Volume 8 No 1 June (2025)

e-ISSN: 2615-3270
p-ISSN: 2615-3599

Eigen Mathematics Journal

Homepage Journal: https://eigen.unram.ac.id/index.php/eigen

Implementation of Fast Fourier Transform and Least Mean Square Algorithms in The Denoising Process of Audio Signal

Putri Rahmasari Rayes[1], Nuzla Af'idatur Robbaniyyah[1*], Syamsul Bahri[1]

[1] *Department of Mathematics, Universitas Mataram, Indonesia*
[*] *Corresponding author: nuzla@unram.ac.id*

A B S T R A C T

Audio signals play an important role as a medium for storing information, such as lecture materials, interview results, and other archives. However, audio signals are often contaminated by noise, which is unwanted interference that can affect their quality. Therefore, a denoising process is needed to reduce or eliminate noise components in the signal. This research aims to enhance audio signal quality using the Fast Fourier Transform (FFT) and Least Mean Square (LMS) algorithms, which are known for their simplicity and ease of implementation. This research uses primary data, specifically audio signals recorded under two noise conditions: rain noise as Audio Signal 1 and guitar instrument noise as Audio Signal 2, both stored in WAV format. The denoising process was performed using MATLAB software and evaluated based on the signal-to-noise ratio (SNR) and mean squared error (MSE) metrics. Higher SNR values and lower MSE values indicate the success of the denoising process in improving audio signal quality. The results of this study demonstrate the effectiveness of the applied algorithms, where the SNR value reached 38.2596 dB with an MSE of 0.0000028211 for Audio Signal 1, and an SNR value of 38.6881 dB with an MSE of 0.0000014988 for Audio Signal 2. An SNR value between 25 dB and 40 dB is categorized as a very good signal, indicating that the quality of the processed audio signals falls into the very good signal category.

Keywords: Audio signal, denoising process, FFT, LMS, signal to noise ratio, mean square error.

Received : 03-12-2024;
Revised : 01-05-2025;
Accepted : 30-05-2025;
Published : 13-06-2025;

DOI: https://doi.org/10.29303/emj.v8i1.255

1. Introduction

The existence of signals in today's digital era is crucial, playing a central role in facilitating connectivity, communication, and access to information. Signals have the capacity to transmit desired data or information from one location to another quickly [1]. In the context of communication, audio signals are the most commonly used, closely related to how audio information (sound) is recorded, processed, and delivered back to listeners in digital format [2]. However, audio signals often face challenges, such as contamination by noise. Noise is an unwanted signal that can affect the quality of

the original signal, leading to disruptions in processes like speech recognition, especially in conversation recordings [3]. To enhance the quality of audio signals, such as recordings, it is important to perform denoising effectively and efficiently.

Denoising is a process aimed at reducing or eliminating noise components by filtering audio signals to improve their quality, enabling the conveyed information to be identified and understood clearly [4]. In this context, the Fast Fourier Transform (FFT) and Least Mean Square (LMS) algorithms are often utilized in the denoising process due to their simple structure and ease of implementation in software. FFT is an algorithm used to transform audio signals from the time domain to the frequency domain. Representing signals in the frequency domain allows for analyzing their frequency spectrum, identifying both desired signal components and unwanted noise [5]. Meanwhile, LMS is an algorithm used to adaptively adjust filters to reduce noise or produce an optimal estimation of the desired audio signal [6].

Several studies have discussed denoising processes using FFT and LMS algorithms. For instance, research by [7] implemented the FFT algorithm to develop a simulation system for reducing noise levels in fan motor sounds. Similarly, research by [6] employed the LMS algorithm to minimize noise in gunshot sound signals. Both algorithms aim to achieve the same goal, reducing noise in audio signals. The use of FFT and LMS in the denoising process is supported by a strong theoretical foundation and has been validated through various previous studies. Both have relatively simple structures and are easy to implement in software. The FFT transforms the signal to the frequency domain for easy noise identification, while the LMS filters out the noise by adaptively adjusting the filter coefficients, resulting in a cleaner and more optimized signal estimation.

Previous research has proven the effectiveness of FFT and LMS separately in noise reduction. This research combines both algorithms in one complementary framework. The difference lies in the use of one homogeneous audio signal source, which is the author's voice recording containing noise. The FFT is used to analyze the frequency spectrum and identify noise, which then becomes input for the LMS in adjusting the adaptive filter to produce a more optimal signal.

2. Research Methods

This research implements the FFT and LMS algorithms to perform denoising on recorded audio signals. The process begins with data collection through the author's voice recording using a mobile phone. There are two data used, the first data is 1.90 MB in the form of audio signals with rain noise, and the second data is 1.93 MB in the form of audio signals with guitar instrument noise. Both recordings are about ± 10 seconds long and contain the greeting "Assalamu'alaikum Warrahmatullahi Wabarakatuh." The data is stored in WAV format to maintain the original audio quality without compression and facilitate signal processing. Implementation stages include:

a) **Data Preprocessing**

This stage is performed to prepare the audio signals before the denoising process using the FFT and LMS algorithms, including the following steps:

- Data Normalization

  Data normalization is performed to standardize the amplitude scale of the signal so that it falls within the range of -1 and 1. The goal is to stabilize the amplitude and prevent distortion or unwanted deviations caused by extreme amplitude scale differences, which can make the sound appear as a constant buzzing noise. Normalization is calculated using the

Max Abs Scaler method. The equation used is as follows [8]:

$$x_{\text{scaled}} = \frac{x}{|\max(x)|} \tag{1}$$

Where,

x : the original value.
$|\max(x)|$: the absolute value of the attribute.

- Zero-Padding

  Zero-padding is used to optimize the performance of the Fast Fourier Transform (FFT) by adding zeros to the data so that the signal reaches a certain length. This process is carried out to extend the number of samples in the signal to a power of 2^n, because the structure of the FFT algorithm is designed to optimally divide and combine signals whose lengths are powers of two. The calculation of zero-padding involves the base-2 logarithm, as shown below:

$$N' = 2^{\lceil \log_2(N) \rceil} \tag{2}$$

  Where $2^{\lceil \log_2(N) \rceil}$ is the smallest power of two greater than N, to get the recommended padding length [9].

- Windowing

  Windowing is used to minimize the side effects of cutting the signal in the Fast Fourier Transform (FFT). The windowing process involves multiplying the signal by a window function to produce smoother transitions at the edges of the signal (the start and end). A commonly used windowing technique is the Hamming window, as it generates smoother transitions. The equation for the Hamming window is as follows [10]:

$$w(n) = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N-1}\right) \tag{3}$$

  Where,

  $w(n)$: the value of the Hamming window at index n
  N : the total number in the window
  n : the index that varies from 0 to $N-1$

b) **Fast Fourier Transform (FFT)**

The Fast Fourier Transform (FFT) is a numerical algorithm used to compute the Discrete Fourier Transform (DFT) quickly and efficiently by reducing the time complexity from $O(N^2)$ to $O(N \log N)$ [11, 12]. The DFT represents a discrete signal in the time domain as a frequency domain signal.

The FFT equation is derived by decomposing the DFT into smaller components by dividing the total number of samples N into even and odd components, as follows:

$$X(m) = \sum_{n=0}^{\frac{N}{2}-1} \left[x(2n) e^{-\frac{j2\pi(2n)m}{N}} + x(2n+1) e^{-\frac{j2\pi(2n+1)m}{N}} \right] \tag{4}$$

Where,

$x(2n)$: the signal value at even indices.
$x(2n+1)$: the signal value at odd indices, with n being a positive integer.
$e^{-\frac{j2\pi(2n+1)m}{N}}$: the complex factor representing the phase for each sample.

This stage converts the audio signal from the time domain to the frequency domain with the goal of identifying noise [12].

c) **Inverse Fast Fourier Transform (IFFT)**

The Inverse Fast Fourier Transform (IFFT) is used to convert a signal processed by the FFT from the frequency domain back to the time domain. The equation for the continuous IFFT is given by [5]:

$$x(t) = \int_{-\infty}^{\infty} x(f) e^{j2\pi ft}\, df \tag{5}$$

Where,

$x(t)$: the continuous signal in the time domain.
$x(f)$: the frequency spectrum of the signal $x(t)$.
$e^{j2\pi ft}$: the complex exponential function with frequency f and time variable t.

This stage is performed to reconstruct the audio signal from the frequency domain back to the time domain, aiming to preserve the frequency information [11].

d) **Least Mean Square (LMS)**

The Least Mean Square (LMS) algorithm is an iterative method used to optimize filter coefficients, such as the learning rate and filter order (signal length), to minimize the error between the generated output signal and the desired target signal [4]. The general form of the LMS algorithm for updating the adaptive filter coefficients is given by [13]:

$$W_{n+1} = W_n + \mu e(n)\mathbf{x}^*(n) \tag{6}$$

where,

W_n : the filter coefficient vector at the n-th iteration.
W_{n+1} : the updated filter coefficient vector at the $(n+1)$-th iteration.
μ : the learning rate.

This stage is used to optimize the adaptive filter in order to minimize the error [4].

e) **Signal To Noise Ratio (SNR)**

Signal-to-Noise Ratio (SNR) is an indicator used to measure the quality of a signal affected by noise [14]. The SNR value represents the ratio between the power of the signal and the power of the noise, and is mathematically expressed as [15]:

$$\text{SNR} = 10 \log \left(\frac{P_{\text{Signal}}}{P_{\text{Noise}}} \right) \tag{7}$$

where,

P_{Signal} : power of the signal.
P_{Noise} : power of the noise.

The SNR value indicates the level of signal quality compared to the level of noise [16].

This stage is used to measure the quality of the audio signal affected by noise. In this research, the SNR is considered optimal when SNR ≥ 25 dB [17].

f) **Mean Square Error (MSE)**

Mean Squared Error (MSE) is an indicator used to evaluate the minimum error value that can be achieved by a noise removal system [14]. MSE is calculated as the average of the squared differences between the predicted values and the actual values [18]. The mathematical formulation is given as:

$$\text{MSE} = \frac{1}{N}\sum_{i=0}^{N}(x_i - y_i)^2 \tag{8}$$

where,

N : number of signal data points,
x_i : actual value,
y_i predicted value.

A lower MSE value, approaching zero, indicates more accurate predictions that closely match the actual values [19].

This stage is used to evaluate the minimum error achieved by the noise removal system. In this research, the MSE is considered minimum when the value is below the defined error tolerance threshold of 0.00001 [20].

3. Result and Discussion

3.1. Input Sinyal Audio

The first step is to read the audio signal from the wav file and represent it in the time domain. Below is the plot of the audio signal contaminated by noise:

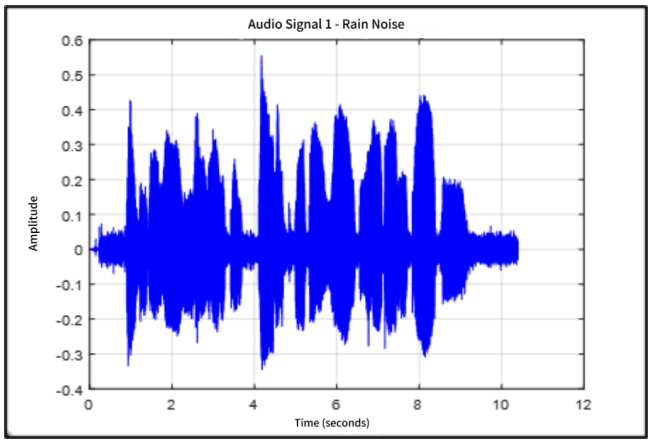

Figure 1. Audio Signal 1 in the Time Domain

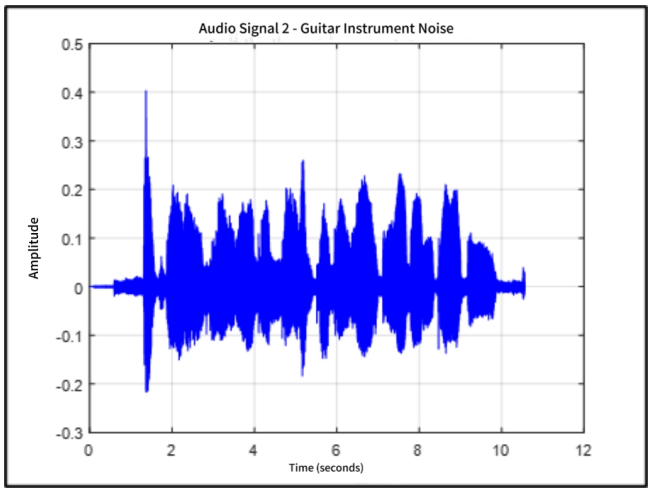

Figure 2. Audio Signal 2 in the Time Domain

Based on Figure 1 and 2, the horizontal axis represents time (seconds), and the vertical axis represents amplitude. Each point represents the signal's amplitude at a specific time. The amplitude pattern shows random variations, indicating the presence of noise and reflecting the sound intensity over time.

3.2. Data Normalization

The following is a representation of the normalized data sample presented in the form of a histogram.

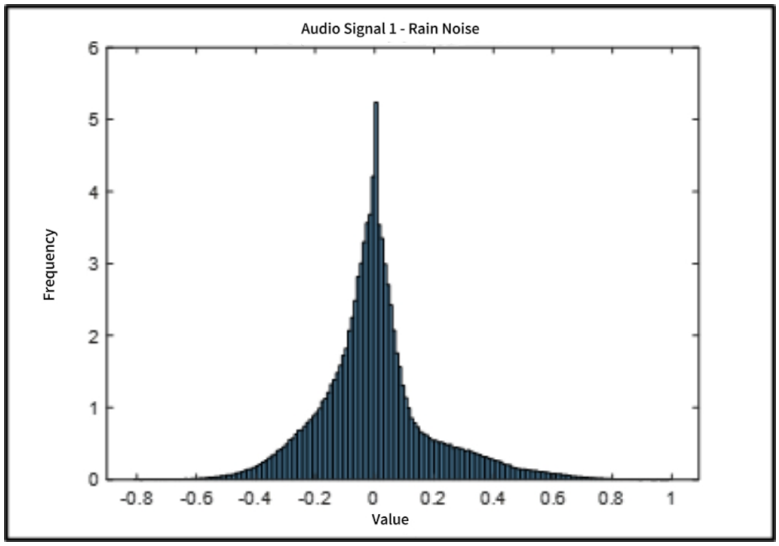

Figure 3. Normalized Data Histogram of Audio Signal 1

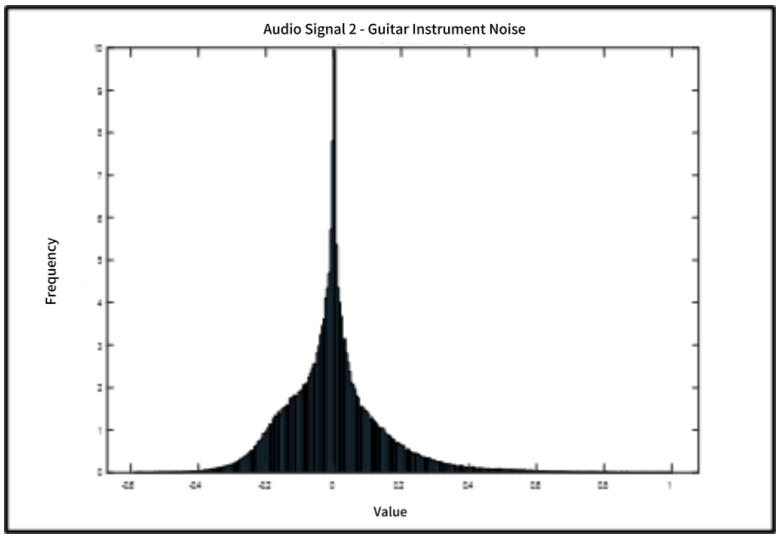

Figure 4. Normalized Data Histogram of Audio Signal 2

Each histogram has a relatively symmetrical distribution shape with a peak around the value of 0. This indicates that most of the amplitude data ranges close to zero. This is typical of a normal distribution, where data tends to be concentrated around the mean (in this case, zero) with a decrease in frequency in both directions as the amplitude value moves away from zero.

3.3. Zero-Padding

Zero-padding is calculated using the base 2 logarithm. The lengths of audio signals 1 and 2 of 499200 and 506880 respectively are enlarged to 524288, which is the nearest power of two of both signal lengths. This step optimizes the FFT algorithm by reducing the time complexity. The following is the calculation of time complexity by FFT:

$$\text{FFT} = O(N \log N)$$
$$\text{FFT} = O(524288 \log 524288)$$
$$\text{FFT} = O(9961472)$$

This calculation shows that the time complexity of the FFT is much smaller than the DFT for the same signal size, $9961472 < 249600640000$.

3.4. Windowing

The following is a representation of the hamming window value on audio signal 1 and audio signal 2.

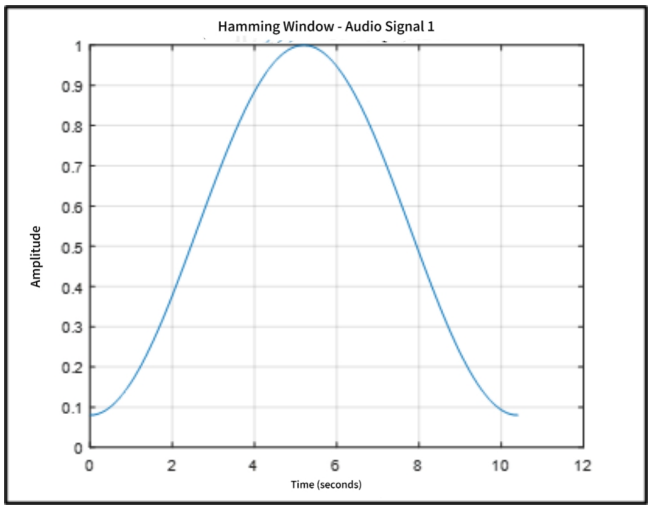

Figure 5. Hamming Window Audio Signal 1

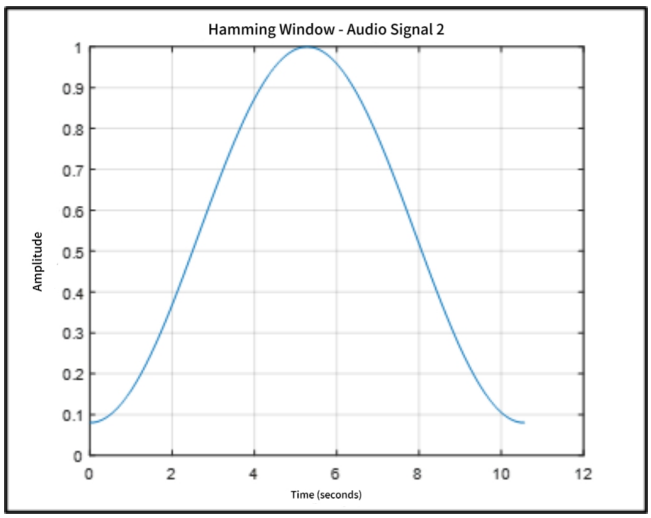

Figure 6. Hamming Window Audio Signal 2

The application of hamming window changes the amplitude scale, especially at the ends of the signal (where the signal is cut off). The signal amplitude becomes smaller at the edges and closer to its original value in the middle. This shows how hamming window is used to reduce the effects of frequency spectrum leakage when performing signal analysis in the frequency domain by Fast Fourier Transform (FFT).

3.5. The Fast Fourier Transform Process

The Fast Fourier Transform (FFT) in the denoising process is used to convert a signal from the time domain to the frequency domain to identify signal and noise components. This process involves thresholding, which separates the signal and noise components by setting a threshold based on the signal amplitude in the frequency domain, as shown in the following plot:

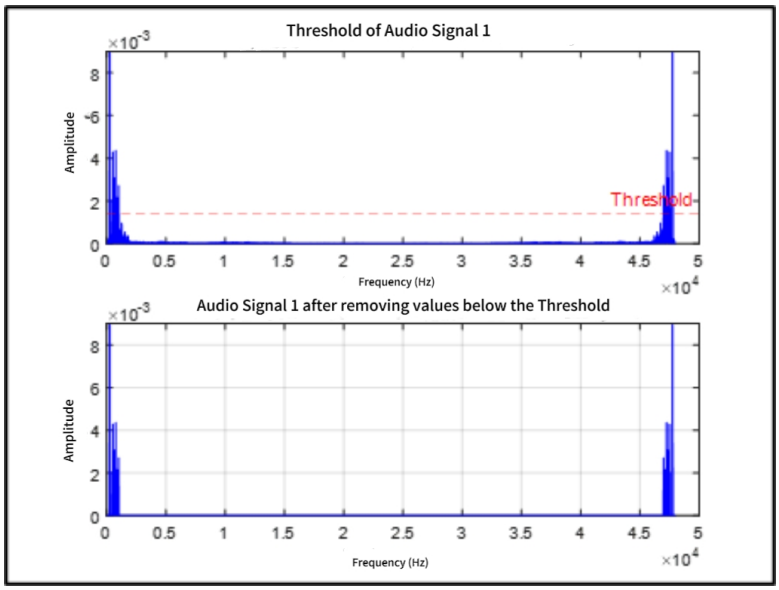

Figure 7. Thresholding on Audio Signal 1

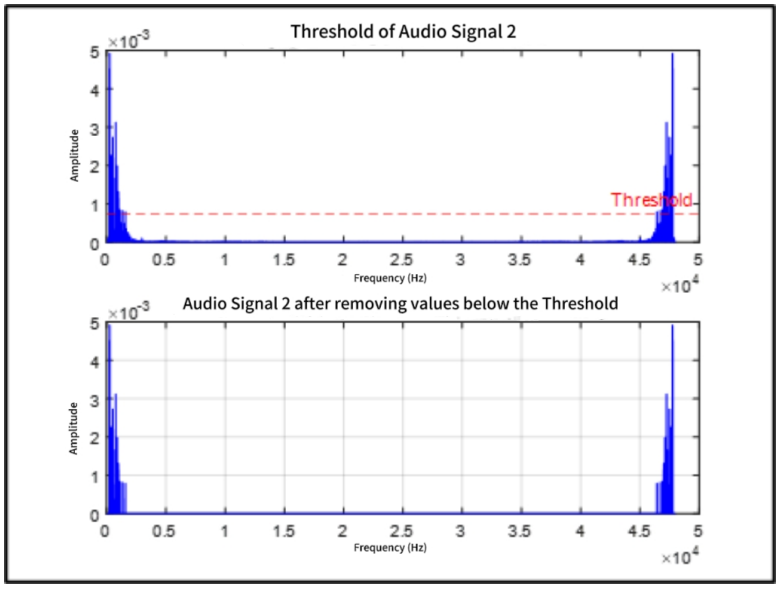

Figure 8. Thresholding on Audio Signal 2

3.6. The Inverse Fast Fourier Transform Process

The Inverse Fast Fourier Transform (IFFT) is used to convert the signal from the frequency domain back to the time domain because signals in the time domain are the form that can be utilized, interpreted, and applied in real-life scenarios. The following is a plot of the signal after applying IFFT.

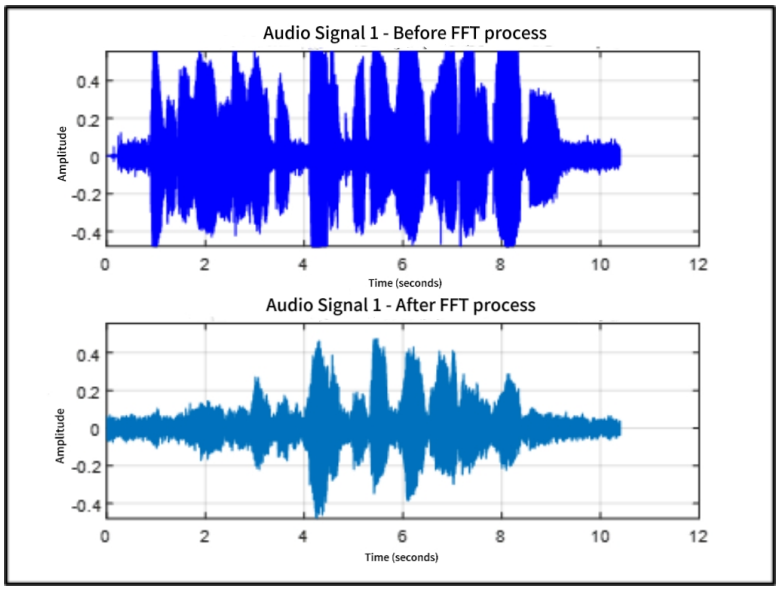

Figure 9. Audio Signal 1 before and after the FFT Process

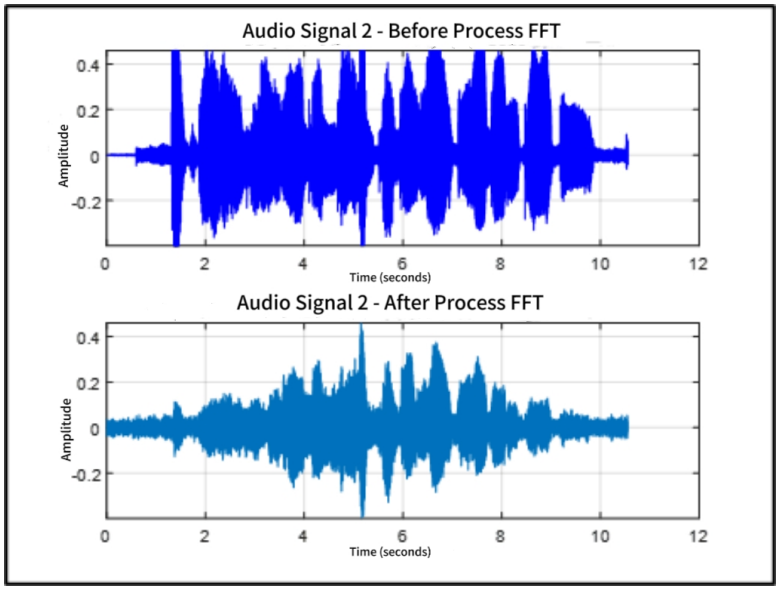

Figure 10. Audio Signal 2 before and after the FFT Process

Figure 9 and 10 shows significant changes in the audio signal after filtering with FFT. The decrease in signal amplitude indicates that the filter effectively reduced noise.

3.7. The Least Mean Square Process

The use of the Least Mean Square (LMS) algorithm, which operates in the time domain, optimizes filter coefficients such as the learning rate and filter order to minimize the error between the generated output signal and the desired target signal. Below is the plot of the denoising results performed by the Least Mean Square (LMS) algorithm.

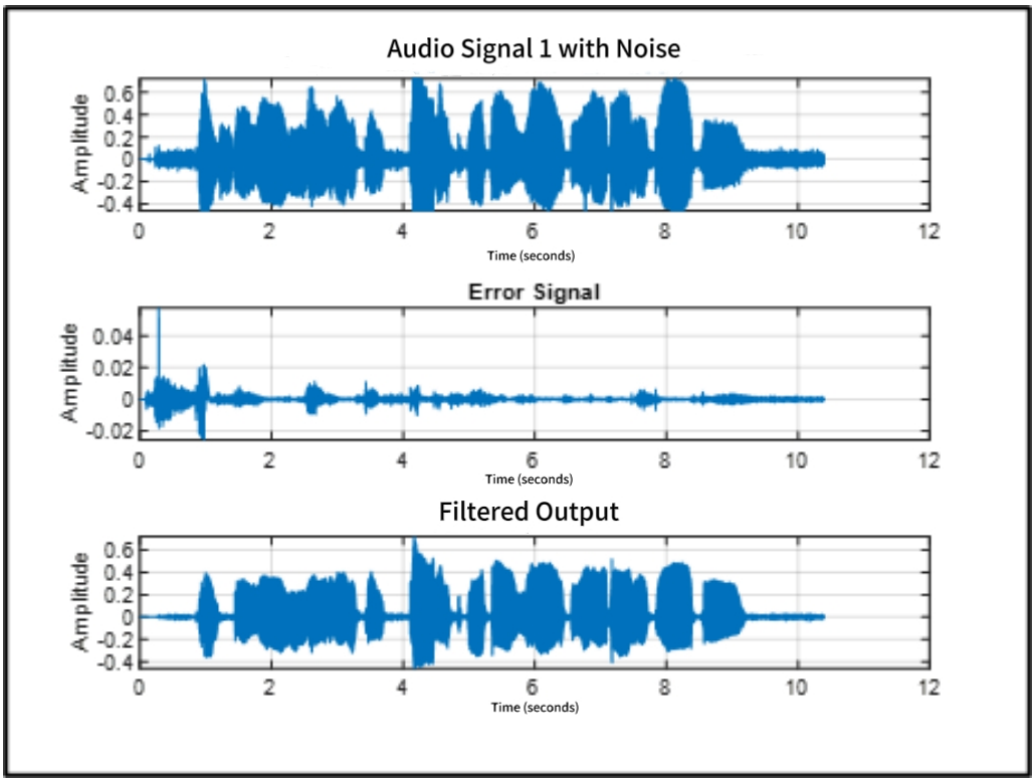

Figure 11. Denoising Results Display on Audio Signal 1

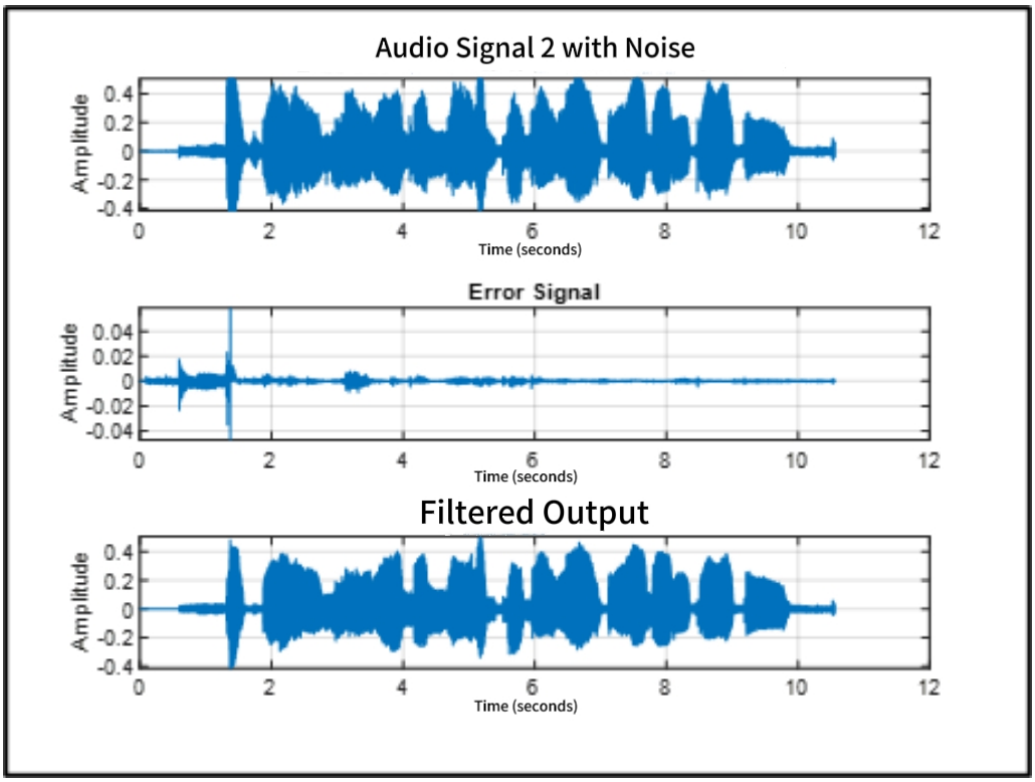

Figure 12. Denoising Results Display on Audio Signal 2

There are three subplots showing the filtering results using the LMS algorithm. The first subplot depicts the audio signal contaminated with noise, showing random amplitude variations. The second subplot displays the error signal, which is the difference between the input signal after the FFT process and the filter output at each iteration. The decrease in the error signal amplitude indicates that the LMS filter is effective in reducing noise. The third plot shows the output signal after denoising, which appears cleaner and more stable, indicating the success of the LMS filter.

3.8. Signal To Noise Ratio (SNR)

SNR is a parameter used to measure the quality of a signal affected by noise. The higher the SNR value, the greater the success of the denoising process. Below is the table of the output SNR values obtained.

Table 1. Output SNR Value of Audio Signal 1

Learning Rate	SNR Before Denoising (dB)	SNR After Denoising (dB)		
		Order 16	Order 32	Order 128
0.001	4.345	17.9677	18.2905	19.3479
0.005	4.345	23.5995	24.1944	25.8284
0.01	4.345	26.3791	26.9824	28.4583
0.05	4.345	33.0720	33.5454	34.6160
0.1	4.345	35.8634	36.3616	37.2911
0.12	4.345	36.5898	37.1071	37.9691
0.13	4.345	36.9079	37.4350	38.2596
0.14	4.345	37.2020	37.7388	3.7942
0.15	4.345	37.4756	38.0218	0.000000021

Table 2. Output SNR Value of Audio Signal 2

Learning Rate	SNR Before Denoising (dB)	SNR After Denoising (dB)		
		Order 16	Order 32	Order 128
0.001	6.6953	16.1019	16.4033	17.1599
0.005	6.6953	21.8014	22.3999	23.7629
0.01	6.6953	24.3167	24.9675	26.4627
0.05	6.6953	30.9689	31.5323	32.5773
0.1	6.6953	33.8280	34.3586	35.2659
0.12	6.6953	34.5736	35.0994	35.9669
0.13	6.6953	34.9003	35.4246	36.2720
0.14	6.6953	35.2024	35.7256	36.5525
0.15	6.6953	35.4834	36.0059	36.8116
⋮	⋮	⋮	⋮	⋮
0.27	6.6953	37.8660	38.3922	38.6881
0.28	6.6953	38.0127	38.5397	7.2912
0.29	6.6953	38.1540	38.6819	0.00043

Tables 1 and 2 show that increasing the Learning Rate (μ) generally improves the SNR value; however, a value of μ that is too large can lead to instability. This is evident from the decrease in SNR observed in each audio signal, with Audio Signal 1 dropping from 38.2596 dB to 3.7942 dB, and Audio Signal 2 dropping from 38.6881 dB to 7.2912 dB. Furthermore, a higher filter order also tends to improve the SNR, but it must be adjusted with μ to avoid instability. In this case, selecting the correct learning rate and filter order is crucial to achieving the best results. For Audio Signal 1, the SNR reaches its optimal value when $\mu = 0.13$ and filter order $= 128$, achieving 38.2596 dB compared to the pre-denoising SNR of 4.345 dB, which shows a significant improvement. Similarly, for Audio Signal 2, the SNR reaches its optimal value when $\mu = 0.27$ and filter order $= 128$, achieving 38.6881 dB compared to the pre-denoising SNR of 6.6953 dB, which also shows a significant improvement.

3.9. Mean Square Error (MSE)

MSE is a parameter used to evaluate the minimum MSE value achieved by the noise removal system. The closer the value is to the minimum, the more accurate the prediction. Below are the minimum output MSE values.

Table 3. Output MSE Value of Audio Signal 1

Learning Rate	MSE		
	Order 16	Order 32	Order 128
0.001	0.0002900	0.0002696	0.0002127
0.005	0.0000815	0.0000712	0.0000490
0.01	0.0000432	0.0000376	0.0000268
0.05	0.0000093	0.0000083	0.0000065
0.1	0.0000048	0.0000043	0.0000035
0.12	0.0000041	0.0000036	0.0000030
0.13	0.0000338	0.0000340	0.0000028
0.14	0.0000035	0.0000317	0.0135380
0.15	0.0000011	0.0000297	394153033

Table 4. Output MSE Value of Audio Signal 2

Learning Rate	MSE		
	Order 16	Order 32	Order 128
0.001	0.0002551	0.0002386	0.0002030
0.005	0.0000720	0.0000628	0.0000460
0.01	0.0000406	0.0000350	0.0000248
0.05	0.0000088	0.0000077	0.0000061
0.1	0.0000045	0.0000040	0.0000032
0.12	0.0000038	0.0000034	0.0000028
0.13	0.0000035	0.0000031	0.0000026
0.14	0.0000033	0.0000029	0.0000024
0.15	0.0000031	0.0000027	0.0000023
⋮	⋮	⋮	⋮
0.27	0.0000028	0.0000260	0.0000014
0.28	0.0000027	0.0000025	0.0025414
0.29	0.0000036	0.0000035	112329980

For each filter order, the MSE value tends to decrease as μ increases, indicating that an increase in μ accelerates the reduction of error. However, if μ increases too much, it can lead to instability or divergence. For Audio Signal 1, the MSE sharply increases from 0.0000028 to 0.0135380 and 394153033 as μ changes from 0.13 to 0.14 and 0.15 with a filter order of 128. A similar trend occurs in Audio Signal 2, where the MSE rises from 0.0000014 to 0.0025414 and 112329980 as μ changes from 0.27 to 0.28 and 0.29 with a filter order of 128. The minimum MSE value is achieved when it is below the error tolerance of 0.00001, which is 0.0000028 for Audio Signal 1 and 0.0000014 for Audio Signal 2.

4. Conclusions

The implementation of the Fast Fourier Transform (FFT) and Least Mean Square (LMS) algorithms aims to reduce noise in audio signals. FFT is used to identify the noise, while LMS is used to optimize the filter to minimize the noise. This combination significantly reduces noise, as seen in the generated graphs, which show a decrease in amplitude intensity and smaller fluctuations compared to the pre-filtered signal. The quality of the resulting audio signal is categorized as "very good signal" based on the SNR and MSE values. For Audio Signal 1, the SNR reaches 38.2596 dB with an MSE of 0.0000028211, and for Audio Signal 2, the SNR reaches 38.6881 dB with an MSE of 0.0000014988.

References

[1] R. Y. Sipasulta, A. S. M. Lumenta, and S. R. U. A. Sompie, "Simulasi Sistem Pengacak Sinyal Dengan Metode FFT (Fast Fourier Transform)," *E-Journal Teknik Elektro Dan Komputer*, vol. 3, no. 2, pp. 1–9, 2014. https://doi.org/10.35793/jtek.v3i2.4448.

[2] S. T. Wahyudi, E. Safrianti, and Y. Rahayu, "Aplikasi Spectrum Analyzer untuk Menganalisis Frekuensi Sinyal Audio Menggunakan MATLAB," *Jom FTEKNIK*, vol. 2, no. 2, 2015. https://jnse.ejournal.unri.ac.id/index.php/JOMFTEKNIK/article/view/7964.

[3] S. Wirawan and E. Prasetyo, "Implementasi Metode Noise Gate, Low Pass Filter, dan Silent Removal untuk Menghilangkan Noise pada File Suara Menggunakan Parameter Dinamis," *Jurnal Ilmiah Teknologi Rekayasa*, vol. 21, no. 3, pp. 152–162, 2017. https://ejournal.gunadarma.ac.id/index.php/tekno/article/view/1594.

[4] S. Gunawan, E. S. Rahman, and Mardhatillah, "Metode Penentuan Perbaikan Noise Pada Data Musik Menggunakan Algoritma Least Mean Square," *Journal of Digital Technology and Computer Science*, vol. 01, no. 1, pp. 48–56, 2023. https://doi.org/10.61220/digitech.v1i1.20235.

[5] N. D. Pah, *Pemrosesan Sinyal Digital*. Surabaya, Indonesia: Graha Ilmu, 2018. .

[6] R. M. Untsa, F. S. Akbar, H. Briantoro, N. Rachmaningrum, and U. Mustakim, "Filter Least Mean Square (LMS) untuk Mengurangi Noise pada Sinyal SuaraTembakan," *Proceedings of the National Conference on Electrical Engineering, Informatics, Industrial Technology, and Creative Media*, vol. 3, no. 1, pp. 104–112, 2023. https://conferences.ittelkom-pwt.ac.id/index.php/centive/article/view/165.

[7] A. A. Bimantara, M. S. Adhi, D. Priambodo, H. M. Azhar, and A. Junaidi, "Simulasi Penghilangan Noise Pada Sinyal Suara Menggunakan Metode Fast Fourier Transform (FFT)," *Journal of Informatics, Information System, Software Engineering and Applications (INISTA)*, vol. 1, no. 2, pp. 20–25, 2019. http://doi.org/10.20895/inista.v1i2.45.

[8] Y. Wang and S. Jia, "MADRAS-NET: A deep learning approach for detecting and classifying android malware using Linknet," *Measurement: Sensors*, vol. 33, p. 101113, 2024. https://doi.org/10.1016/j.measen.2024.101113.

[9] L. R. Rizkina, Edwar, and L. O. Nur, "Perancangan dan implementasi pengolahan sinyal radar untuk pengukuran doppler, range, dan sar imaging menggunakan raspberry pi."

[10] M. W. Setiawan, L. Novamizanti, and I. N. A. Ramatryana, "Pemisahan chorus pada musik mp3 menggunakan koefisien korelasi 2-d berbasis discrete cosine transform (dct) dan k-nearest neighbor (k-nn)."

[11] E. O. Brigham, *The Fast Fourier Transform and Its Applications*. Prentice Hall Signal Processing Series, Englewood Cliffs, N.J: Prentice-Hall, 1988.

[12] S. S. Randhawa, "Analysing & Implementing Cooley Tukey Fast Fourier Transform Algorithm," *Research Gate*, pp. 1–4, 2018. http://doi.org/10.13140/RG.2.2.19037.33767.

[13] M. H. Hayes, *Statistical Digital Signal Processing and Modeling*. New York: John Wiley & Sons, 1996. .

[14] H. G. Alfarizi, J. Raharjo, and J. H. Manurung, "Noise Cancellation of Speech Signal Using Dual Microphone System with Discrete Cosine Transform Least Mean Square Algorithm," *e-Proceeding of Engineering*, vol. 5, no. 2, pp. 2161–2168, 2018. https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/6641/0.

[15] Khairunnisa, Sarifudin, and Z. Ahyadu, "Uji Kualitas Sinyal Audio dengan Metode Fourier dan Metode Wevelet," *Seminar Nasional Terapan Riset Inovatif*, vol. 7, no. 1, pp. 555–562, 2021. .

[16] F. N. Rahmawati, A. Hambali, and M. I. Maulana, "Analisis kinerja kanal berkabut pada free space optics." https://openlibrary.telkomuniversity.ac.id/home/catalog/id/155819/slug/analisis-kinerja-kanal-berkabut-pada-free-space-optics.html.

[17] E. I. Alwi, "Analisis Kualitas Sinyal WiFi pada Universitas Muslim Indonesia," *INFORMAL: Informatics Journal*, vol. 4, no. 1, pp. 30–39, 2019. http://doi.org/10.19184/isj.v4i1.10153.

[18] A. D. Hendrata and A. Prihanto, "Analisis Kualitas Suara Stego Audio Penyisipan Informasi Tersembunyi dengan Metode Least Significant Bit," *Journal of Informatics and Computer Science (JINACS)*, vol. 2, no. 03, pp. 178–184, 2021. https://doi.org/10.26740/jinacs.v2n03.p178-184.

[19] A. Fitriatul Aisyah and A. Noortjahja, "Implementasi Hidden Markov Models (HMM) sebagai Filter untuk Mereduksi Noise pada Esophageal Speech," *Jurnal Inovasi Fisika Indonesia*, vol. 4, no. 3, pp. 7–14, 2015. https://doi.org/10.26740/ifi.v4n3.p%p.

[20] D. H. Tanjung, "Jaringan Saraf Tiruan dengan Backpropagation untuk Memprediksi Penyakit Asma," *Creative Information Technology Journal*, vol. 2, no. 1, p. 28, 2015. https://doi.org/10.24076/citec.2014v2i1.35.