Delft University of Technology

Array Processing
EE4715

Speech Enhancement

Authors:
Chaufang Lin (5466091) Maxmillan Ries (5504066)

August 23, 2022

1 Introduction

For this project, we were tasked with implementing a speech enhancement system for far-end noise reduction.

2 Data Generation

This multi-microphone speech enhancement system is designed for far-end noise reduction. Five speech signals are given, including two clean speeches, a babble noise, an artificial nonstationary noise and a stationary speech shaped noise. Concurrently, five sets of multi-microphone impulse responses are also provided for target and interference. Each set contains the impulse responses from a source location to the 4 microphone locations. Each of these 5 sets is convolved with a sound signal to model the received signal at each of the microphone locations. The target is set to be the clean speech 1 signal, and the others are all considered to be interference.
The five audio files given have different length, so after convolution, the microphone data can be clipped or padded to ensure all received signals have the same duration. In our case, we found that padding all files to the longest signal resulted in 3 times the longer files, and instead cut all fits down to the target signal.
The target speech and the noisy speech is shown as Fig.1.

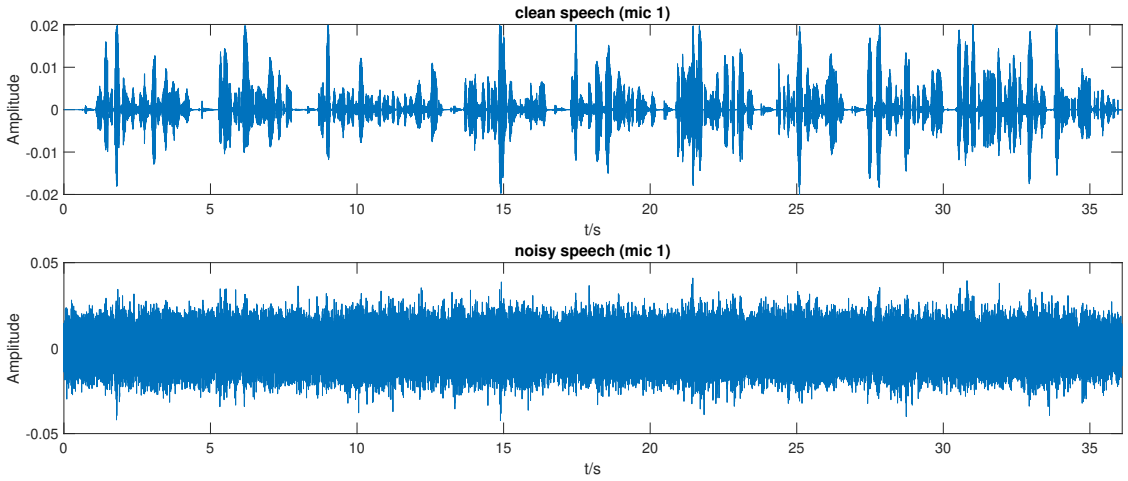

Figure 1: Clean speech 1 and noisy speech on the first microphone

3 Short-Time Fourier Transform

Signals are framed by a sliding window, and the Discrete Fourier Transform of the windowed data is calculated separately. Instead of using the Matlab function, we implemented STFT by ourselves. Here, we chose a Hamming Window of length 400, which has the same length as the impulse response, and 25% hop size. The STFT is combined into the overlap-add method.

4 Noise Correlation Estimation

There are several ways to do the noise correlation estimation. The main method taught in this course consists of using a moving average R_n estimation. There are two cases when estimating

R_n, described below:

- **Target is not present**: When the target is not present, we want to update the R_n estimate with the measured values of that window (per frequency bin).
- **Target is present**: When the target is present, we do not wish to update R_n, as it would corrupt it to include the signal we are interested, resulting in our beamforming removing what we are trying to look for (at the very least distort it).

In our case, we took a simpler approach of using an initial estimate, by taking the first 0.5 seconds of the noisy input where the individual does not speak, and creating the R_n estimate. Specifically, the estimate is created by calculating $R_n(k,l)$ for each frequency bin per segment, and averaging it per frequency bin.
While this estimate is perhaps not as accurate, we found it sufficient to show our understanding, and show we understood what could be done to improve upon it. The main advantage the moving average estimate has is the "up-to-date" estimate it provides, notably if a sudden spike in high frequency noise occurs. Since our method assumes the noise is generally the same across the entire signal, it is a less accurate estimation.

5 Estimate the ATF

Estimating the Acousting Transfer Function (ATF) is something we struggled with the most. We decided to study and try estimating the ATF assuming the abscence of the noise, spatially white noise and colored noise. Below we show our understanding of the subject matter:

5.1 No Noise

Given the expression for $R_x = R_s + R_n$, in the absence of noise, the expression simplifies itself to $R_x = R_s$. In order to estimate the ATF, as R_s or an estimate of it is needed, the problem trivializes itself.
By taking the eigenvalue decomposition of R_s (in this specific case R_x), $R_s = U\Lambda U^H$, one finds that the eigenvalues Λ correspond to a matrix:

$$\begin{bmatrix} \sigma_s^2(k,l)||a(k,l)||^2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

The eigenvector corresponding to the non-zero eigenvalue is hence the scaled ATF $u = a(k,l)/||a(k,l)||$.

5.2 White Noise

When there is spatially white noise, R_x is no longer simply equal to R_s, but is instead equal to $R_s + R_n$. When performing an eigenvalue decomposition, the following result is not longer trivial to solve: $R_x = U(\Lambda + \sigma_n^2 I)U^H$.
However, by taking the principle eigenvector (which is the same as the previous section with the addition of $\sigma_n^2 I$ noise), and subtracting $\sigma_n^2 I$ noise (which can be taken from the other eigenvalues), one can retrieve an estimate of R_s, and hence an estimate for the ATF.

5.3 Non-White Noise

The ATF Estimation based on non-white noise corresponds can be solved by either pre-whitening the noise or using the Generalised Eigenvalue Decomposition (GEVD). We chose to implement an estimation of the ATF using the GEVD.
Using the GEVD theorem, we find that the generalised eigenvalues and eigenvectors of R_s and R_n correspond to the eigenvalues and eigenvectors of the matrix $R_n^{-1}R_s$. While we do not have R_s as we aim to estimate it, we can use R_x instead, resulting in the following formulation: $EVD(R_n^{-1}R_x) = EVD(R_n^{-1}R_s + R_n^{-1}R_n) = EVD(R_n^{-1}R_s + I)$.
By taking the eigenvector corresponding to the principle eigenvalue and computing it's inverse Hermitian (after subtracting I), one finds the EVD of R_s, and hence the estimated ATF.

Under normal circumstances, we think the GEVD would allow a perfect estimation of R_s, however, as we estimate R_n and use it for the GEVD, the estimated R_s is never perfect. With the ATF in hand however, we can compute some beamformers to try and recover our target signal.

6 Beamforming

For our own education, we decided to make three presented beamformers, the Delay & Sum, the MVDR and the Multi-Channel Wiener beamformers.

6.1 Delay & Sum Beamformer

The Delay & Sum beamformer is very simple and does not take into consideration the noise. This means that, while the original signal is reconstructed, it will still be warped by the noise which is not taken into account. This can be visualized in the plot below:

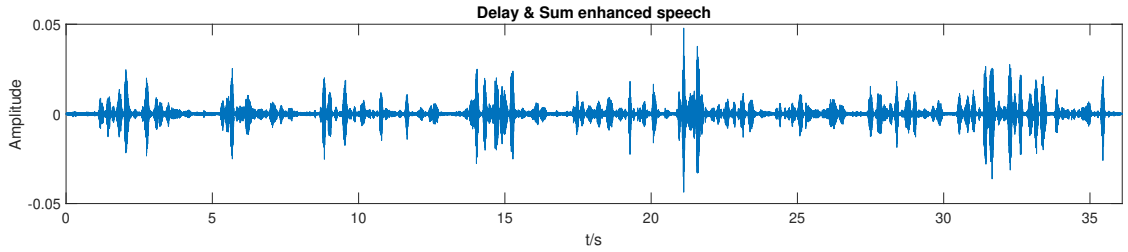

Figure 2: Recovered speech after Delay & Sum beamformer

6.2 MVDR Beamformer

The Minimum Variance Distortionless Response (MVDR) beamformer ensures that there is no change in the direction of the original signal, while trying to minimize the variance of signals coming from all other directions. While we do not fully understand the derivations to get to the MVDR beamformer, we do find that it reconstructs the signals better than the Delay & Sum beamformer, as shown below:

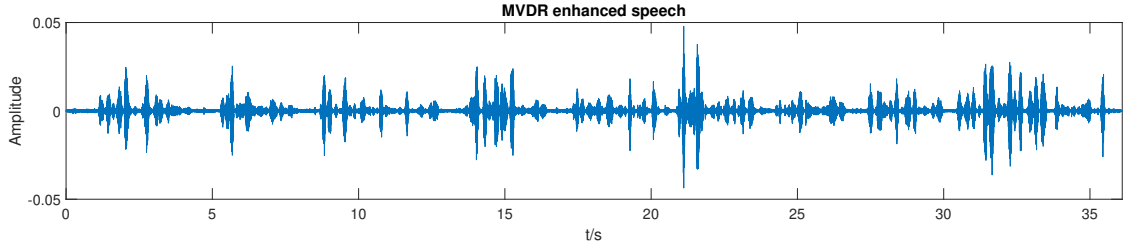

Figure 3: Recovered speech after MVDR beamformer

It should be noted, that in the presence of spatially uncorrelated noise, the MVDR behaves like the Delay & Sum beamformer.

6.3 Multi-Channel Wiener Filter

In general, the Wiener filter is a combination of the MVDR beamformer as well as a Single Channel Wiener filter. The Single Channel Wiener filter computes an output signal estimate based on the correlation estimate of the noise. The results of applying this filter can be seen below:

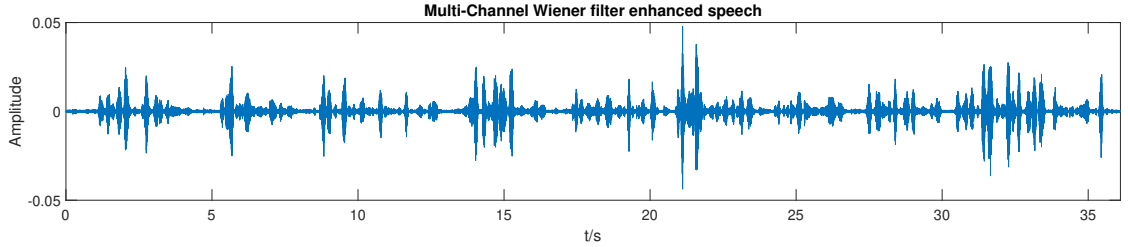

Figure 4: Recovered speech after Multi-Channel Wiener Filter

Generally comparing the plots, the immediate observation is that the Delay & Sum and the MVDR beamformer appear to have identical reconstructed signals. This is normally the case when the noise is spatially uncorrelated. However, our estimated noise correlation R_n is not a diagonal matrix, indicating a spatial correlation between the different noise sources. When looking into the data further, while the beamformers w_{das} and w_{mvdr} are entirely different, the multiplication output with the fft segment is the same: $w_{das} * fft_{seg} = w_{mvdr} * fft_{seg}$, which is something we found very odd, but could not resolve.
Comparing the two with the Multi-Channel Wiener Filter, we find that the Wiener filter has a cleaner signal, in that less noise seems to be present. However, in order to evaluate the difference between the performance of each beamformer, we need to find an appropriate evaluation metric. In the following section, we provide such an evaluation.

7 Evaluation

7.1 Speech Quality

Perceptual Evaluation of Speech Quality (PESQ) is a family of standards comprising a test methodology for automated assessment of the speech quality as experienced by a user of a telephony system. The perceptual model of the PESQ measure is used to calculate the distance (PESQ score) between the original signal $X(t)$ and the degraded signal $Y(t)$, which is the result

of $X(t)$ passing through a communication system. The range of the PESQ score is –0.5 to 4.5, although for most cases the output range will be a MOS-like score, i.e., a score between 1.0 and 4.5. The higher the PESQ score is, the better the speech quality is.
The reason for choosing PESQ as our metric is that the effects of loudness loss, delay, sidetone, echo, and other impairments related to twoway interaction are not reflected in PESQ scores. In this assignment, it is a far-end environment, so loudness can't be perfectly recovered, so here we use a metric that doesn't take loudness into account.

	Before	Delay & Sum	MVDR	Multi-Channel Wiener Filter
PESQ	0.516	1.783	1.783	1.783

Viewing the results, the far-end speech enhancement system improves the speech quality, but only by what seems to be a small margin.

7.2 Speech Intelligibility

Short-Time Objective Intelligibility (STOI) is an important criterion to measure speech intelligibility. As for one word in a speech signal, there are only two cases: understood and misunderstood. From this perspective, we can say the intelligibility is binary, so the range of STOI is $[0, 1]$. STOI represents the percentage of how a word could be understood. The bigger STOI is, the higher the intelligibility is.

	Before	Delay & Sum	MVDR	Multi-Channel Wiener Filter
STOI	0.404	0.734	0.734	0.734

Unlike the results from the PESQ speech quality evaluation, speech intelligibility seems to have be improved to a decent level. Roughly speaking, it means more than 73% of the speech can be understood, compared to a previous 40%. This is enough for us to consider our far end noise reduction an initial success, though we are fairly confident that both the speech quality and intelligibility can be further improved by taking a better estimate of R_n.