshaped)-delay));
89      Mask = reshape(U(index,:),size(
        G12));
90
91    case 'wfcm'
92    %% Weighted Fuzzy c-means
        clustering
93      options = [NaN 100 1e-10 1];
94
95      % Calculate the weights for
        wfcm
96      w = zeros(size(
        timeDiff_complex_norm));
97      K = 10^-4;
98      SNRmax = 100;
99      for i = 1:size(w,1)
100       minTimeIndex = max(1,i-
        timeWin);
101       maxTimeIndex = min(i+timeWin,
        size(w,1));
102       for j = 1:size(w,2)
103         minFreqIndex = max(1,j-
        freqWin);
104         maxFreqIndex = min(j+
        freqWin,size(w,2));
105         window =
        timeDiff_complex_norm(
        minTimeIndex:maxTimeIndex,
        minFreqIndex:maxFreqIndex);
106         window = reshape(window
        ,[],1);
107         w(i,j) = var(window);
108       end
109     end
110     w = max(w,K);
111     w = 1+1./w;
112 %     w = min(SNR,SNRmax); %SNR
        weighted
113     w = reshape(w,[],1);
114     timeDiff_reshaped_norm =
        reshape(timeDiff_complex_norm
        ,[],1);
115
116     [centers,U] = wfcm(
        timeDiff_reshaped_norm,w,numSrc
        ,options);
117
118     %Soft Mask
119     [~,index]= min(abs(centers*norm
        (timeDiff_reshaped)-delay));
120     Mask = reshape(U(index,:),size(
        G12));
121
122   case 'wcfcm'
123   %% Weighted Contextual FCM
124     options = [2 100 1e-10 1];
125
126     % Calculate the weights for
        wfcm
127     w = zeros(size(
        timeDiff_complex_norm));
128     C = cell(size(w));
129     K = 10^-3;
130     for i = 1:size(w,1)
131       minTimeIndex = max(1,i-
        timeWin);
132       maxTimeIndex = min(i+timeWin,
        size(w,1));
133       for j = 1:size(w,2)
134         minFreqIndex = max(1,j-
        freqWin);
135         maxFreqIndex = min(j+
        freqWin,size(w,2));
136         window =
        timeDiff_complex_norm(
        minTimeIndex:maxTimeIndex,
        minFreqIndex:maxFreqIndex);
137         window = reshape(window
        ,[],1);
138         w(i,j) = var(window);
139         window = zeros(size(
        timeDiff_complex_norm));
140         window(minTimeIndex:
        maxTimeIndex,minFreqIndex:
        maxFreqIndex)=1;
141         C{i,j} = find(window==1);
142       end
143     end
144     w = max(w,K);
145     w = 1+1./w;
146     w = reshape(w,[],1);
147     C = reshape(C,[],1);
148     timeDiff_reshaped_norm =
        reshape(timeDiff_complex_norm
        ,[],1);
149
150     %Validation
151     omega = length(
        timeDiff_reshaped_norm);
152     idx_ = randsample(omega,round(
        omega/10));
153     idx_v = false(size(
        timeDiff_reshaped_norm));
154     idx_v(idx_)= true;
155     idx_e = ~idx_v;
156     beta = 1e-2; %initial value of
        beta
157
158     [~,~,J_wfcm] = wfcm(
        timeDiff_reshaped_norm,w,numSrc
        ,options);
159     [~,~,J_wcfcm] = wcfcm(
        timeDiff_reshaped_norm,w,beta,C
        ,numSrc,options);
160
161
162     beta_inc = 0.001*J_wfcm(end)/(
        J_wcfcm(end)-J_wfcm(end))*beta;
163     beta = beta_inc;
164     q = options(1);
165     E_cv_best = Inf;
```

```
166     U=[];
167     for i=1:100
168       [centers,U] = wcfcmval(
        timeDiff_reshaped_norm,w,beta,C
        ,idx_e,U,numSrc,options);
169       dist = distfcm(centers,
        timeDiff_reshaped_norm(idx_v));
170       dist = dist.^2;
171       E_cv = sum(sum(dist.*[w(idx_v
        ) w(idx_v)]'.*(U(:,idx_v).^q)))
        ;
172       if E_cv <E_cv_best
173         E_cv_best = E_cv;
174         beta_best = beta;
175       else
176         break;
177       end
178       beta = beta+beta_inc;
179     end
180
181     [centers,U] = wcfcm(
        timeDiff_reshaped_norm,w,
        beta_best,C,numSrc,options);
182
183     %Soft Mask
184     [~,index]= min(abs(centers*norm
        (timeDiff_reshaped)-delay));
185     Mask = reshape(U(index,:),size(
        G12));
186
187 end
188 %% Apply the Mask
189 %Plot the masks
190 t = (1:size(G12,1))/fs*INC;
191 f = (0:size(G12,2)-1)/(size(G12,2)
        -1)*fs/2;
192 figure('pos',[150 300 400 300]);
193 PlotMask(Mask.',f,t,'TF Mask');
194
195 %Apply the mask and Put them back
        into a cell array
196 F_output = F{1}.*Mask;
```

Listing 2: Time-frequency Masking through clustering of phase difference