

# BEAMFORMING BASED SPEECH ENHANCEMENT AND NOISE SUPPRESSION

Autumn Coe, Hilary Mogul, Yiting Zhang

University of Rochester, ECE472/AME272

## ABSTRACT

Speech enhancement is used in all kinds of applications including hearing aids, mobile phones, speech recognition, etc. There are lots of DSP algorithms for speech enhancement and noise cancellation, including filtering techniques, spectral restoration etc. This project focuses on superdirective beamforming for speech enhancement in a hearing aid application. Superdirective beamforming is advantageous in a hearing aid situation because it isolates a source from a specific direction, while still maintaining some semblance of directionality. Wiener postfiltering is then also used to reduce the SNR of the signal.

**Index Terms**— Beamforming, Speech Enhancement, Noise Suppression, Hearing Aids, Binaural Audio

## 1. INTRODUCTION

In a typical hearing aid situation, there is no knowledge of the desired noiseless signal, but there are multiple inputs of the noisy signal. Thus, an algorithm to enhance a single source in a noisy environment based on multiple inputs is desired. Lotter and Vary propose such an algorithm based on superdirective beamforming that allows enhancement of a source signal at a given steering angle [1]. For the purpose of this paper, we will refer to this algorithm as the Lotter and Vary system. This project implements their algorithm as a JUCE based VST plugin to create a real time and cross platform solution.

## 2. BACKGROUND INFORMATION

### 2.1. Beamforming and HRTF

Multiple microphones can be configured in an array to form a directional response or a beam pattern. When given an array with  $M$  microphones with known distances apart and a known steering angle, we can combine the noisy signals to get a constructive superposition towards the desired angle. Superdirective beamforming achieves heightened directionality by using HRTF functions to construct the constructive sum rather than the simple delay and sum model based only on the steering angle.

In this project, instead of microphone arrays, a binaural input-output system is used. In order to get the derivation of a binaural beamformer, an appropriate binaural

signal model is required. The HRTF database which comprises recordings of head-related impulse responses at 24 different interaural-polar azimuths is used to get the information about interaural time differences and amplitude differences between two ears.

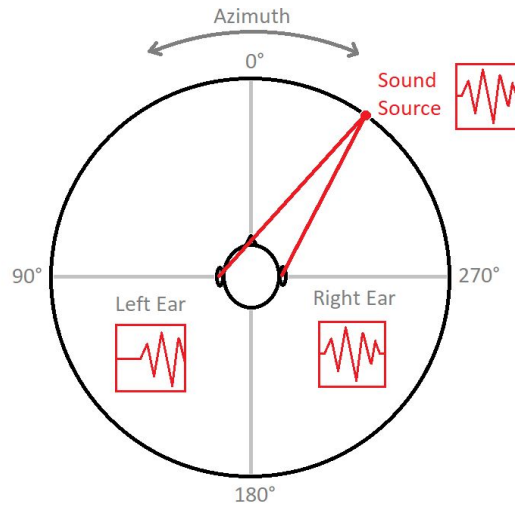


Figure 1: Dual channel binaural input from a mono source

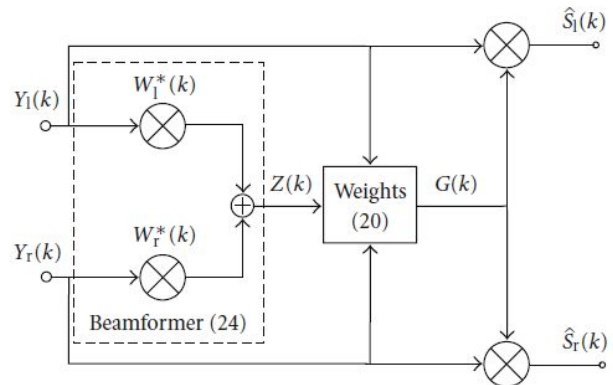


Figure 2: Superdirective binaural input-output beamformer. [1]

### 2.2. Wiener Filter

A Wiener Filter is a frequency domain application of an adaptive noise reduction filter. The ultimate goal of it would be to, using an observed signal  $x[n]$  and knowledge of a desired signal,  $s[n]$ , create the coefficients for a filter to process  $x[n]$  and create a minimum mean square error

(MMSE) output signal with regards to  $s[n]$ . To generate such a filter, some statistical knowledge of the power spectral density is calculated. Overall, the general transfer function of such a filter is:

$$(1) H(e^{j\omega}) = \frac{\Phi_{sx}(e^{j\omega})}{\Phi_{xx}(e^{j\omega})}$$

Where  $\Phi_{sx}$  represents the cross power spectral density, and  $\Phi_{xx}$  is the power spectral density of the observed signal. [2]

We tried implementing an LMS adaptive filter with the desired signal as a delayed version of the noisy signal. This worked well in Matlab with a sinusoidal desired signal, but did not work well on speech. Ultimately, we decided to implement the Wiener filter described above.

### 3. LOTTER AND VARY SYSTEM

Ultimately, the Lotter and Vary system can be broken down into three main components, all of which, when in series, calculate the frequency domain coefficients  $G(k)$  for both input channels. These components are the beamformer, the superdirective binaural filter, and the single channel adaptive Wiener postfilter.

The first of these, the beamformer, uses the known desired steering angle and the two inputs to generate a single channel recreation of the original signal, without any noise reduction. The next section,  $G_{super}(k)$ , covers using this beamformed signal to recreate the directionality of that source while still maintaining the isolation of the signal. The final section, the Wiener postfilter  $G_{post}(k)$ , is a single channel Wiener filter to reduce noise artifacts. In the next section, we will cover the mathematical derivation of each of these transfer functions.

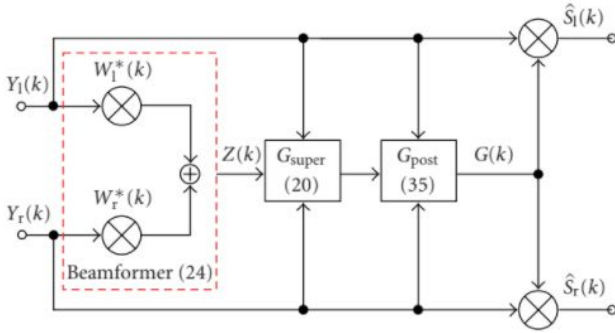


Figure 3: Superdirective input-output beamformer with postfiltering. [1]

## 4. LOTTER AND VARY SYSTEM MATHEMATICS

### 4.1. Beamformer

The necessary filters to create the beamformed signal,  $W_{L/R}(\theta_S, k)$ , can be calculated using the frequency domain

HRTF's of the steering angle  $D_{L/R}(\theta_S, k)$  in the following manipulation:

$$(2) W_{L/R}(\theta_S, k) = \frac{\Phi_{MM}^{-1}(k)D_{L/R}(\theta_S, k)}{D_{L/R}(\theta_S, k)^{-1}\Phi_{MM}^{-1}(k)D_{L/R}(\theta_S, k)}$$

Where  $\Phi_{MM}(k)$  represents the cross power spectral density matrix of the microphone array. Since we have only two microphones, this matrix can be simplified to:

$$(3) \Phi_{22}(k) = \begin{pmatrix} 1 & \Gamma(k) \\ \Gamma(k) & 1 \end{pmatrix}$$

Where  $\Gamma(k)$  corresponds to the coherence function, a frequency domain function to measure the correlation between the left and right channels [3]. For the Lotter and Vary system, this is approximated to:

$$(4) \Gamma(k) = \frac{\sum_{n=1}^N D_L(\theta_n, k)D_R(\theta_n, k)}{\sqrt{\sum_{n=1}^N |D_L(\theta_n, k)|^2} \sqrt{\sum_{n=1}^N |D_R(\theta_n, k)|^2}}$$

Where N corresponds to the number of steering angle indices. Since we were using precalculated HRTF's where the steering angle was adjusted every fifteen degrees, our N value was 24.

### 4.2. Binaural Superdirective Beamformer

This function is simply a transfer function based on the magnitude response of the beamformed signal and those of the original inputs:

$$(5) G_{super}(k) = \frac{|Z(k)|}{|Y_L(k)| + |Y_R(k)|}$$

### 4.3. Wiener Postfilter

Since we do not have any knowledge of the desired signal, the beamformed signal is used in its place. Lotter and Vary also make the distinction that the correlation of the noise in each input would be low, so a transfer function can be written as:

$$(6) H_{post}(k) = \frac{\Phi_{ZZ}(k)}{\frac{1}{M} \sum_{i=1}^M \Phi_{ii}(k)}$$

Where  $\Phi_{ii}$  represents the power spectral density of each microphone. Further, Lotter and Vary adapt this filter to use a frequency domain correction factor based on the angle, ultimately creating the following adaptive equation for calculating the coefficients for  $G_{post}(k)$  as a function of the steering angle  $\theta_S$ :

$$(7) G_{post}(\theta_S, k) = \frac{|D_L(\theta_S, k)|^2 |D_R(\theta_S, k)|^2}{(|D_L(\theta_S, k)| + |D_R(\theta_S, k)|)^2} \frac{|Z(k)|^2}{|Y_L(k)|^2 + |Y_R(k)|^2}$$

## 5. FACTORIZATION OF LOTTER AND VARY SYSTEM

To facilitate implementation, we factored the post beamformer processing into an adaptive section, which is based on the incoming signal, and a static section, which is based on the HRTF's, and thus could be serialized. All

values for  $G_{static}(k)$  and  $W_{L/R}(\theta_S, k)$  are calculated in Matlab and saved as .wav files.

$$(8) G_{adaptive}(k) = \frac{|Z(k)|^2}{(|Y_L(k)|^2 + |Y_R(k)|^2)(|Y_L(k)| + |Y_R(k)|)}$$

$$(9) G_{static}(\theta_S, k) = \frac{|D_L(\theta_S, k)|^2 |D_R(\theta_S, k)|^2}{(|D_L(\theta_S, k)| + |D_R(\theta_S, k)|)^2}$$

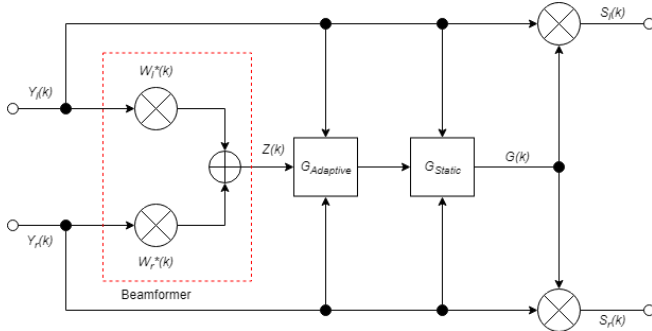


Figure 4: Superdirective input-output beamformer with postfiltering factored into adaptive and static phases.

## 6. IMPLEMENTATION

### 6.1. .wav File Calculation in Matlab

We calculated values for  $G_{static}(k)$  and  $W_{L/R}(\theta_S, k)$  in Matlab and saved them as .wav files. We load the HRTF database and extract all the left and right ear impulse responses from all 24 azimuths on the 0 degree elevation. Then we do calculations in DFT domain in order to get the beamformed signal and convert the signal back to time domain and export them as .wav files.

### 6.2 JUCE Implementation

JUCE is a cross platform framework for the development of audio applications and plugins. Its DSP module additionally provides the use of a library to handle frequency domain operations, including Fast Fourier Transforms and their inverses, and a Processor interface to apply filters to incoming audio blocks. The Convolution object is an implementation of this interface, and loads an impulse response from an (optionally stereo) audio file, and does the calculations in the frequency domain to accelerate computing when the impulse response is of a high order. The module also provides the means to chain multiple Processor objects together in an aptly named template called ProcessorChain [4].

Our implementation ultimately loads a respective audio file for each static component, including the beamformer transfer functions, based on a selected steering angle from the user into two separate convolution objects, the beamforming filters in one, and the static portion of the postfilter in another chain which also contains a separate Convolution object for the adaptive portion.

The incoming stereo signal is copied into a temporary buffer, then processed by the beamformer chain

and summed to create the beamformed signal. The FFT of this is then taken to gain  $Z(k)$ , and  $G_{adaptive}(k)$  is calculated using these coefficients. The IFFT of this filter is then taken and converted into an AudioBuffer object, then loaded into the second ProcessorChain. The original input signal is then processed against this chain, the transfer function of which is  $G_{adaptive}(k)G_{static}(k) = G(k)$ , the original desired signal.

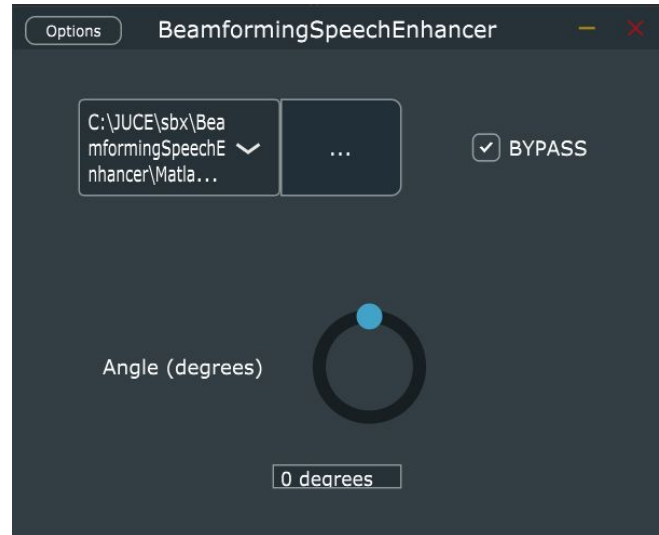


Figure 5: User interface of plug-in

### 6.3. Hardware

To simulate hearing aids, the Roland binaural in-ear monitors are used. These are earbuds equipped with microphones used for binaural recording. The earbuds are routed through a Zoom H2 handy recorder into a laptop running the VST plugin.



Figure 6: Hardware used

## 7. SUBJECTIVE RESULTS

To subjectively compare the filtered signal to the raw signal, we created added a bypass to the GUI that allows the user to turn the filtering on and off. We created a noisy environment to perform qualitative tests by generating white noise from multiple noise sources. With one person using the system, someone else spoke in a soft voice while the user experimented with turning the bypass on and off. We determined there was a noticeable improvement in speech comprehension with the system running.

## 8. FUTURE WORK

First of all, we can calculate HRTF instead of using the database to refine angular resolution and allow customization. The information extracted from HRTF database delivers fixed HRTFs. However, in real life, HRTFs varies depending on the radius of the head and the position of the microphones. Also, HRTF database has 24 fixed azimuths with 15 degree apart from each other on 0 degree elevation. Calculating HRTF from model which approximates ITDs and IIDs can definitely increase the resolution.

Secondly, we can include head tracking to maintain source while source angle changes. In our system, the user has to approximate the angle to the source, and input the approximated angle to the system.

Finally, we should look into keeping the transfer functions in the frequency domain, as we approximate the actual calculated impulse responses into essentially FIR filters when we take the IFFT of them. More development of JUCE's DSP module would be necessary for this, as it only currently supports biquad filters as IIR filters.

## 9. APPENDIX

All code and generated impulse responses can be found at:  
<https://github.com/hkmogul/BeamformingSpeechEnhancer>

## 10. REFERENCES

- [1] T. Lotter and P. Vary, "Dual-Channel Speech Enhancement by Superdirective Beamforming," *EURASIP Journal on Advances in Signal Processing*, vol. 2006, no. 1, 2006.
- [2] "3.11. The Wiener Filter," *Digital Signal Processing 0.0 Documentation*. [Online] Available: [http://dsp-nbsphinx.readthedocs.io/en/nbsphinx-experiment/random\\_signals\\_LTI\\_systems/wiener\\_filter.html](http://dsp-nbsphinx.readthedocs.io/en/nbsphinx-experiment/random_signals_LTI_systems/wiener_filter.html), [Accessed: 30-Apr-2018]
- [3] "Coherence Function," *Mathematics of the DFT*. [Online]. Available: [https://www.dsprelated.com/freebooks/mdft/Coherence\\_Function.html](https://www.dsprelated.com/freebooks/mdft/Coherence_Function.html), [Accessed 30-Apr-2018]
- [4] "JUCE," *JUCE*. [Online]. Available: <https://juce.com/>. [Accessed: 28-Apr-2018].