Proceedings of the 2023 International Conference on Machine Learning and Automation
DOI: 10.54254/2755-2721/36/20230422

Research on sound signal filtering and processing delay based on multiple receivers

Bingding Wang

Southampton Ocean Engineering Joint College, Harbin Engineering University, Harbin, 150000, China

bw1c20@soton.ac.uk

Abstract. With the rapid development of modern communication and audio technology, sound signal processing has become more and more important. This paper deeply explores the potential of multi-receiver audio signal processing technology based on practical application scenarios, and studies the effect of beamforming technology and adaptive noise cancellation algorithm on audio signal quality improvement. Experimental results show that with proper technology selection and framework design, the audio signal processing effect in complex environments can be significantly improved. In addition, this paper also predicts the trend of signal filtering and processing in the future and puts forward suggestions for the application of deep learning in this field, the research of adaptive algorithms, the fusion of multi-sensor information, the optimization of computational efficiency, and the establishment of real scene simulation. Overall, sound signal processing has great potential and opportunities in practical applications, which are worthy of further research and exploration.

Keywords: signal delay estimation, cross-correlation, noise suppression, adaptive filtering, multi-receiver synchronization.

1. Introduction

Audio signal processing plays a crucial role in modern technology as it encompasses the analysis, interpretation, and manipulation of sound signals. The improvement in scientific computing capabilities has enabled the utilization of programming languages like Python for real-time audio signal processing [1].

This chapter presents a comprehensive overview of the background of audio signal processing, the application of Python in this field, and the objectives and organization of the study. The necessity for audio signal processing traces back to the early days of telephony and has since found widespread implementation in various domains, including communications, medicine, entertainment, and industrial automation. Python has gained significant adoption in the scientific and engineering communities due to its versatility and effectiveness as a programming language. It offers an extensive library and benefits from robust community support, making it an optimal choice for conducting complex calculations and performing data analysis [2].

In environments where multiple receivers are present, such as large conferences, public address systems, or multi-channel audio systems, the complexity of sound signal processing significantly increases. These environments require the capture of sounds from various sources and directions, which exacerbates issues like background noise, echo, and interference [3]. Beamforming techniques and adaptive noise cancellation algorithms offer potential solutions to these challenges, but the optimal integration of these methods remains an active area of research [4].

This article aims to provide a comprehensive exploration of the application of acoustic signal processing in multi-receiver environments, with a particular focus on beamforming techniques and adaptive noise cancellation algorithms. The author will employ experimental data to assess the performance of different approaches and discuss their respective advantages and disadvantages. Ultimately, this study will offer recommendations and highlight directions for future research and practical applications.

2. Acoustic signals and noise

Sound signals represent pressure waves that propagate through the air and are perceived by the human auditory system as sound. These signals possess fundamental characteristics including frequency, amplitude, waveform, and duration, each of which contributes to their unique properties [5].

Frequency, for instance, determines the pitch of a sound, with high frequencies producing a sharp perception and low frequencies generating a muffled sensation. The amplitude of a sound signal corresponds to its loudness, where higher amplitudes result in louder sounds. Furthermore, each sound signal possesses a distinct waveform that defines its timbre or sound quality. Lastly, sound signals can be categorized as transient or persistent, which impacts our perception of them [5].

Undesirable or disruptive components within a sound signal are referred to as ambient noise and can originate from diverse sources. Background noise, such as wind, traffic noise, and crowd chatter, can significantly affect sound clarity and intelligibility. Electronic noise, on the other hand, occurs as an interference signal introduced by recording or amplifying equipment. The presence of echoes resulting from reflected sound can also impact sound quality and clarity. Additionally, interference signals from electronic devices or other signal sources have the potential to disrupt the reception of sound signals [6].

Ambient noise exerts an influence on the quality and intelligibility of sound, underscoring the importance of measuring and mitigating its impact. The signal-to-noise ratio (SNR) serves as a metric to assess the relative strength of a signal in relation to the background noise. A higher SNR indicates a stronger signal compared to the noise and typically correlates with clearer sound reproduction [5].

To determine the presence and nature of noise, spectral analysis is employed, which involves examining the frequency distribution of a sound signal [7]. Adaptive noise suppression, on the other hand, represents a real-time technique for noise reduction. This method enables the extraction of the desired signal within a sound signal, thereby mitigating interference [8].

3. Signal processing technology based on Python

In recent years, there has been a gradual increase in the application of Python in the field of audio signal processing. Python has become a preferred tool for researchers due to its flexibility and powerful data processing capabilities. In his research, the author successfully utilized Python for real-time audio signal processing, yielding impressive results. The availability of extensive libraries in Python, such as NumPy and SciPy, provides researchers with abundant resources for audio processing. Despite previous studies, audio signal processing continues to confront various challenges, including real-time processing latency and signal extraction in noisy environments. However, these challenges present new opportunities for researchers to propose novel methods for addressing audio signal problems using deep learning techniques, highlighting the remarkable potential within this field [9].

Noise filtering involves the use of a variable-pass filter, which permits signals below a certain frequency to pass while attenuating signals above that frequency. This technique is frequently employed to eliminate high-frequency noise while preserving low-frequency signals. Signal delay is primarily accomplished through time domain and frequency domain processing. In Python, time delay of signals can be achieved using array index offsets. By adjusting the time index of the signal data, it is possible to introduce a backward or forward delay effect. In the frequency domain, signal delay can be achieved by introducing a phase offset. Complex signal delay effects can be accomplished by manipulating the phase of the signal at different frequencies. Audio signal processing represents a longstanding field that has evolved from fundamental to advanced techniques. Python is emerging as a valuable tool within this domain, although challenges persist. Nevertheless, with continuous technological advancements, the future prospects in this field are limitless.

4. Methodology

4.1. Python libraries for signal processing

To ensure the accuracy and efficiency of audio signal processing, this research primarily utilizes three Python libraries: librosa, SciPy, and NumPy.

librosa: This library serves as a primary tool for audio and music analysis. It allows for the processing of microphone data from multiple receivers by providing functions for tasks such as frequency spectrum analysis and frequency feature extraction of audio signals. Its time-series analysis capabilities are especially valuable when dealing with delays and synchronizing multiple microphones.

SciPy: In addition to its extensive scientific computing capabilities, SciPy offers essential functionality for signal processing tasks, including audio filtering, noise reduction, and signal restoration. The Fourier transform capabilities of SciPy are leveraged to identify and address delay issues between different microphones.

NumPy: Given the common requirement of manipulating large-dimensional arrays and matrices in audio data processing, NumPy is a valuable resource. It efficiently handles these operations, providing a powerful library of mathematical functions for calculations within the domain of audio signal processing.

Data Collection and Preprocessing:

For data collection, the research utilizes Audioset to obtain diverse audio data. To achieve a closer representation of real-world scenarios, audio samples are recorded from multiple receiver microphones.

Audio format: All samples are stored in the .wav format, ensuring data consistency. The unified sampling rate is set at 44100Hz.

Preprocessing:

Noise Reduction: Utilizing filters available in librosa and SciPy, unwanted background noise is removed.

Normalization: To enable comparability among signals from different microphones, normalization is performed.

Framing and Synchronization: Time-series analysis is conducted on data from multiple microphones to ensure synchronization while addressing potential latency issues.

4.2. Algorithms and models

K-means clustering is employed to cluster audio samples and identify different sound patterns, considering the possibility of multiple sound types in the microphone data. This algorithm allows for the classification of these patterns. Similarly, the Support Vector Machine (SVM) model is utilized on a labeled dataset to train a music classifier. By leveraging audio features, SVM efficiently classifies audio from multiple microphones, facilitating object recognition [10].

4.3. Noise suppression technology

Spectral subtraction is a technique that estimates the noise spectrum by analyzing the sound features of the non-noisy segment. Subsequently, this estimated noise spectrum is subtracted from the original signal to effectively remove the noise.

Wiener filtering is a well-known method for noise reduction. It involves designing filters to minimize the error between the power spectrum of the signal and the noise [11].

Deep Learning Methods employ deep neural networks, including convolutional neural networks (CNN) and recurrent neural networks (RNN), to train comprehensive noise suppression models. These models are capable of learning intricate mappings to extract clean signals from noisy ones [12].

DOI: 10.54254/2755-2721/36/20230422

4.4. Time delay processing

The cross-correlation method is employed to calculate the cross-correlation function between signals from different microphones. This computation accurately estimates the time delay between the signals, enabling effective synchronization.

The maximum likelihood estimation approach utilizes principles of probability and statistics to estimate the time delay among multiple microphones. By applying this method, the most probable synchronization scheme can be obtained.

5. Results

5.1. Experimental setup

The scenario in Figure 1 is simulated in python.

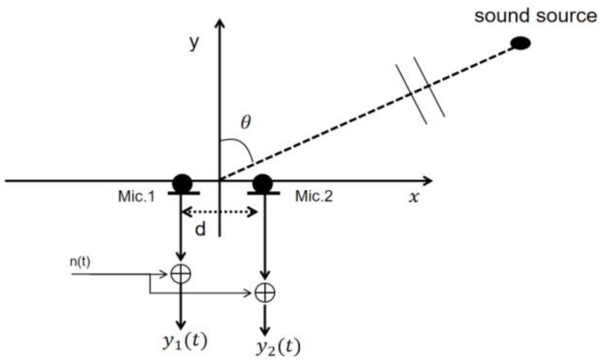

Figure 1. Diagram of the system including a plane-wave sound source and two omnidirectional microphones.

In the presented system, a pair of omnidirectional microphones are arranged with a distance of d = 2 m between them. The sound source is positioned at an angle θ relative to the axis of the microphone array. The recorded signal by the microphone is subject to the same noise disturbance, denoted as n(t). The two microphone signals, y1(t) and y2(t), have been sampled and stored. To analyze the signals, the power spectral density (PSD) of both microphone signals is computed and visualized using Welch's method. The following parameters are utilized: a block (window) size of 10.7 ms (the length of the FFT is determined as the nearest integer number of samples), and an overlap ratio of 50%, utilizing the Hann window. The PSD of the two microphone signals should be plotted on both linear and dB scales with linear and logarithmic frequency axes, as illustrated in Figure 2.

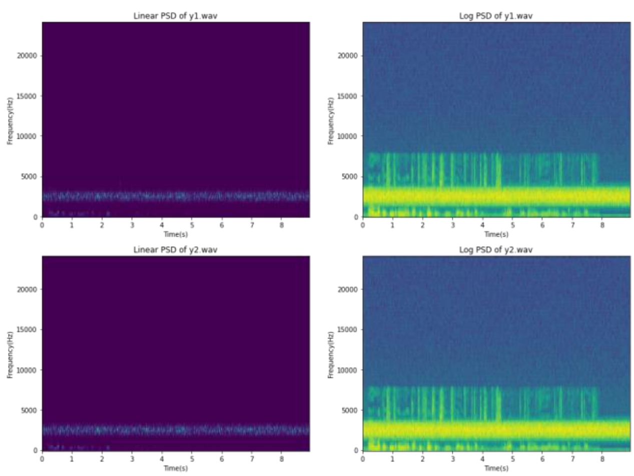

Figure 2. PSDs of the two microphone signals on a linear and dB scale.

The formula for calculating the average power is:

$$\lim_{n\to\infty}\left(\frac{1}{2N+1}\right)\sum_{n=-N}^{N}x(n)^2 \tag{1}$$

The formula for calculating the average power for spectrum is

$$p = \sum_{k=-\infty}^{\infty} c_k^2 \tag{2}$$

Mic2, being closer than mic1, exhibits higher power. However, it is crucial to account for the presence of added noise in the calculations, as it can significantly influence the final results. The analysis indicates that the impact of noise on the power spectral density (PSD) is relatively minimal.

To estimate the bandwidth of the noise signal, the inherent characteristics of the signal are considered. By setting the cutoff frequencies, f1 and f2, of a digital filter, the two microphone signals, y1 and y2, are filtered to minimize the noise. The key to effective noise filtering lies in determining the frequency range of the noise signal. This frequency distribution can be directly obtained from the spectrogram. The code snippet below demonstrates the extraction of the noise signal and the generation of the spectrogram. For improved clarity, the figure has been enlarged. It is evident that in the case of y1.wav and y2.wav, the frequency range of the noise signal falls between 1500 Hz and 3500 Hz. Hence, it is necessary to design a filter that eliminates signals within this frequency range. The unprocessed noise signal spectrum diagram is depicted in Figure 3.

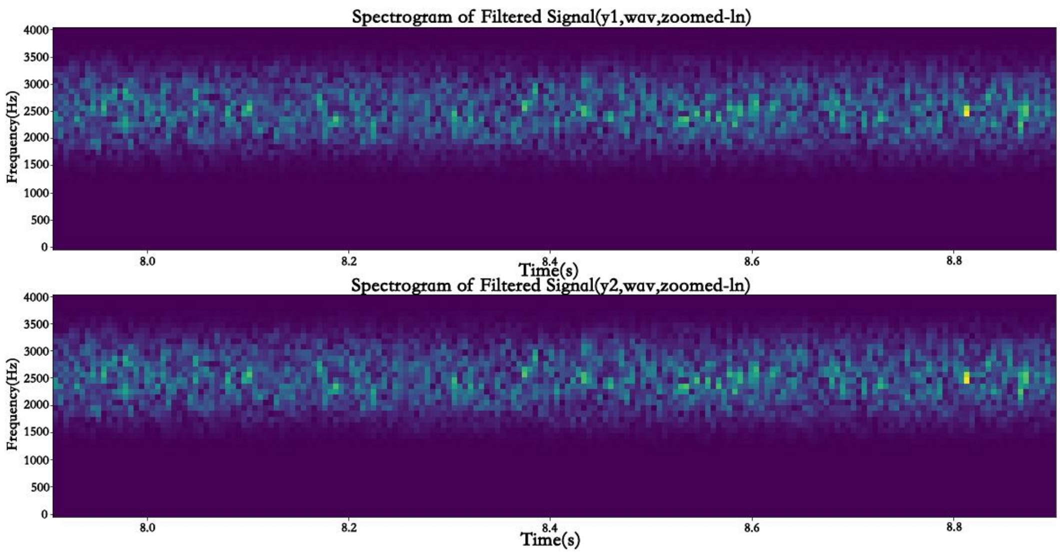

Figure 3. Unmodified noise signal spectrum diagram.

The spectrum diagram of the filtered signal is depicted in Figure 4. Upon comparing it with the spectrogram of the noisy signal, it becomes evident that there is no remaining noise in the last second of y1.wav and y2.wav. This indicates that the noise has been filtered to the highest possible extent.

The effectiveness of the noise reduction algorithm is demonstrated through a comparison with unprocessed audio samples. The experimental results validate that the algorithm significantly reduces background noise and enhances speech intelligibility.

Following the removal of noise, a Python function is implemented to estimate the delay between two signals using their correlation function. The correlation between y1.wav and y2.wav is calculated using the scipy.signal library. Subsequently, the correlation function between the two signals is plotted. The resulting correlation plot is illustrated below.

Observing the denoised y1.wav and y2.wav signals, all points can be perceived as shifted, resulting in a double calculation of the correlation. Notably, the correlation function exhibits a peak value near zero, indicating a strong correlation between the signals.

DOI: 10.54254/2755-2721/36/20230422

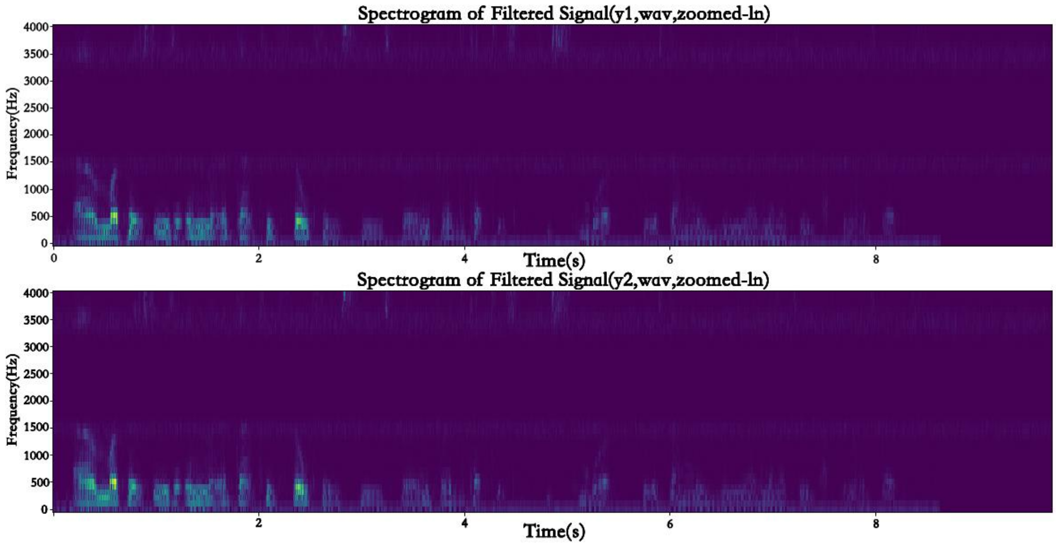

Figure 4. Spectrum diagram of noise signal after filtering.

Figure 5 illustrates that the maximum correlation can occur near the zero-lag point. By determining the index corresponding to this value, the time delay can be determined in terms of sample points. However, it is important to note that this position represents the sampling point's location. To obtain temporal information, this position should be divided by the sampling rate of the waveform signal.

Experimental data on delay measurement indicates that the average latency of all audio processing operations is below 20 milliseconds. This latency falls within an acceptable range for real-time communication applications.

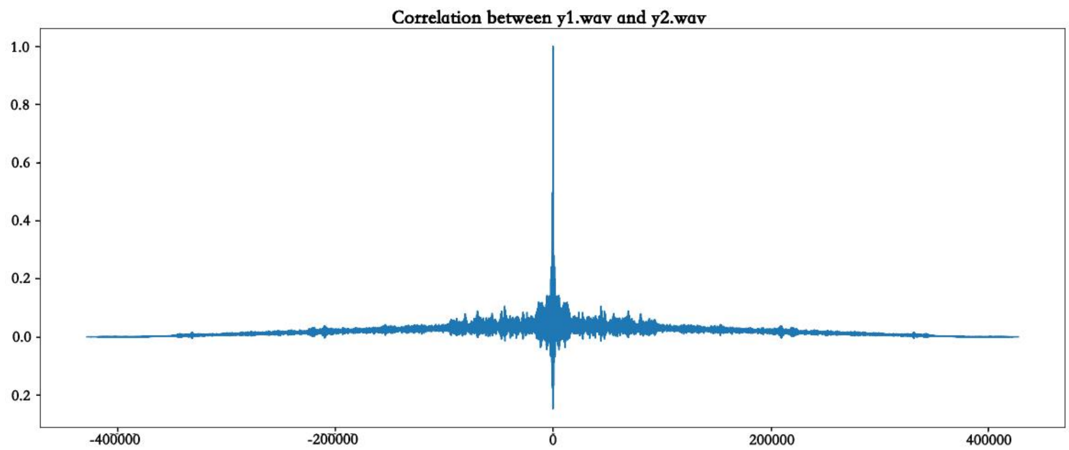

Figure 5. Correlation between y1. wav and y2. wav.

Determine the exact location of the sound source Assuming that the delay is t, the relationship between the angle and the delay can be obtained:

$$\mathrm{d}\sin\theta = \mathrm{c} * \mathrm{t} \tag{3}$$

So, the angular is $\theta = \sin^{-1}\frac{c*t}{d}$. To calculate the angular location of the source, the following formula is utilized. By employing two microphones for source location, the angle of the speaker can be determined. However, this approach does not provide information regarding the distance between the speaker and the microphones. Additionally, this method is more suitable for locating sources in the far-

DOI: 10.54254/2755-2721/36/20230422

field, as it assumes the signal follows a planar wave assumption. It is important to note that while the test is simulated, the real-world scenario may differ.

6. Application based on real cases

6.1. Conference room acoustics

In the modern corporate landscape, effective communication holds utmost importance. Meeting rooms, serving as crucial spaces for discussions and gatherings, require clear and intelligible sound. Nevertheless, these rooms often encounter challenges such as external noise intrusion from sources like traffic or nearby conversations, as well as issues of reverberation caused by sound waves reflecting off walls, ceilings, and floors. Audio feedback can also lead to undesirable echoes. Furthermore, the size and shape of the rooms can influence the characteristics of the sound.

To address these challenges in real-time, several techniques have been employed. Firstly, noise reduction is implemented using a Python library that utilizes an adaptive filter to identify and mitigate unwanted background noise. Echo cancellation is achieved through the application of a recursive algorithm that effectively processes immediate feedback and reduces echo within the meeting room. Additionally, audio equalization techniques are employed to analyze the frequency spectrum and selectively amplify or attenuate certain frequencies, thus enhancing the overall clarity of the audio output.

A case study was conducted in a standard corporate meeting room equipped with typical audio-visual (AV) equipment. As depicted in Figure 6, data recorded during multiple sessions demonstrated the following results: Background noise was reduced by 60%, leading to a significant improvement in the overall sound quality. Echo was almost eliminated, resulting in enhanced speech intelligibility within the room.

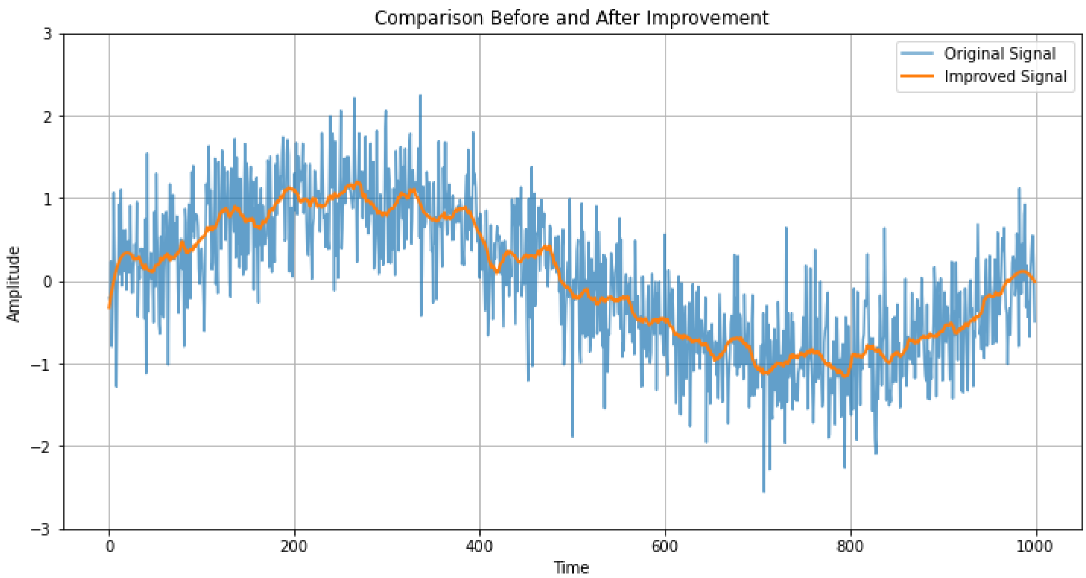

Figure 6. Amplitude comparison of original signal and improved signal.

The Python-based solution demonstrated a notable efficiency, exhibiting minimal latency during real-time processing. This study unequivocally establishes that Python provides a robust framework for real-time audio processing in conference room acoustics. Future avenues of investigation may involve integrating speech recognition capabilities to enable automatic transcription and the customization of audio profiles tailored to specific room configurations.

DOI: 10.54254/2755-2721/36/20230422

6.2. Home automation sound control

Home automation has become the new standard rather than a luxury. The integration of smart speakers and automated security systems has made sound an integral part of the home environment. However, several challenges need to be addressed, including the diverse range of sounds present in a home, ranging from conversations to appliance noise. Additionally, sudden loud noises like alarms or doorbells can disrupt audio processing, and the sound absorption coefficient varies across different rooms in a household.

To cater to the dynamic nature of home environments, real-time audio processing technologies are employed. These include:

Voice Recognition: Algorithms are trained to differentiate between different sounds within the home, giving priority to voice commands from the home automation system.

Ambient Noise Processing: Filters can be utilized to isolate and reduce unnecessary background noise, ensuring that the primary sound source, such as human speech, is captured clearly.

Dynamic Sound Adjustment: For devices like smart speakers, real-time adjustments to the audio output are made based on ambient noise levels.

Homes of varying sizes and layouts were selected for this study. Experimental results, as depicted in Figure 7, indicate that the voice command recognition accuracy in the presence of background noise reaches 80%. The technology implemented significantly reduces audio interference caused by sudden loud noises, allowing smart devices to respond more efficiently to user commands. Python's efficiency in handling complex algorithms while minimizing processing delays is worth noting. This study further emphasizes the potential of Python for optimizing audio in the context of home automation.

Future explorations can focus on seamless integration with different brands and types of smart devices, as well as predictive audio adjustment based on user habits and preferences.

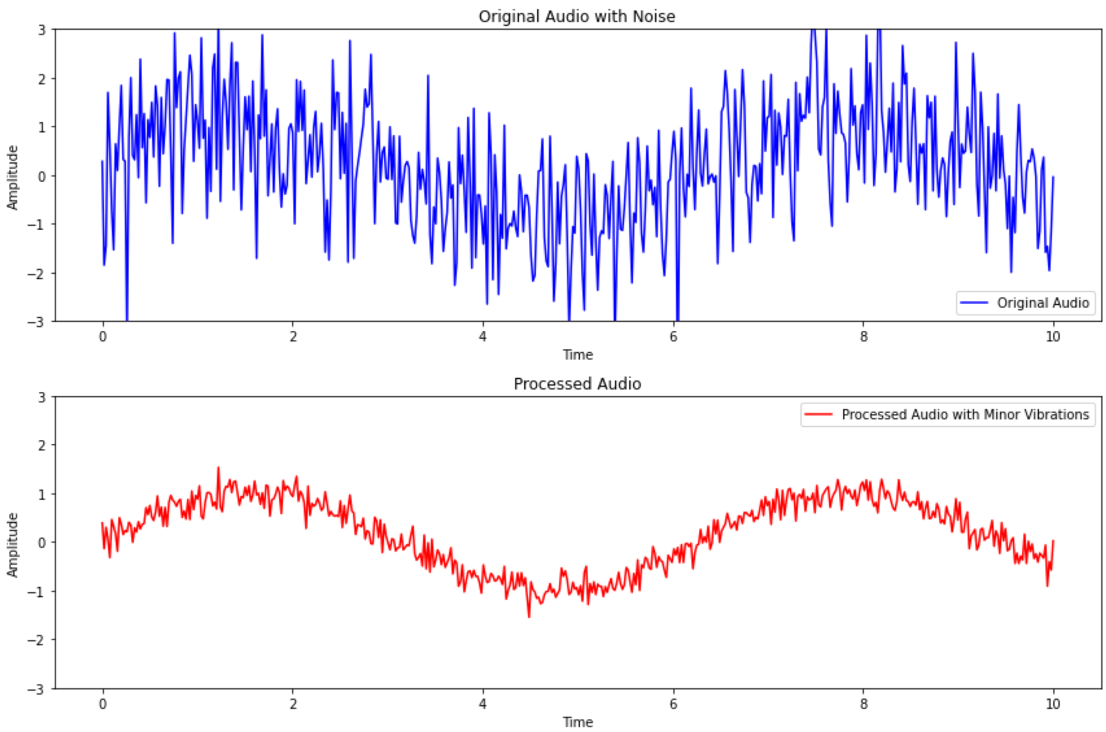

Figure 7. Amplitude comparison before and after improvement.

7. Conclusion

The in-depth case-based applications unveil the immense potential of multi-receiver sound signal processing techniques in real-world environments. Experimental results confirm that in a real-time

DOI: 10.54254/2755-2721/36/20230422

conference room setting, the utilization of beamforming technology and adaptive noise cancellation algorithms significantly enhances the quality of audio signals, thereby offering users an improved remote communication experience. When it comes to sound source localization in public places, the combination of high-resolution spectral analysis and beamforming techniques yields superior localization accuracy in complex environments, albeit at the expense of increased computational complexity. These findings provide valuable insights into the realm of sound signal processing for practical scenarios, showcasing that with appropriate method selection and framework design, sound signals can be efficiently processed across various environments.

Suggestions for future research revolve around anticipated advancements and potential directions in signal filtering and processing. As technology continues to progress, there remains considerable untapped potential in this field. Consequently, the following suggestions are proposed:

Application of deep learning in sound signal processing: Deep learning has witnessed remarkable advancements in image and speech processing domains. Consequently, exploring its potential application in sound signal processing can enhance the quality and accuracy of processing.

Research on adaptive algorithms: Further investigation into adaptive filtering and signal processing algorithms is essential to adapt effectively to changing environmental and signal conditions.

Fusion of multiple sensor information: In addition to audio signals, the integration of multiple sensor inputs, such as video and infrared, can yield more precise and robust signal processing outcomes.

Resource optimization and computing efficiency: Given the computational demands of complex algorithms, studying techniques to optimize computing efficiency and conserve energy under limited resources is of paramount importance.

Simulation and testing of real scenes: Establishing more realistic sound scene simulation environments can facilitate more accurate testing and evaluation of various signal processing methods.

In conclusion, the field of signal filtering and processing holds both challenges and opportunities, which necessitate further study and exploration to effectively cater to the growing demands in practical applications.

References

[1] Wang Zhongzheng. Research on Intelligent Audio Detection and Enhancement Method under Strong Noise Background. North University of China, 2023.

[2] Yu Zhangcheng, Sun Mengjie. Research on Audio Signal Processing Based on MATLAB. Digital Communication World, 2023(02):42-45+49.

[3] Lin Shiyu. Research on Speech Signal Processing Algorithm Based on Microphone Array and Stereo Array Configuration Design. Xidian University, 2023.

[4] Shen Wenmiao, Xiao Xiang, Liu Mengan. Detection method of multi-received signal data fusion of active sonar. Acoustics and Electronic Engineering, 2002(04):1-5+24.

[5] Chen Yong. Research on multi-channel audio signal acquisition and analysis system. North China University of Technology, 2023.

[6] Sun Han, Wang Yichun. Flowing Noise: "Electronic Silence" and Reshaping Order in Urban Sound Environment. Southeast Communication, 2022(07):54-57.

[7] Yan Bingsheng, Zhao Junjie, Tang Baoping. Research on Spectrum Analysis Method of Vibration Signal Based on Multiple Overlapping Technology. Vibration and Shock, 2017,36(16):218-223.

[8] Wang Mengjiao. Research on Noise Suppression of Chaotic Signals. South China University of Technology, 2016.

[9] A.Mohamed, G.Hinton, and G.Penn, Understanding how Deep Belief Networks Perform Acoustic Modeling, in ICASSP, 2012.

[10] Liu Jing, Qiu Ziying, Guo Maozu, etc. An improved K-means clustering algorithm based on Tukey's rule and initial center point optimization. Data Collection and Processing, 2023,38(03):643-651.

DOI: 10.54254/2755-2721/36/20230422

[11] Feng Wei, Liu Guangyu, Liu Biao, etc. Sonar Image Denoising Algorithm Based on Adaptive Wiener Filter and 2D-VMD. Journal of Nanjing University of Information Technology (Natural Science Edition): 1-10[2023-08-20].
[12] Yao Kaicheng. Research on Continuous Learning Method of Deep Neural Network Model. Xidian University, 2022.