Extract Target Speech Signal from 3 Channel Recordings

Hongyu Zou 7493642
Liu Cheng 7486632
Department of Electrical and Computer Engineering
University of Ottawa, ON, Canada, K1N 6N5
Email: hzou047@uottawa.ca, lchen156@uottawa.ca,

***Abstract*—Currently a good method to extract directional audio is based on microphone array system. A novel technique introduces such three stages for extraction, which involve Time Difference of Arrival calculation, MVDR beamformer filtering with directional interference deduction and parametric spectral subtraction for general background noise deduction. In this paper, principles of certain methods will be explored and explained. Furthermore, a specific experiment targets on given piece of audio (captured by three microphone array system and collect three different directional source signals), as a result, audio of source of English female will be extracted. A general look will discuss the quality of result in terms of human perceptual hearing. Related experiment procedure will be given for further development.**

***Index Terms*—Time difference of arrival, beamformer filtering, parametric spectral subtraction, microphone array, multi source audio extraction, phased array toolbox.**

I. Introduction

This paper targets on application of extracting audio. Mixture audio is captured by three microphone array system. In this stage, the direction of sources will be calculated based Time Difference of Arrival (TDOA) [1]. Through the results of time difference and transform function, angles of direction will be brought to next stage calculation which aims to eliminate certain direction of interference. In our last stage, choosing simple parametric spectral subtraction method to deduct background noise. In following sections, principles are further developed with experimental parameters and results, core code and formulas are illustrated, too. Experiment section is mainly discussing the result of experiment, while part of future work will be explored.

Our stimulate signal source has speeches of English female voice, English male voice and Germany female voice. The distance of each microphone is 25cm; thus we have two spaces in these three microphones. Assuming the source comes from far filed so that calculation base line is not fixed. Moreover, sound speed is assumed as 343 meters per second.

As shown in Figure 1, basic structure of microphone array is assumed in the same line and no consideration of altitude.

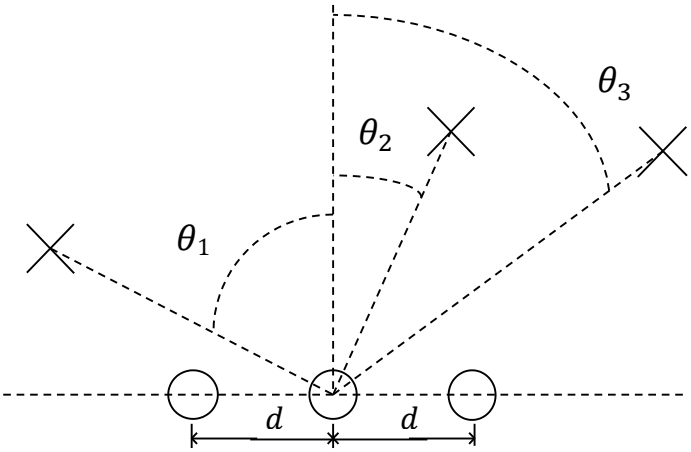


Figure 1 General view of microphone array in application

MVDR beamformer is used for directional interference elimination and it can be seen an adaptive beamformer, which has been widely used in signal extraction. In this paper, we will explain the basic principle and how this method works in our project. One of the most important thought is to observe the interferences as the point noise. We will also show the implementation procedure and result in the experiment part.

After the MVDR beamformer, we will get the English female voice with background noise. Based on the characteristics of the noise, and the speech we get which has two second clean noise signal before audible audio starts, we decide to use *Parametric Spectral Subtraction* method to reduce background noise. With progress in experiment, we will discuss principle of this method, and also its advantages and shortages.

II. Basic Background

TDOA algorithm is seen as improvement of TOA (Time of Arrival). TDOA method is originally an effective method to detect location of moving object. Experiment takes use of three microphone with three sources [4]; thus three TDOA should be concluded for each source. The advantage of TDOA includes multiple time delay and synchronize error. Beside TOA timestamp, TDOA only keeps length of time difference. In terms of two dimensional distance expression, it is concluded as [5]

$$\tau(t_1, t_2) = \sqrt{(x - x_1)^2 + (y - y_1)^2} - \sqrt{(x - x_2)^2 + (y - y_2)^2} \quad (1)$$

Microphone is well-known as sound pressure level pick-up which transform to voltage variation as electric signal. Generally, microphones have an Omni-directional polar pattern which means they are not extremely sensitive to certain direction. To locate direction of source, a method is taking use of TDOA technique. Cross-correlation function measures the similarity of signals which have different start point, defined as

$$R_{x1,x2}(\tau) = \sum_{n=-\infty}^{\infty} x_1(n)x_2(n+\tau) \quad (2)$$

Where τ denotes the time difference, to calculate maximum value of $R_{x1,x2}(\tau)$, τ_{true} will be got. Furthermore, as we preset the space d of each microphone by 25cm, simple expression for horizontal signal propagation will be

$$\tau_{true} = \frac{d}{c} \quad (3)$$

However, in our application, propagation is from two-dimensional propagation such that formula (3) is modifies as

$$\theta = arcsin\frac{\tau_{true}c}{d} \quad (4)$$

Which is an approximately calculation for angle θ, denoted in Figure 4.

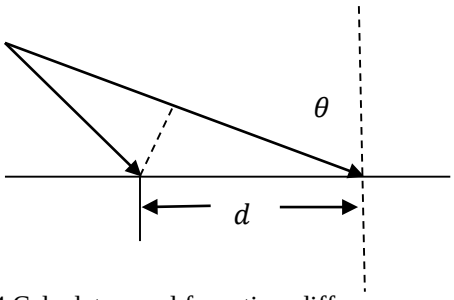


Figure 4 Calculate angel from time difference

The assumption which is used is based on following sets as pre knowledge which reduces the complexity of experiment:

- *all the sources are spatially stationary*
- *free field conditions are simulated, as opposed to reverberant conditions with possibly long echoes, this means that the impulse responses / frequency responses between the sources and the sensors are simple, i.e., they correspond to pure delays*
- *plane wave propagation is assumed, meaning that we consider that the sources are in the far field, with no near propagation field effects*
- *there is no physical object between microphones*
- *we assume a known and constant speed of sound here, but in practice the speed of sound is a variable that depends on temperature and humidity*
- *time-frequency statistics of the noise are stationary, and the simplest type of noise was used: white Gaussian noise. In addition, 2 seconds of noise-only signals are available at the beginning of the files.*
- *offline processing is possible*
- *it is often not possible in physical systems to have such large distances between microphones and large distances can help in the detection of the angles $\vartheta_1, \theta_2, \theta_3$ and in the target source extraction*

A microphone array, which has a set of microphones placed in a specific way, can easily get the signal information. Each sensor in the microphone array collect signal from its own field which is specified by the characteristic of the sensor. The output of the microphone array system will contain the desired signal, interference, and also the background noise. Nowdays, microphone array system has been widely used in localizing the sound sources, separating the multiple signals and extracting the desired signal. This method is named beamforming, which plays an important role in the system. There are two kinds of beamformers: fixed beamformers and adaptive beamformers.

One typical fixed beamformer is delay-and-sum beamformer which has static coefficients and independent signal responses. It is to make such length of delay to each sensor so that the desired signal can be got by all sensors. Weighting and add these delays together coherently, the signal of interest will be enhanced, while other interferences and noise will be eliminated. However, there is a backward: this method perform well only in processing the narrowed signal.

The signal in our project is speech which is a broadband signal. This kind of signal usually contains different frequencies which will have different spatial responses. The characteristics of the signal may lead to the distortion of the desired signal.

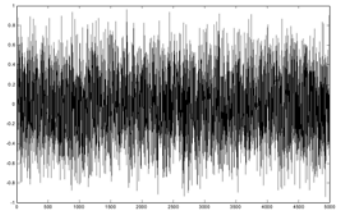
Even though fix beamformer can find some way to circumvent this issue and some of the methods have been widely used in processing the signal. There are still some shortages in fix beamformer. Fix beamformer always performs well in a field where the noise is not known as pre knowledge and isotropic. However, such assumption does not have impact in practice and this method is not the optimal way. In order to solve this problem, we can change the beamformer filter coefficients which will lead to new class of beamformer-adaptive beamformer.

The fundamental principle of adaptive beamforming is to track the statistics of the surrounding noise field and adaptively search for the optimum location of the nulls that can most significantly reduce noise under the constraint that the desired speech signal is not distorted at the beamformer's output.[6]

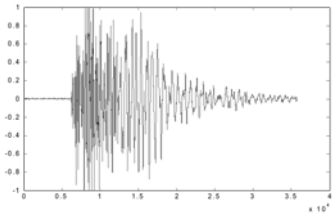
The most typical adaptive beamformer is the linearly constrained minimum variance (LCMV) and the minimum variance distortionless response (MVDR) is a particular case of LCMV method. The MVDR method has been widely used in separating multiple sound sources and acoustic signal processing. Nowdays, many applications have use this method process the speech signals.

In real world, there are many kinds of background noise such as white noise which is the most common one and the impulse noise usually occurs in explosion. In order to reduce the

background noise, Speech enhancement technique has been widely used in signal processing.



(a)



(b)

Figure 5 (a)white noise (b)impulse noise

In terms of noise deduction, there are numbers of choices of potential filters and methods. While in our experiment, after comparing several different methods with respect to Signal to Noise Ratio (SNR), parametric spectral subtraction is easy to realize and it has satisfied consequence, too. After filtering out the English female voice, parametric spectral subtraction keeps acoustic characteristic of voice without much distortion. Therefore, some brief introduction for this kind of method will be referred to in later session with experiment result.

III. CALCULATE TIME DIFFERENCE OF ARRIVAL FOR DIRECTIONAL FILTERING

Time differences of arrival (TDOA) is also known as an effective method to evaluate direction of audio source. In our experiment, 25cm microphone spacing is generally a large spacing; thus using generalized cross-correlation with phase transform (GCC-PHAT) is a good choice [7].

Since the whole audio piece is already known, it's possible to use Short Time Fourier Transform (STFT) to get this job for analyzing. To construct $R(\tau)$, illustrated in background session, the peaks in $R(\tau)$ indicate the arrival of three sources. Specifically in this project, there are three peaks and indicate for each of them. As a result, three mixture audio will generate nine peaks in terms of different comparison. While it's possible to get time differences by only two mixture audio, far field effect actually has less impact on different choices.

For STFT, mixture signals can be denoted as $\mathbf{x}(\mathrm{t},\mathrm{f}) = [x_1(t,f),[x_2(t,f),[x_3(t,f)]^T$, also known as time-frequency bin (the length of each bin is preset as parameter). The source signals will be denoted as $\mathbf{s}(\mathrm{t},\mathrm{f}) = [s_1(t,f),s_2(t,f),s_3(t,f)]^T$, where t and f are bin indexes. In addition, $\mathrm{d}(\mathrm{f},\tau_n)$ is steering vector for arrival time of n-th source, and $\mathbf{b}(\mathrm{t},\mathrm{f})$ denotes background noise in mixture signals. Therefore, mixture signals can be expressed as

$$\mathbf{x}(\mathrm{t},\mathrm{f}) = \sum_{n=1}^{N} \boldsymbol{d}(f,\tau_n)s_n(t,f) + \boldsymbol{b}(t,f) \tag{5}$$

After previous discussion, $R_{x1,x2}(\tau)$ is thus modified to

$$\phi^{sum}(\tau) = \sum_{t=1}^{T}\sum_{f=1}^{F}\phi(t,f,\tau) \tag{6}$$

Note here indexes are from 1 to T and 1 to F for t and f, respectively. We actually need to provide better performance, in other word, more data as possible as we can. Because of that, taking all frames is the best choice (to iterate all frames along the signal). This part is then described as

$$\phi^{max}(\tau) = \max_{t}\sum_{t=1}^{T}\sum_{f=1}^{F}\phi(t,f,\tau) \tag{7}$$

The exist method maps calculation of cross-correlation with time-frequency bin, forming *empirical covariance matrix* $\widehat{\boldsymbol{R}}_{\boldsymbol{xx}}(t,f)$, thus

$$\widehat{\boldsymbol{R}}_{\boldsymbol{xx}}(t,f) = \frac{\sum_{t',f'} w(t'-t,f'-f)\boldsymbol{x}(t',f')\boldsymbol{x}(t',f')^H}{\sum_{t',f'} w(t'-t,f'-f)} \tag{8}$$

Then we can apply it to local angular spectrum

$$\phi^{GCC}(t,f,\tau) = \Re\left(\frac{\widehat{\boldsymbol{R}}_{\boldsymbol{xx}}(t,f)_{1,2}}{\left|\widehat{\boldsymbol{R}}_{\boldsymbol{xx}}(t,f)_{1,2}\right|}e^{-2j\pi f\tau}\right) \tag{9}$$

In our experiment, *BSS Locate* toolbox is used for source localization in stereo convolutive audio mixtures by Blandin's work. As it is already been open source code, it's easier to do the modification based on related libraries. In fact, not only GCC-PHAT method is involved in, but also such method like GCC-NONLIN. We only focus on GCC-PHAT though. From the input of function, five parameters are specified:

- *x: samples which contain two mixture signals*
- *fs: sampling rate, unit in Hz*
- *d: microphone spacing in meters*
- *nsrc: number of sources*
- *local: angular spectrum function (such as GCC-PHAT, GCC-NONLIN, etc)*

Obviously, the input of signals contain two pieces of audio which forms as two row of vectors. Each vector include 12s audio signal, while sampling rate is 16000 Hz. Thus, totally 19200 samples are used. Spacing is 25 cm which is indicated by project assumption. Finally, number of sources is already known as 3 and spectrum method is GCC-PHAT. In the meanwhile, a dependent library is involved as *stft_multi* which will be used to measure STFT bin and related operation based on multiple bins. The output of function is not used since the average summation of mixture audio is not needed.

After calculated TDOA, using formula (4) to calculate angles for each source.

IV. BEAMFORMER FILTERING FOR ELIMINATING DIRECTIONAL INTERFERENCE

The purpose of our project is to make MVDR beamformer performs best in enhancing the signal of interest and eliminate interference as well as background noise reduction. As mentioned in part I, the speech contains three audio signals and a background noise, while English female audio is the one we need to extract. Up to now, we have got three angles of these three audio signals by using TDOA method in part III. Next step is to separate these three audio signals by using MVDR technique. We will use linear microphone array to collect signals and deal with interferences in the same manner as for point noise. Next session targets on the algorithm of MVDR.

Linear microphone array collects all the information which comes from the real acoustic environment. The information consists of the signal of interest, interferences and background noise. In order to make the method easier and more clear, we assume there is only one desired signal propagation from far field and collect it by a uniform linear microphone array. The number of the microphones is *M*. If we use first microphone as the object of reference, after the delay we will get the output of the *n*th microphone is:

$$y_n(t) = x_n(t) + v_n(t) = x(t - d_n) + v_n(t) \tag{10}$$

Where $y_n(t)$ ---complex signals collect by the *n*th microphone

$x_n(t)$ ---clean signal of the desired speech collected by the *n*th microphone

d_n ---time delay between the *n*th microphone and the reference one, $d_n = (n\text{-}1)\tau_0 \cos\theta_d$, $\tau_0 = s/c$, *s* is the distance between two neighbor sensors, *c* is the speed of the sound in air, which is 343m/s, θ_d is the angle of the desired signal.

Have Got the speech from the microphones (there are three speeches in our project), we operate the Fourier transform to these speeches and then work in the frequency domain.

$$\begin{aligned} Y_n(w) &= X_n(w) + V_n(w) \\ &= e^{-j(n-1)w\tau_0\cos\theta_d} X_n(w) + V_n(w) \end{aligned} \tag{11}$$

Where $Y_n(w)$, $X_n(w)$, $V_n(w)$ is the Fourier transform of $y_n(t)$, $x_n(t)$, $v_n(t)$.

As shown in figure 6, it is the microphone array system we used in our project, the waveform of the signal is plane wave.

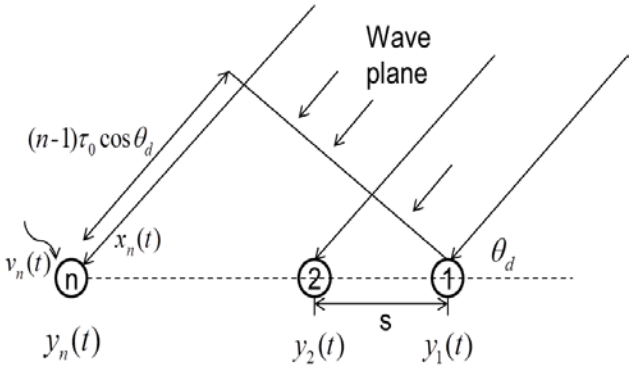


Figure 6 a uniform linear microphone array system

In order to use Matlab to implement the project, we prefer to use matrix to process the signal, so we rewrite the equation (11) into vector form:

$$\begin{aligned} Y(w) &= \left[Y_1(w)Y_2(w)...Y_M(w)\right]^T \\ &= \mathrm{D}_{\theta_d}(w)X(w) + \mathrm{v}(w) \end{aligned} \tag{12}$$

Where

$$\mathrm{D}_{\theta_d}(w) = \left[1 \quad e^{-jw\tau_0\cos\theta_d} \quad ..... \quad e^{-j(n-1)w\tau_0\cos\theta_d}\right]^T$$

and $\mathrm{v}(w)$ is the noise vector.

Our object of the project is to recover the clean speech using the $y(t)$ which is collected by the microphone array system. Usually, we need to figure out a filter $h(t)$ to reconstruct the clean signal.

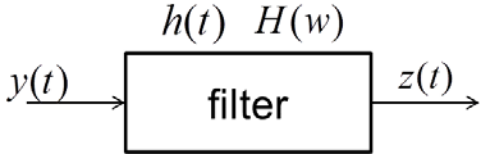


Figure 7 a simple reconstruct system

As shown in figure 7, the original signal can be reconstructed by the filter system: $z(t) = y(t) * h(t)$. In the specific case of our project we can operate the signal in frequency domain:

$$\begin{aligned} Z(w) &= \sum_{n=1}^{M} H_n^*(w)Y_n(w) \\ &= H^H(w)Y(w) \\ &= H^H(w)\mathrm{D}_{\theta_d}(w)X(w) + H^H(w)\mathrm{v}(w) \end{aligned} \tag{13}$$

Where $Z(w)$ ---reconstructed signal which is an estimate of $X(w)$.

$H(w)$ ---optimal filter and

$$H(w) = \left[H_1(w)H_2(w)...H_M(w)\right]^T \tag{14}$$

In order to obtain desired signal by using MVDR beamformer, we need to minimize the variance of noise and enhance signal in desired direction. If we want to get the original signal without distortion, *H(w)* should be

$$H(w) = \frac{Sn(w)^{-1} \mathrm{D}_{\theta_d}(w)}{\mathrm{D}_{\theta_d}(w)^H Sn(w)^{-1} \mathrm{D}_{\theta_d}(w)} \tag{15}$$

From the equation(10), we can know that the MVDR beamformer function consist of two parts, one is the steering vector defined by the desired direction and another one is the noise matrix which may consist of background noise and interferences.

In many conditions, the environment may be more complicated, sometimes there may be other sound sources which are the interferences. In our project, the German-speaking female and the English-speaking male are observed as interferences which will be processed as the point noise.

If there is only one point noise in the environment, then noise comprises of background noise and the point noise. The noise matrix can be written as:

$$Sn(w) = (1-\alpha_I) I_M + \alpha_I Sn_I(w) \tag{16}$$

Where α_I --- a parameter that controls the level of the point source noise relative to that of the white noise.[a]

I_M --- unitary matrix with the dimension of M

$Sn_I(w)$ ---defined by the direction of the point noise

$$Sn_I(w) = \mathrm{D}_{\theta_I}(w)\mathrm{D}_{\theta_I}(w)^H \tag{17}$$

In our project, there are two interferences, so we will talk about two points noise below. It is not hard to figure out we just need to modify the equation (16)

$$Sn_I(w) = \mathrm{D}_{\theta_{I_1}}(w)\mathrm{D}_{\theta_{I_1}}(w)^H + \mathrm{D}_{\theta_{I2}}(w)\mathrm{D}_{\theta_{I2}}(w)^H \tag{18}$$

Where

$$\mathrm{D}_{\theta_i}(w) = \begin{bmatrix} 1 & e^{-jw\tau_0\cos\theta_i} & ..... & e^{-j(n-1)w\tau_0\cos\theta_i} \end{bmatrix}^T \tag{19}$$

θ_i is the direction of the *i*th interference.

For now we have introduced that how we used the MVDR method to process the condition of our project. The Matlab result and analysis will be showed in part VI

V. Parametric Spectral Subtraction for Background Noise Deduction

Speech enhancement is a technology to extract useful speech signal from the noise in the background when the useful signal is interfered by all sorts of noise using the method of suppressing and decreasing the noise. In a word, it is to extract the pure original speech from the complex speech.

As for the broad band noise, we usually use the frequency-domain speech enhancement technology, which is an important technology in speech signal processing. It is widely used due to the simple principle and this method has been used in the application of reducing the noise of mobile phone.

Parametric spectral subtraction is a typical method of frequency-domain speech enhancement. There are three steps to describe the basic principle of this method:

Step1: Do the DFT operation to the speech signal with noise and the pure noise signal. After this we will get the amplitude spectrum of these two signals.

Step2: Do the operation of square to both signals' amplitude spectrum.

Step3: Reduce the result of noise signal which is got from the step2 from the result of speech signal.

Step4: Extract a root, after this we get the amplitude spectrum of the pure speech signal

Step5: Do the IDFT operation to the pure speech signal and this method needs the phase of the noise.

As the figure 8 shown, $x(n)$ is the speech with noise, $s(n)$ is the clean speech, $s'(n)$ is the estimation of the clean speech, $d(n)$ is the noise and $d'(n)$ is the estimation of noise

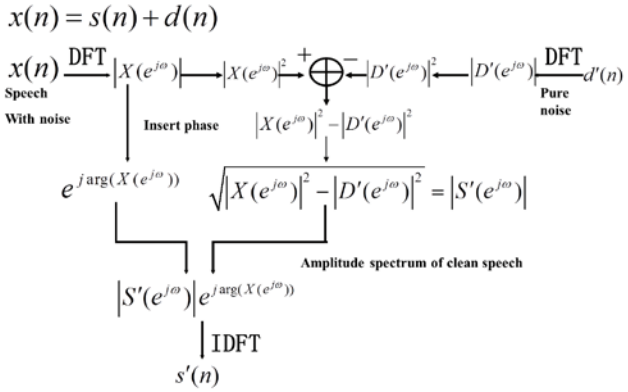


Figure 8 The flow chart of Parametric spectral subtraction

In order to apply this method, we first assume the speech (with noise) is the result of adding the clean signal and the pure noise together in a linear way. The clean signal and the noise are independent.

If the speech signal can be written as:

$$x(n) = s(n) + d(n) \tag{20}$$

then after Fourier transform, we will get:

$$X(e^{j\omega}) = S(e^{j\omega}) + D(e^{j\omega}) \tag{21}$$

$$X(e^{j\omega}) * X^*(e^{j\omega}) = \left[S(e^{j\omega}) + D(e^{j\omega})\right] * \left[S(e^{j\omega}) + D(e^{j\omega})\right]^* \tag{22}$$

Where $X^*(e^{j\omega})$ ---the conjugate of $X(e^{j\omega})$

$$\left|X(e^{j\omega})\right|^2 = S(e^{j\omega}) * S^*(e^{j\omega}) + S(e^{j\omega}) * D^*(e^{j\omega}) + D(e^{j\omega}) * S^*(e^{j\omega}) + D(e^{j\omega}) * D^*(e^{j\omega}) \tag{23}$$

$$\left|X(e^{j\omega})\right|^2 = \left|S(e^{j\omega})\right|^2 + \left|D(e^{j\omega})\right|^2 + S(e^{j\omega}) * D^*(e^{j\omega}) + D(e^{j\omega}) * S^*(e^{j\omega}) \tag{24}$$

To take the mathematical expectation on both sides of equation (24), we will get:

$$E\left[\left|X(e^{j\omega})\right|^2\right] = E\left[\left|S(e^{j\omega})\right|^2\right] + E\left[\left|D(e^{j\omega})\right|^2\right] + E\left[S(e^{j\omega}) * D^*(e^{j\omega})\right] + E\left[D(e^{j\omega}) * S^*(e^{j\omega})\right] \tag{25}$$

Where $E[.]$ --- the mathematical expectation

Due to noise and clean speech signal are independent and the operation of Fourier transform will not change the correlation of these two signals, so the last two terms of the equation (25) will be zero, and it will become:

$$E\left[\left|X(e^{j\omega})\right|^2\right] = E\left[\left|S(e^{j\omega})\right|^2\right] + E\left[\left|D(e^{j\omega})\right|^2\right] \tag{26}$$

Due to the clean speech signal is short-time stationary and also the noise. After the Fourier transform the statistical characteristics of the signals will not be changed. $X(e^{j\omega})$, $S(e^{j\omega})$ and $D(e^{j\omega})$ are all stationary in a frame. Therefore, in one frame, we can use a single value instead of the mean.

$$\left|X(e^{j\omega})\right|^2 = \left|S(e^{j\omega})\right|^2 + \left|D(e^{j\omega})\right|^2 \Rightarrow P_X(\omega) = P_S(\omega) + P_D(\omega) \tag{27}$$

Where

$$P_X(\omega) = \frac{1}{N}\left|X(e^{j\omega})\right|^2 \tag{28}$$

Due to the power spectrum of the noise dose not change no matter before or after the sound appears, we use the estimation of the noise $d'(n)$ to estimate $P_D(\omega)$ which is the power spectrum of the noise.

$$P_S'(\omega) = P_X(\omega) - P_D'(\omega) \tag{29}$$

So that

$$\left|S'(e^{j\omega})\right|^2 = \left|X(e^{j\omega})\right|^2 - \left|D'(e^{j\omega})\right|^2 \tag{30}$$

After the extraction of a root, we will get:

$$\left|S'(e^{j\omega})\right| e^{j\arg(X(e^{j\omega}))} \leftrightarrow s'(n) \tag{31}$$

In general, there will be negative power when we do the spectral subtraction. Using equation (14) to avoid the problem:

$$P_S'(\omega) = \begin{cases} P_X(\omega) - P_D'(\omega) & P_X(\omega) \geq P_D'(\omega) \\ 0 & P_X(\omega) < P_D'(\omega) \end{cases} \tag{32}$$

The method of using the $d'(n)$ to estimate the $\left|D'(e^{j\omega})\right|$ will lead to an issue, that there will leave a lot of noise if we reduce the amplitude spectrum directly:

$$\left|X(e^{j\omega})\right| = \left|S(e^{j\omega})\right| + \left|D(e^{j\omega})\right| - \left|D'(e^{j\omega})\right| \tag{33}$$

VI. Experiment Result

A. General parameters setting

In this part, general parameters are set up for future use, please see Appendix for details:

1) *Parameters*
   Sampling rate, sound speed in air, number of microphone and sources are specified. In addition, number of samples is calculated based on sampling rate and frame numbers.
2) *Audio and time indexes*
   Waves are read by audio related function. In addition, time indexes are calculated by sampling rate.

Note that waves should contain 192000×3 samples.

B. Calculate direction of 3 audio sources and generalize parameters

Repeat to specify spacing to apply different scene. Since only two pieces of audio are used, input of *BSS Locate* takes the mixture signals. Then, apply formula (4) to calculate angles, which is named by *theta*.

As a result, *tdoa* vector is round up by (meters)

$$[-7.2076 \times 10^{-4}, 7.2076 \times 10^{-4}, 1.2958 \times 10^{-4}]$$

In addition, *theta vector* is round up by (degrees)

$$[-81.4509, 81.4509, 10.2403]$$

Note: unit of *tdoa* vector is meter and *theta* vector is degrees. Baseline of 0 degree is the vertical line in central.

C. Beamforming Filtering

In this part I will show you the implementation procedure and the result of beamforming step by step based on the principle I have introduced in part IV.

Step 1:load three speech signals from the microphone array

Matlab code:

```
y1=wavread('mixture1.wav');
y2=wavread('mixture2.wav');
y3=wavread('mixture3.wav');
```

These three speech signals are collected by three microphones, they are all sampled at a frequency of 16000Hz, after this operation y1, y2 and y3 all contains 192000 samples.

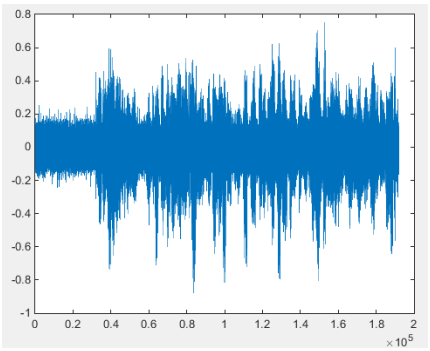


Figure 9 speech signal sampled at 16000Hz

As the figure 9 shown, it is the original signal sampled at 16000Hz in time domain, the first two second is the white noise.

Step 2:take the Fourier transform operation to these three speech signal to get $Y(w)$:

Matlab code:

```
Y1=fftshift(fft(y1'));
Y2=fftshift(fft(y2'));
Y3=fftshift(fft(y3'));
V=[Y1;Y2;Y3];
```

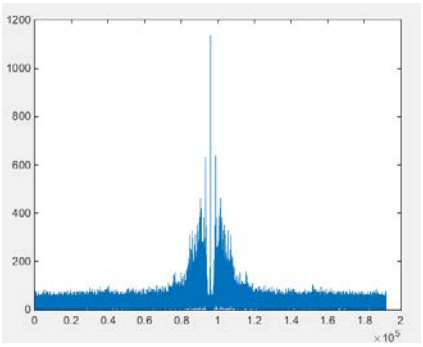


Figure 10 the spectrum of speech signal

As the figure 10 shown, it's the spectrum of the original speech signal after the Fourier transform. Due to the Fourier transform operation in matlab will reverse the order of the first and the latter part of the spectrum, we use *fftshift(fft(*))*to implement this step.

If we use *fft(*)* method, we will get the result shown in figure11.

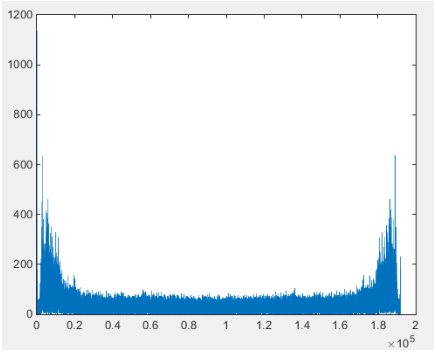


Figure 11 the spectrum used *fft(*)*method

Step 3: we need to build the steer vectors of the desired signal and the interferences.

Matlab code:

```
v=[1;exp(-i*t0*2*pi*f*cos(signal));exp(-i*2*t0*2*
    pi*f*cos(signal))];
I1=[1;exp(-i*t0*2*pi*f*cos(IR1));exp(-i*2*t0*2*pi
    *f*cos(IR1))];
I2=[1;exp(-i*t0*2*pi*f*cos(IR2));exp(-i*2*t0*2*pi
    *f*cos(IR2))];
```

The range of the variance f is from -8000 to 8000 Hz and V is the steering vector of the desired signal, the I1 and I2 are the steering vectors of interferences.

Step 4: build the filter $H(w)$ which is used to reconstruct the desired signal

Matlab code:

```
Sn=(1-a)*I+a*(I1*I1'+I2*I2');
WW=v'*inv(Sn)*v;
W=inv(Sn)*v/WW;
H=W';
```

These part of codes based on the equation (13)-(16)

Step 5: reconstruct the desired signal

Matlab code:

```
B(k)=H*V(:,k);
k=k+1;
```

We take a loop in this part which is limited by *k*, and the range of *k* is from 1 to 192000. This is a method to do the reconstruction on each sample, *B(k)* is the reconstructed sample.

$$Z(w)=\sum_{i=1}^{192000} H_i(w)Y_i(w) \tag{34}$$

Step 6 : we need to do the inverse Fourier transform to get the desired signal in time domain.

Matlab code:

```
Z=[B(96001:192000),B(1:192000)];
X=ifft(Z,192000);
x=real(X);
wavwrite(x,16000,'new_0.985.wav');
```

It is not hard to figure out, we change the order of the first part and the latter part back before the inverse Fourier transform and we don't use the *ifftshitf(ifft(*))*to implement this step. Instead of that, we use the method `Z=[B(96001:192000),B(1:192000)];`to change the order

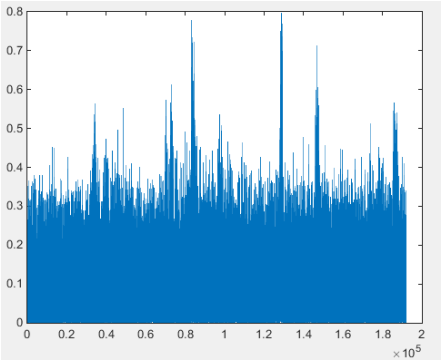


Figure 12 The desired signal after MVDR beamforming

There is an important parameter α means how close the interferences with the white noise. The range of the α is $[0,1)$. When we change the value of α, we find the interferences can be reduced if the value is greater than 0.97. In our project, we choose the value to be 0.985 which will make a good quality of the output.

VII. Conclusion and Future Work

It is challenge to get out of one channel signal from 3 sources while using microphone array system. Based on GCC-PHAT, it is useful to calculate TDOA for following procedures. In second stage, while it's a trade off between performance of noise deduction and directional interference elimination. In previous attempt, several different ways have been involved to try the best performance, while we do not subjectively compare these methods in terms of SNR and error variation. It's possible to test results and comparison. In future, we would like to have a deep look and research on TDOA of multiple microphones detection instead of only two.

Above all, the result is satisfied and English female audio is extracted effectively. Furthermore, it is audible to listen to the speech of English female while others do not have much impact on it. Furthermore, different methods of noise deduction are involved to pick the best one, while others are also remained.

References

[1] Acoustic source localization. (2014, October 24). *Wikipedia*. Retrieved November 25, 2014, from http://en.wikipedia.org/wiki/Acoustic_source_localization

[2] Phased Array System Toolbox. (n.d.). *Mathworks*. Retrieved November 25, 2014, from http://www.mathworks.com/products/phased-array/

[3] Documentation. (n.d.). *Mathworks*. Retrieved November 25, 2014, from http://www.mathworks.com/help/phased/examples/acoustic-beamforming-using-a-microphone-array.html

[4] TDOA. (n.d.). *Wikipedia*. Retrieved November 29, 2014, from http://zh.wikipedia.org/wiki/TDOA

[5] Multilateration (n.d.). *Wikipedia*. Retrieved November 29, 2014, from http://en.wikipedia.org/wiki/Multilateration#Measuring_the_time_difference_in_a_TDOA_system

[6] Chao Pan, Jingdong Chen, Jacob Benesty. Performance Study of the MVDR Beamformer as a Function of the Source Incidence Angle[J].2014.

[7] Blandin C, Ozerov A, Vincent E. Multi-source TDOA estimation in reverberant audio using angular spectra and clustering[J]. 2011.

Appendix

A. *Manual:*

Following code is made in MATLAB and I keep the format of highlight which is easy for readers to read. There are several pieces of scripts; main entrance has been declared as main function which use other function with proper parameters. If you intend to run it, please place them separately with function name and keep them in common directory, then run the main script. Note that the provided audio files should be also placed with same name in main function (as default). The phased array toolbox is based on new version of MATLAB, please install at least 2014a version to access new features. (keep all the dependent function scripts in case to work properly)

B. *Code:*

Main Function

```
%% Extraction a Target Speech Signal from 3-channel
Recordings
% Copyright (c) 2014-2015, Ottawa-Carleton Institute
for Electrical
% and Computer Engineering of University of Ottawa
% Author: Liu Cheng & Hongyu Zou
% Student Number: 7486632 & 7493642
% Contact Email: lchen156@uottawa.ca &
hzou047@uottawa.ca

% This script is based on bss_locate_spec.m written by
Charles Blandin
% and Emmanuel Vincent in 2011
% All work was modified by Liu Cheng & Hongyu Zou in Nov
2014 to
% apply related algorithms for actual application

%% General parameters setting
clc;
clear all;
close all;

fs = 16000;  % sample frequency is set to 16kHz
c = 343;  % propagation rate of sound in air (m/s)
nsrc = 3;  % indicate number of source of three
nmic = 3;  % similar, indicate number of microphones
nsample = fs * 12;  % total samples in all three mixture
audio
space = 0.25;  % space for each micphones is 25cm
t = 0:1/fs:(12-1/fs); % total time length in terms of
samples
% read three mixture audio into waves
waves = [audioread('mixture1.wav'),
audioread('mixture2.wav'),...
    audioread('mixture3.wav')];


%% Calculate direction of 3 audio sources and generalize
parameters
% Thanks for the work of Charles Blandin and Emmanuel
Vincent, related
% MATLAB scripts are used and modified for calculation
% 'stft_multi.m' is used as dependent function for
bss_locate_spec function
% by work of Charles Blandin and Emmanuel Vincent

% distance between microphone 1 and microphone 3
% for applying calculation of TDOA
d = 0.25;
% read mixture audio
```

```
waves_mixture23 =
[audioread('mixture2.wav'),audioread('mixture3.wav')
];
% calculate tdoa using parameters above
tdoa = bss_locate_spec(waves_mixture23, fs, d,
nsrc,'GCC-PHAT');
% store values of 3 angles into vector theta
theta = [asind(tdoa(1)*c/d), asind(tdoa(2)*c/d),
asind(tdoa(3)*c/d)];

%% Beamforming Filtering
% Start by scratch, followed by equation of realization
M=3; % Size of Rx Matrix
space=0.25; % distance between two sensors
t0=space/c;
a=0.985; % a parameter that controls the level of the
point source noise relative to that of the spatially
white noise
sita0=90-theta(3); % The direction of desired signal
sita1=90-theta(1); % The direction of interference1
sita2=90-theta(2); % The direction of interference2
y1=wavread('mixture1.wav');
y2=wavread('mixture2.wav');
y3=wavread('mixture3.wav');
Y1=fftshift(fft(y1'));
Y2=fftshift(fft(y2'));
Y3=fftshift(fft(y3'));
V=[Y1;Y2;Y3];
k=1;
I=eye(3); % white noise
signal=sita0*pi/180;
IR1=sita1*pi/180;
IR2=sita2*pi/180;
for f=-fs/2:fs/(192000-1):fs/2

v=[1;exp(-i*t0*2*pi*f*cos(signal));exp(-i*2*t0*2*pi*
f*cos(signal))]; % steer vector of desired signal

I1=[1;exp(-i*t0*2*pi*f*cos(IR1));exp(-i*2*t0*2*pi*f*
cos(IR1))]; % steer vector of interference1

I2=[1;exp(-i*t0*2*pi*f*cos(IR2));exp(-i*2*t0*2*pi*f*
cos(IR2))]; % steer vector of interference2
    Sn=(1-a)*I+a*(I1*I1'+I2*I2'); % psn
    WW=v'*inv(Sn)*v;
    W=inv(Sn)*v/WW;
    H=W';
    B(k)=H*V(:,k);
    k=k+1;
end
Z=[B(96001:192000),B(1:192000)];
X=ifft(Z,192000);
x=real(X);
wavwrite(x,16000,'Original_0.985.wav');

%% Parametric Spectral Subtraction
% directly use SSPARAB98 function which implement
Parametric Spectral
% Subtraction method
output=SSPARAB98(x,fs,2);
wavwrite(output,16000,'output.wav');

% output1=SSBoll79(x,fs,2);
% wavwrite(output1,16000,'output1.wav');
%
% output2=MMSECohen2004(x,fs,2);
% wavwrite(output2,16000,'output2.wav');
%
% output3=MMSESTSA85(x,fs,2);
% wavwrite(output3,16000,'output3.wav');
%
% output4=MMSESTSA84(x,fs,2);
% wavwrite(output4,16000,'output4.wav');
%
% output5=WienerScalart96(x,fs,2);
% wavwrite(output5,16000,'output5.wav');
%
% output6=SSBerouti79(x,fs,2);
% wavwrite(output6,16000,'output6.wav');
%
% output7=SSMultibandKamath02(x,fs,2);
% wavwrite(output7,16000,'output7.wav');
%
% output8=SSScalart96(x,fs,2);
% wavwrite(output8,16000,'output8.wav');
%
% output9=SSPARAB98(x,fs,2);
% wavwrite(output9,16000,'output9.wav');
```