Speech Enhancement

Group 2: Guillermo de Dios & Junchi Liu

October 31, 2023

1 Introduction of the problem

We often face less than perfect signal quality when using speech technology in real environments. For instance, if two people are making a phone call, typically some unavoidable audio signal will be picked by microphones such as the sound of cars passing by and voices from other people talking near the speaker. These effects alter the intended speech signal, making it less pleasant, harder to comprehend, or even less clear at the receiving end.

In addition, considering the interaction between human and machine especially voice assistant in AI devices, it is more important to apply the filter to noise speech signals as AI speech recognition system is not developed enough. Speech enhancement could reduce the noise signal for non-stationary noise therefore avoiding misunderstanding of voice control systems.

Speech enhancement techniques aim to minimize these distortions, enhancing the speech quality, easing comprehension, and making it more enjoyable for the listener. From the signal processing perspective, speech enhancement tech is used to extract clean speech signal or in other words, improve signal-to-noise ratio(SNR).

2 Motivation

Speech enhancement has a wide range of applications. In modern society, speech often occurs in noisy environments, like crowded cafes or busy streets. Speech enhancement helps in isolating and amplifying the speech signal while reducing unwanted background noise, making communication more effective. For individuals with hearing impairments, speech enhancement technologies can significantly improve their ability to understand spoken language. This promotes inclusivity and equal access to information and communication.

Besides, in telemedicine, where doctors and patients communicate remotely, speech enhancement is essential for clear and accurate diagnosis and treatment discussions. As we discussed above, speech enhancement is a fundamental component of voice recognition systems. Clear speech input is essential for accurate transcription and command execution in voice-activated devices and applications.

The algorithms of speech enhancement for noise reduction can be categorized into three fundamental classes: filtering techniques, spectral restoration, and model-based methods.[1]

- Filtering techniques: [2, 3, 4]
  - Spectral Subtraction Method
  - Wiener Filter
  - Signal subspace approach (SSA)
- Spectral Restoration [5]
  - Minimum Mean-Square-Error Short-Time Spectral Amplitude Estimator (MMSE-STSA)
- model-based methods

Spectral subtraction is one of the traditional methods used for enhancing speech degraded by additive stationary background noise. [6] It falls into the non-parametric category, requiring an estimation of the noise spectrum. However, it also does not attenuate noise sufficiently during the silence period.

Wiener filtering is a type of minimum mean square error filter that is based on a statistical model, aiming to minimize the mean square error between the output signal and the original clean signal. It utilizes estimates of the power spectral density of the signal and the power spectral density of the noise to calculate the optimal filter.

Here based on the audio files, we present the Wiener Filtering and Spectral Subtraction for speech enhancement and compare the results for two methods. [3]

3 Proposed methodology

Our general approach to the project can be divided into two parts. First, we estimated the power spectral density(PSD) of noise $\hat{P}_N(\omega)$ from the audio that contains both the speech and the noise. After that, we use this estimation to reduce the noise of our speech signal, trying to make it as close as possible to the speech without noise, so that it is easier to understand what the speaker is saying.

3.1 Analysis of audio files

Since we were provided with two audio files, one consisting of clean speech audio lasting approximately 36 seconds, and another containing a noise signal lasting about 37 seconds, our initial step was to equalize the number of samples and generate the noisy audio signal. While reading the signal to a matrix variable in MATLAB, we can find that the sample frequency is $16KHz$ for both signals. We determine both signals' lengths are N, which is equal to the length of clean speech signal. To create the audio file where the noise and the clean speech signal are mixed we first scale the noise in a way that it matches the desired SNR. We initially set the SNR to 5 dB, which means that we are assuming that the intensity of speech is higher than the intensity of the noise. In addition,we are concerned that the noise signal, which is audio $N(n)$ from other individuals talking, is obviously independent of the speech signal $S(n)$.

3.2 Estimating power spectral density(PSD)

PSD can be estimated by computing the magnitude squared of its DFT. In MATLAB, this is achieved by simply using the command *fft()*. The procedure for estimating PSD is illustrated in the workflow chart below:

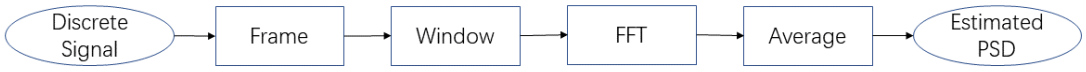


Figure 1: PSD Estimation Module

Each signal's PSD we used in the project was computed under this module. The number of samples L of the frames used to compute the DFT are equal to 1024. This value of L is a power of 2 that in time domain (with $fs = 16$ KHz) is equal to 64 ms which we are assuming is small enough to consider that in that period of time our signals are stationary. Furthermore, with this value of L we are getting a good enough resolution of our PSD. If we had chosen a higher value of L, the frames would have been larger and we would have a better resolution, but also a higher variance and a longer time period of our frame. Once we have selected L we can determine the number of segments (K). Moreover, choice of window used for non-parametric estimation is also essential. We are going to compare the results using different windows, which will give us lower side lobes at the expense of some resolution.

In practice, we are assigned only the audio file with speech and noise mixed, which means that we cannot compute the PSD of noise directly. However, it was noticed that at about the first 1 second of audio, there is nobody talking and it just plays some background noise. this is a characteristic shared by many audios with background noise. For example, when you receive a phone call, it is unusual to start talking instantly. By frame segmentation, we chose the segments corresponding to the first 1.2 second samples, so there is enough time to estimate the noise PSD. Then we applied it to the PSD

estimating module and got $\hat{P}_N(\omega)$. To make the estimated result is applicable, we simulate the PSD of noise from "Babble Noise" file and compare $\hat{P}_N(\omega)$ and $P_N(\omega)$:

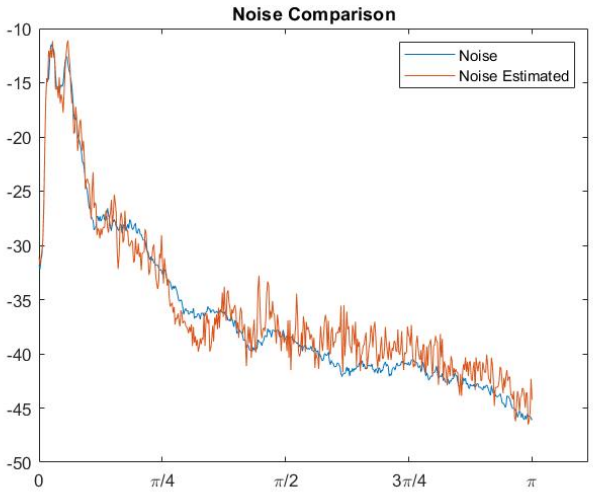


Figure 2: Estimated Noise and Real Noise

From the graph, we can conclude that variance is slightly higher in the estimation of the noise. Since the estimated noise from noisy audio does not have so much difference with real babble noise, we can use $\hat{P}_N(\omega)$ for our speech enhancement methods. The workflow of the calculation of the estimation of the PSD is shown in the graph below:

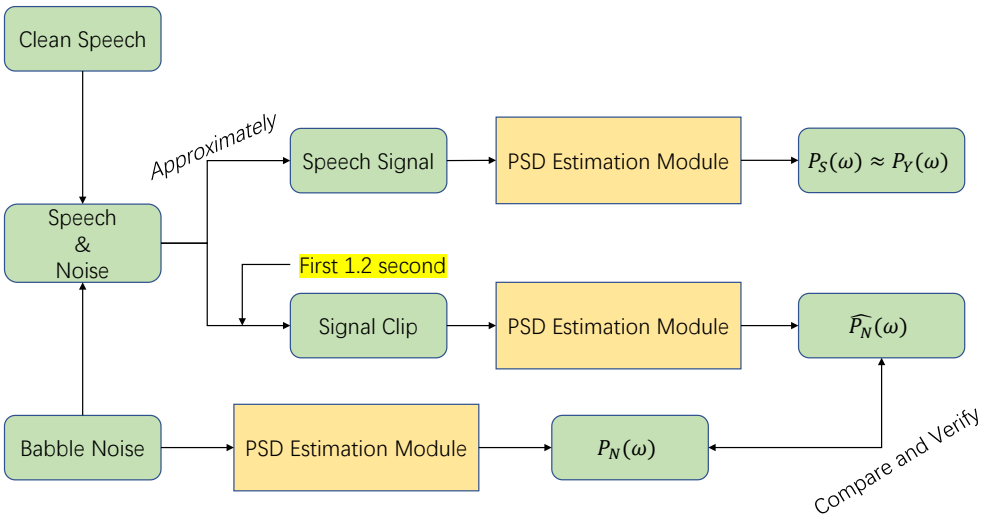


Figure 3: Workflow of PSD Computing from Audio Signal

3.3 Wiener filter for speech enhancement

In the wiener filter approach we tried to minimize the mean square error between the output signal and the desired signal. In our case, the desired signal is the clean speech and the input of our filter is going to be the desired signal with additive noise. In frequency domain, the Wiener-Hopf equations

for filtering become:

$$Fh(\omega) = \frac{P_{SS}(\omega)}{P_{YY}(\omega)} = \frac{P_{SS}(\omega)}{P_{NN}(\omega) + P_{SS}(\omega)} \tag{1}$$

We are going to apply this filter in frequency domain to the PSD of every frame of our noisy speech. To model this filter and use it as a filter function in our project, we are going to approximate the PSD of speech signal to the PSD of the mixed-signal because initially we set the SNR of "speech with noise" signal to 5 dB, which means that the power of speech signal is much higher than the babble noise. Thus equation 1 is rewritten:

$$Fh(\omega) = \frac{P_Y(\omega)}{\hat{P_N}(\omega) + P_Y(\omega)} \tag{2}$$

The final step to get the PSD of enhanced speech signal is to multiply the original signal PSD by the frequency response of the wiener filter:

$$\hat{P_S}(\omega) = P_Y(\omega) \times Fh(\omega) \tag{3}$$

After solving the problem in frequency domain, we need to use the Inverse fast Fourier transform (IFFT) to check the filtered signal $S(n)$ in time domain. To succeed in the process of the IFFT we saved the information of the phase before computing the magnitude square of the DFT of the frame to get the PSD of the frame. The process of wiener filter and IFFT can be concluded into one work flow similar to the PSD estimation:

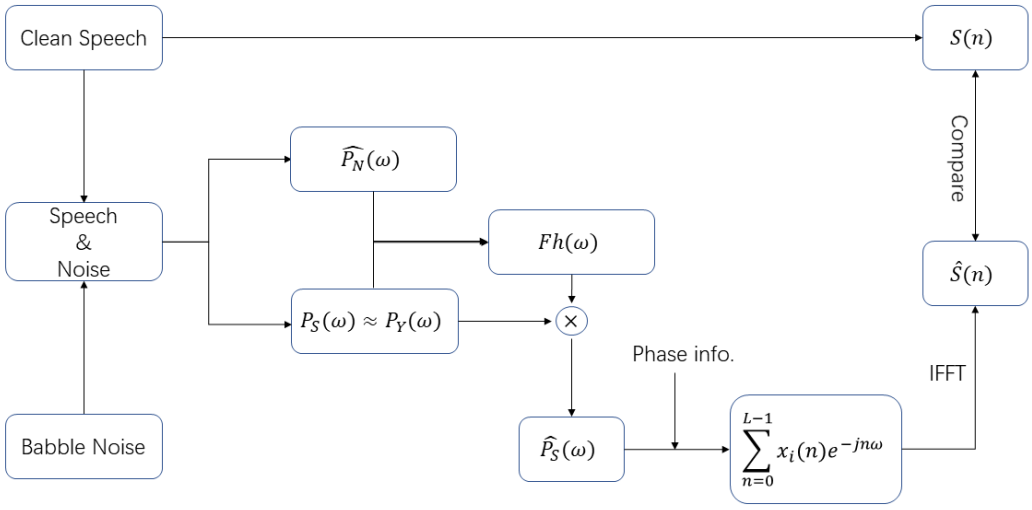


Figure 4: Workflow of Wiener Filter

3.4 Spectral subtraction for speech enhancement

Since the babble is "smooth", which means slowly varying within the analysis window, we consider use spectral to model the filter and enhance the signal.

Spectral subtraction is a straightforward method that subtracts a scaled version of the estimated noise spectrum from the noisy signal's spectrum. The scaling factor is determined based on the assumed or estimated Signal-to-Noise Ratio (SNR). We have chosen a scaling factor of 3% based on empirical observations. This is because the estimation of the noise PSD is similar to the estimation of the clean speech PSD as is shown in the following figure:

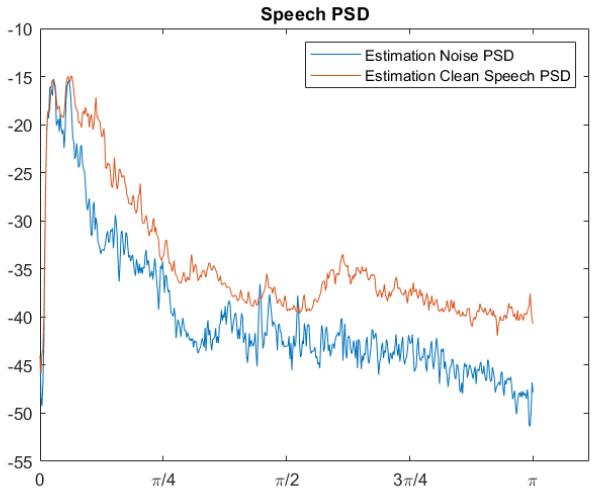


Figure 5: Comparison of estimation of noise PSD with estimation speech PSD

From the graph, we can conclude that the both PSD share the same values in the lowest frequencies, which means that if we subtract the whole noise PSD to our noisy speech PSD, we are going to loose a lot of information of the original speech signal. Thanks to the scaling factor, we only subtract 3% of the intensity of the estimation of the noise PSD from our PSD of the noisy speech. Compared to the Wiener filter, it is relatively easy to implement. Because it doesn't involve complex statistical calculations and is computationally less intensive. Moreover, another difference between two methods is phase information. Spectral subtraction typically ignores phase information, reconstructing the phase of the enhanced signal based on the original noisy signal. This may lead to phase-related artifacts in some cases.

4 Simulation process and results

In the simulation, we primarily investigated the following issues:

1. **Effect of Window Functions:** We applied four different window functions to estimate both power spectral densities of the noise and of the speech signal and used them in the Wiener filtering process. We examined how each window function affected the original signal and observed its influence after filtering.

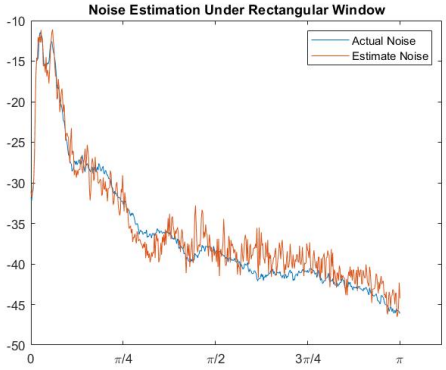
2. **Performance of Two Different Filtering Approaches:** We evaluated the performance of two distinct filtering methods by comparing the Signal-to-Noise Ratio (SNR) difference before and after filtering. Additionally, the results of each filtering process were transformed into the time domain and subjected to visualization for ease of comparison.

5 Discussion

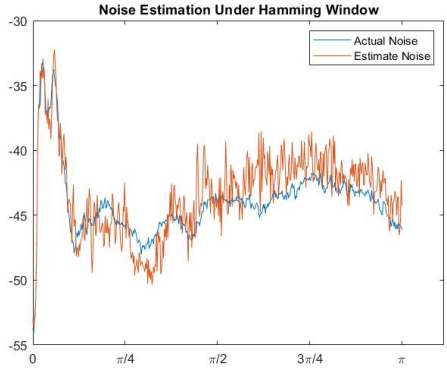
5.1 Effect of Window

While estimating the PSD of noise and the PSD of the speech signal, we can not only apply the rectangular window to the non-parametric estimation. Other widely used windows such as Hamming window, Bartlett window, and Blackman window may also work on PSD estimation in our case. Therefore, we tried these three windows and compared the result of enhanced speech signal with the one applying rectangular window. The following figure shows the comparison between the estimation of the noise PSD using different windows:

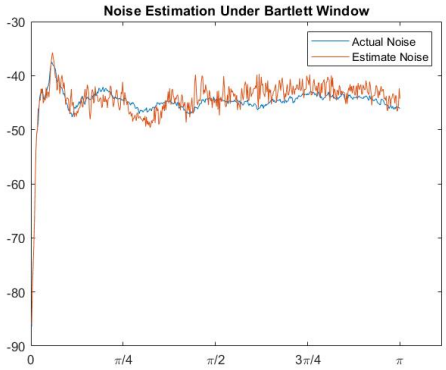
The window gives a way to trade-off between the width of the main lobe and the side-lobe amplitude. From the graphic obtained using a rectangular window, we can conclude that there are two main lobes in the lower frequencies. The rectangular window exhibits the best amplitude resolution. However, the



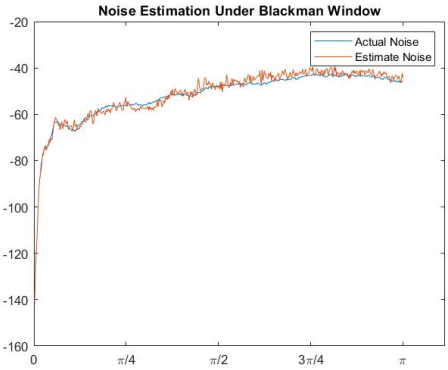
(a) Rectangular



(b) Hamming



(c) Bartlett



(d) Blackman

Figure 6: Noise Estimation Effect of Window

side-lobe level goes down from the rectangular window to the Blackman window. Using the Blackman window, we cannot see the main lobes due to the loss of spectral resolution, which means that we are losing essential information from our PSD, even though the side-lobe level is lower. With the hamming window we obtain more concentrated main lobes and better side-lobe suppression compared to the Bartlett window.

So we continued to verify the SNR effect of the four windows under the wiener filter where we assigned the initial SNR is $5dB$. Since we want to check the SNR enhancement performance, the difference in SNR before and after filtering is particularly important. Thus we drew a table with SNR difference below:

	Rectangular	Hamming	Bartlett	Blackman
SNR Difference	1.0468 dB	3.9051 dB	1.9531 dB	-0.1579 dB

From this table, we can conclude that the Hamming window offers the best results in terms of SNR difference. This is because the Hamming window achieves a good balance between main lobe concentration and side lobe suppression.

5.2 Wiener Filter and Spectral Subtraction

These two methods are effective on speech signal enhancement especially if we assume that both audio signal are wide-sense stationary(WSS) processes. Their advantages and disadvantages are discussed in section 4 generally so here we only present the signal result and discuss their performance in our case. We have used the hamming window in both methods, as it has given the best results in terms of SNR difference.

After wiener filter, enhanced speech signal, clean speech signal, and relevant noise spectrum in time domain are plotted as follows:

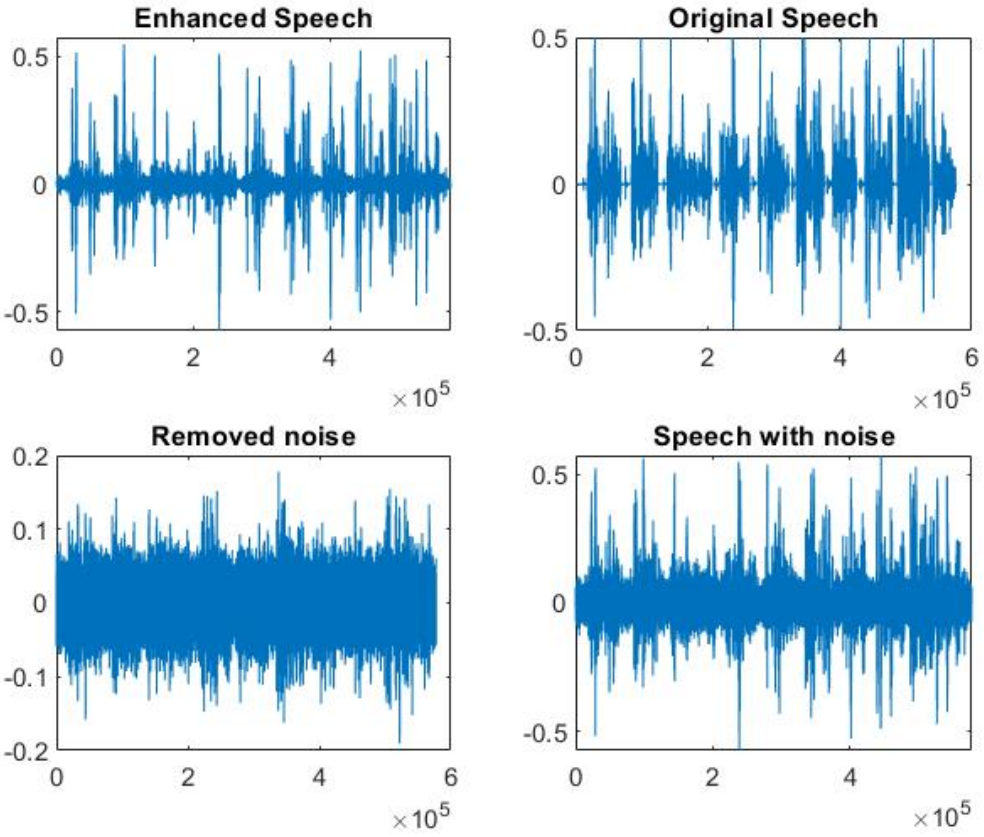


Figure 7: Result of Wiener Filter

From the result figures, we achieve the goal of enhancing speech signal. But unavoidably, some speech signal was filtered as noise in our model, which makes the time-domain spectrum of enhanced speech signal sharper than clean speech. In the audio file, this is reflected as a slight decrease in speech intensity after filtering

And result from spectral subtraction is shown as follows:

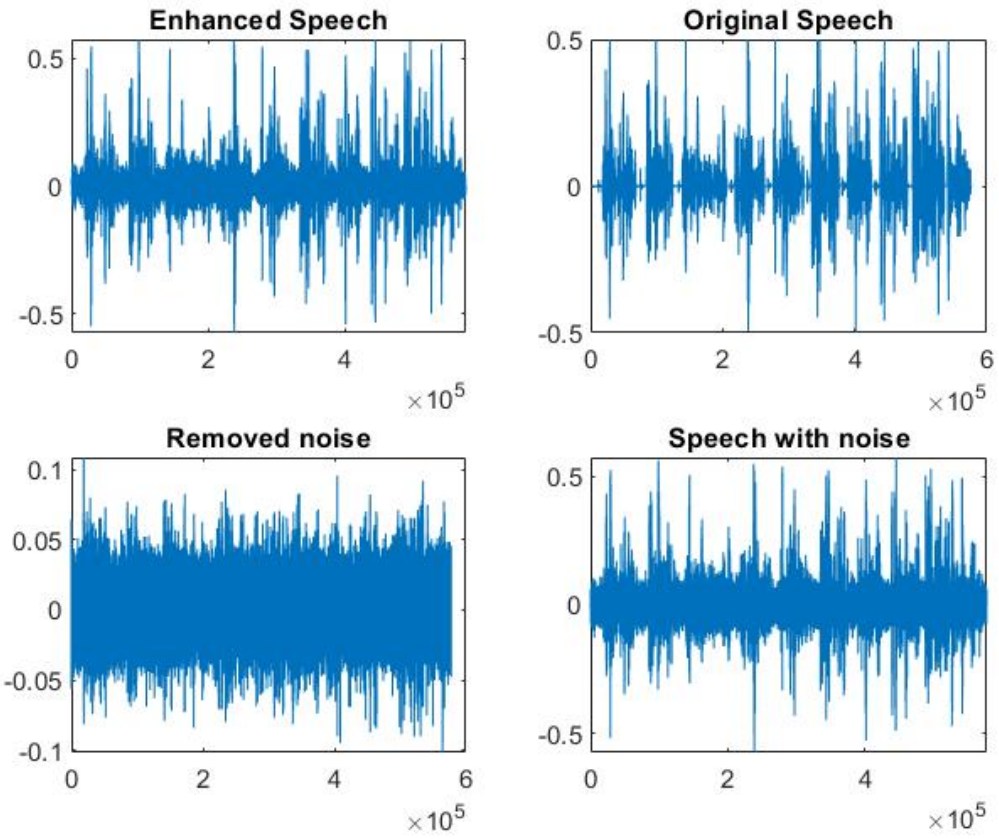


Figure 8: Result of Spectral Subtraction

Compared with wiener filter result, removed noise intensity is higher in some peak points, That means, in our case, spectral subtraction filtered more speech signal. It seems that under this condition, filtered speech signal is "cleaner", however, enhanced speech audio is less clear. Finally the following table shows the SNR difference of both methods with difference values of initial SNR.

	Spectral Subtraction	Wiener Filter
SNR Difference (Initial SNR = -5dB)	6.47 dB	7.15 dB
SNR Difference (Initial SNR = 0dB)	3.76 dB	5.32 dB
SNR Difference (Initial SNR = 5dB)	2.27 dB	3.905 dB
SNR Difference (Initial SNR = 10 dB)	1.21 dB	2.05 dB
SNR Difference (Initial SNR = 15 dB)	-0.27 dB	-0.74 dB

Table 1: SNR Difference with different values of initial SNR

From these results we can conclude that there is an improvement in the SNR in both methods. Moreover, the improvement is greater when the initial SNR is lower, which means that the noise intensity is higher in the noisy speech audio. In addition, we can conclude that when the initial SNR is higher than 10 dB, our algorithm will not enhanced the noisy speech, since the SNR difference is negative. This is because the power level of the speech will be too large compared to the power level of the noise.

6 Conclusion

In conclusion, we have been able to successfully develop two speech enhancement algorithms. In both of them, we have required an estimation of the noise PSD from the noisy speech signal, because in practice we are not given noise and speech separately. We have estimated this PSD during the first seconds of the audio, when the speaker is not actually talking. From the results, we can conclude that the wiener filter worked better since the SNR difference was higher using this method. However, in both methods, some of the frequency components of the original speech were also removed, which slightly affected the audio quality of the enhanced speech.

Keep in mind that we assumed both of the two audio signals (speech and noise) is WSS process, thus the wiener filter model has a great performance on enhancement. When the underlying processes are not WSS, the Wiener filter may not perform optimally, because it relies on the assumption of constant statistical properties. In cases where the signal or noise statistics change over time, the Wiener filter's performance can degrade, as it does not adapt to these changes. Furthermore, we have assumed that the speaker was not talking during the first second of the given noisy speech. Our algorithm would work better if we detected when the speaker is not talking instead of assuming when there is a silence period.

On the other hand, the Kalman filter is designed for estimation and prediction in dynamic systems, including those with non-WSS processes. It can adapt to changing system dynamics and noise statistics, making it a more versatile tool for various applications. The Kalman filter uses a state-space model to estimate the state of a dynamic system based on noisy measurements, and it can update its estimates as new measurements become available. This adaptability allows the Kalman filter to handle time-varying and non-stationary processes effectively.

Appendix

The link for our video on Youtube is here: Video Presentation .
The link of our MATLAB code on Github is here: Matlab Code

References

[1] C. A. Jason, S. Kumar, *et al.*, "An appraisal on speech and emotion recognition technologies based on machine learning," *language*, vol. 67, p. 68, 2020.

[2] K. Paliwal, K. Wójcicki, and B. Schwerin, "Single-channel speech enhancement using spectral subtraction in the short-time modulation domain," *Speech communication*, vol. 52, no. 5, pp. 450–475, 2010.

[3] M. A. Abd El-Fattah, M. I. Dessouky, A. M. Abbas, S. M. Diab, E.-S. M. El-Rabaie, W. Al-Nuaimy, S. A. Alshebeili, and F. E. Abd El-samie, "Speech enhancement with an adaptive wiener filter," *International Journal of Speech Technology*, vol. 17, pp. 53–64, 2014.

[4] J. Benesty, S. Makino, J. Chen, F. Jabloun, and B. Champagne, "Signal subspace techniques for speech enhancement," *Speech Enhancement*, pp. 135–159, 2005.

[5] Y. Tsao and Y.-H. Lai, "Generalized maximum a posteriori spectral amplitude estimation for speech enhancement," *Speech Communication*, vol. 76, pp. 112–126, 2016.

[6] J. R. Deller Jr, "Discrete-time processing of speech signals," in *Discrete-time processing of speech signals*, pp. 908–908, 1993.