Data Denoising With FFT

A PROJECT REPORT

21MAT212

Submitted by

BL.EN. U4AIE21126 Suryamritha M

BL.EN. U4AIE21128 T.VNL Harshitha

BL.EN. U4AIE21139 Varshini Balaji

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING (Artificial Intelligence)

AMRITA SCHOOL OF COMPUTING, BANGALORE

AMRITA VISHWA VIDYAPEETHAM

BANGALORE 560 035

June – 2023

ABSTRACT

This report presents a simple example of audio de-noising using the Short Time Fourier Transform (STFT) technique. The objective is to remove noise from a given speech signal and recover the clean original signal. The provided code generates a noisy signal contaminated with random noise and applies the STFT-based de-noising approach. The noisy and clean signals are plotted, demonstrating the effectiveness of the technique in attenuating noise and preserving the underlying signal. The power spectral density (PSD) is calculated and visualized, showing the frequency content of the signal. A thresholding operation is applied to identify and filter out noise components from the PSD and the Fourier Transform. The filtered signal is obtained through an inverse Fourier Transform. This example showcases the application of STFT for audio de-noising and highlights the potential for improving signal quality in various speech signal processing tasks.

INDEX

INTRODUCTION

FFT(Fast Fourier Transform)

The Fast Fourier Transform (FFT) is an algorithm used to efficiently compute the discrete Fourier transform (DFT) of a sequence or a set of data points. The DFT is a mathematical transformation that converts a time-domain signal into its frequency-domain representation.

The FFT algorithm takes advantage of the symmetry properties of the complex exponential function to break down the DFT into smaller, more manageable sub-problems. It recursively divides the input data into smaller subsets, performs intermediate calculations, and then combines the results to obtain the final frequency-domain representation.

Fast Fourier Transform is a fundamental tool in digital signal processing, enabling the analysis and manipulation of signals in the frequency domain, which has applications in fields ranging from telecommunications and audio processing to medical imaging and scientific research.

IFFT (Inverse Fast Fourier Transform)

Inverse Fast Fourier Transform is the reverse process of the Fast Fourier Transform (FFT). While the FFT converts a signal from the time domain to the frequency domain, the IFFT performs the opposite operation by converting a signal from the frequency domain back to the time domain.

The IFFT algorithm uses the same principles as the FFT but reverses the order of calculations. It takes a complex-valued frequency-domain representation of a signal, typically obtained using the FFT, and computes the corresponding time-domain representation.

The IFFT is commonly used in applications such as digital signal processing, audio and video compression, communications systems, and image processing. It allows for efficient processing and manipulation of signals in the frequency domain, followed by a transformation back to the time domain for interpretation or further processing.

The IFFT is often paired with the FFT in applications where signals need to be transformed back and forth between the time and frequency domains. Together, they provide a powerful toolset for analyzing, processing, and synthesizing signals in various fields of science, engineering, and technology.

Denoising Data with FFT

Denoising data with the Fast Fourier Transform (FFT) involves using the frequency domain representation of signals to remove or reduce noise. The process includes applying the FFT to convert the signal from the time domain to the frequency domain, analyzing the frequency spectrum to identify noise components, and applying filtering techniques to attenuate or eliminate those components. Finally, the inverse FFT is used to transform the modified signal back to the time domain, resulting in a denoised signal.

Literature Survey

1. ECG Denoising Using MATLAB, International Journal of Scientific & Engineering Research, Volume 4, Issue 5, May-2013

In the study, various filtration techniques were employed to address noise corruption in ECG signals recorded using analog circuitry. The researchers aimed to minimize the impact of noise by implementing filters such as low pass, high pass, band pass, notch, and moving averaging filters. The specific use of Fourier Transform (FT) is not mentioned in the study, but FT is a commonly used tool in signal processing for frequency analysis and spectral characterization. It is possible that the researchers may have utilized FT for frequency analysis, spectral analysis, or filter design purposes. Overall, the study focused on improving the quality of ECG signals by mitigating noise from various sources encountered in analog recording circuitry.

2. Transform Domain Techniques for Denoising of Signals: A Review, International Journal for Modern Trends in Science and Technology, 7(11): 18-21, 2021

One of the important research areas in signal processing is removing noise from noisy signals. Denoising of signal addresses the problem of removing noise from noisy signal. This paper reviews various transform domain techniques for denoising of noisy signals. An over view on DFT, DCT and DWT are also presented in this paper. Simulation results are carried out using MATLAB by considering various signals. From the results it is observed that the significant amount of noise is removed from the noisy signals.

3. Audio Denoising Based on Short Time Fourier Transform, Indonesian Journal of Electrical Engineering and Computer Science, Vol. 9, No. 1, January 2018

This paper presents a novel audio de-noising scheme for speech signals using Short Time Fourier Transform (STFT). It focuses on adaptively estimating environmental noise and applies STFT techniques to remove noise components from the signals. The effectiveness of the scheme is evaluated using metrics such as Signal to Noise Ratio (SNR) and Peak Signal to Noise Ratio (PSNR).

IMPLEMENTATION

Denoising Data with FFT

Steps

1. Signal Generation: In this example, a clean signal `fclean` is generated by adding two sinusoidal components of frequencies 50 Hz and 120 Hz.

2. Adding Noise: Gaussian noise is added to the clean signal, creating a noisy signal `f`. The noise is generated using the `randn` function and scaled by a factor of 2.5.

3. Visualization: The first subplot visualizes the noisy signal (cyan) overlaid with the clean signal (black). The second subplot will display the filtered signal, and the third subplot will show the power spectral density (PSD) of the signals.

4. FFT and PSD Calculation: The noisy signal `f` is transformed using the FFT algorithm (`fft` function) to obtain the frequency domain representation `fhat`. The power spectral density (PSD) is calculated by taking the squared magnitude of `fhat` divided by the total number of samples.

5. Frequency Axis and Indices: The frequency axis is created using the sampling rate `dt` and the number of samples `n`. The indices where the PSD is greater than 100 are identified as potential noise components.

6. Filtering in the Frequency Domain: The PSD is modified by zeroing out the values corresponding to the identified noise components. The frequency domain representation `fhat` is also modified by zeroing out the frequencies corresponding to the noise components.

7. Inverse FFT: The modified frequency domain representation `fhat` is transformed back to the time domain using the inverse FFT (`ifft` function), resulting in the denoised signal `ffilt`.

8. Visualization of Results: The second subplot displays the denoised signal `ffilt`, and the third subplot shows the PSD of both the noisy and filtered signals.

This approach leverages the properties of the FFT to analyze and manipulate the frequency content of the signal for denoising purposes. By suppressing or removing the noise components in the frequency domain, a cleaner version of the original signal can be obtained in the time domain.

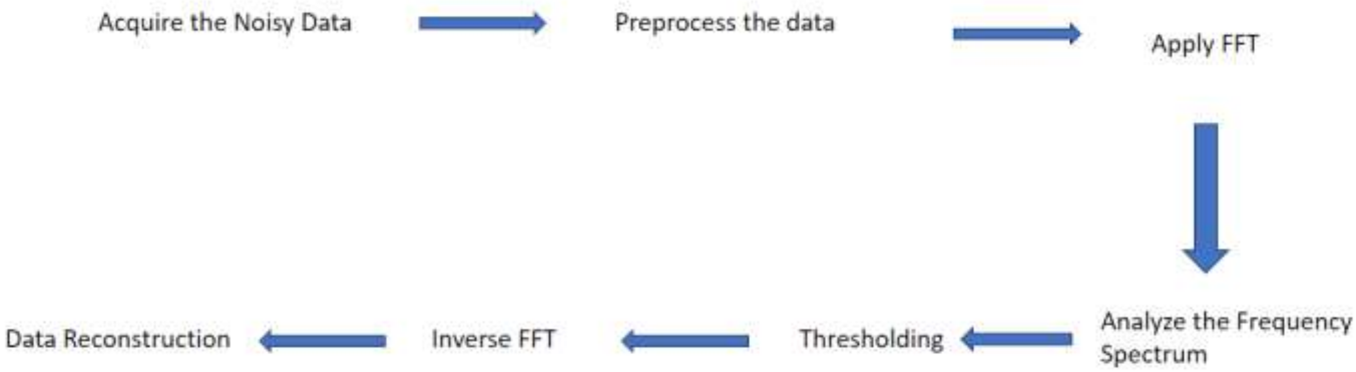


Fig 1. process of Denoising Data

CODE

```
clc
clear all
dt = 0.001;
t = 0:dt:1;
fclean = sin(2*pi*50*t) + sin(2*pi*120*t);
f = fclean + 2.5*randn(size(t));

figure;
set(gcf,'Position',[1500 200 1000 800]) % Adjust figure size
subplot(3,1,1)
plot(t,f,'c','LineWidth',3), hold on
plot(t,fclean,'k','LineWidth',2.5)
l1 = legend('Noisy','Clean');
set(l1,'FontSize',16) % Adjust legend font size
ylim([-10 10]);
set(gca,'FontSize',16) % Adjust axis font size

n = length(t);
fhat = fft(f,n);
PSD = fhat .* conj(fhat) / n;
freq = 1 / (dt*n) * (0:n);
L = 1:floor(n/2);

subplot(3,1,3);
set(gca,'FontSize',12) % Adjust axis font size
plot(freq(L),PSD(L),'c','LineWidth',3)
hold on
set(gca,'FontSize',16) % Adjust axis font size
```

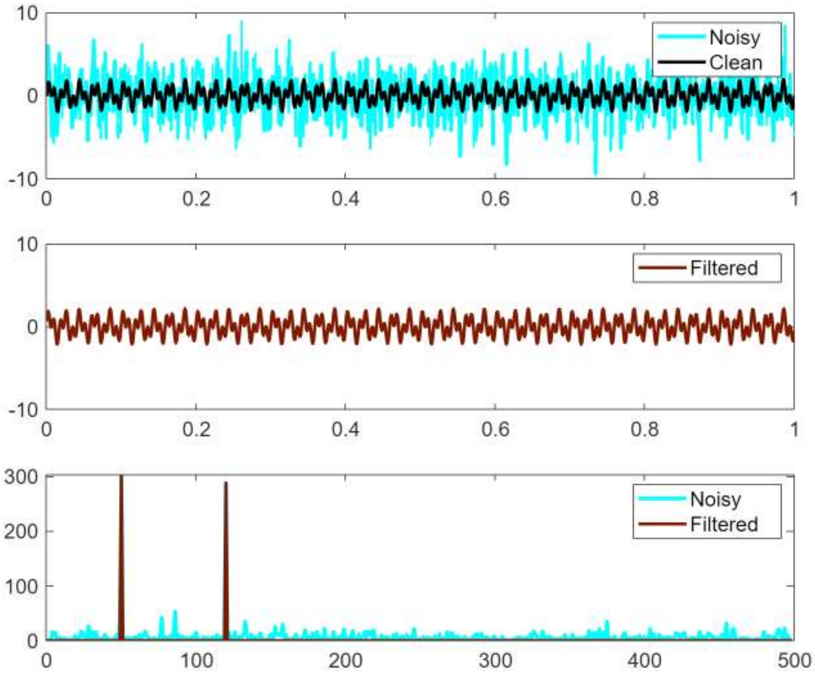
```
indices = PSD > 100;
PSDclean = PSD .* indices;
fhat = indices .* fhat;
ffilt = ifft(fhat);

plot(freq(L),PSDclean(L),'-','Color',[0.5 0.1 0],'LineWidth',2.5)
l1 = legend('Noisy','Filtered');
set(l1,'FontSize',16) % Adjust legend font size

subplot(3,1,2);
set(gca,'FontSize',12) % Adjust axis font size
plot(t,ffilt,'-','Color',[0.5 0.1 0],'LineWidth',2.5)
l1 = legend('Filtered');
set(l1,'FontSize',16) % Adjust legend font size
ylim([-10 10]);
set(gca,'FontSize',16) % Adjust axis font size
```

RESULT

OUTPUT



1. Subplot 1: Noisy and Clean Signals

- This subplot shows the overlaid plots of the noisy signal (`f`) and the clean signal (`fclean`).

- The cyan line represents the noisy signal, while the black line represents the clean signal.

- This plot helps visualize the original signal contaminated with Gaussian noise.

2. Subplot 2: Filtered Signal

- This subplot displays the denoised signal (`ffilt`).

- The denoised signal is obtained by applying the inverse FFT to the modified frequency domain representation.

- The plot shows the denoised signal as a red line.

- By filtering out the noise components in the frequency domain, a cleaner version of the original signal is obtained in the time domain.

3. Subplot 3: Power Spectral Density (PSD)

- This subplot presents the power spectral density (PSD) of the signals.
- The cyan line represents the PSD of the noisy signal, while the orange line represents the PSD of the filtered signal.
- The PSD represents the distribution of power across different frequencies in the signal.
- Initially, the PSD of the noisy signal shows peaks corresponding to the noise components.
- After filtering out the noise components in the frequency domain, the PSD of the filtered signal has reduced power in those noise frequency regions.

REFERENCES

1. ECG Denoising Using MATLAB, International Journal of Scientific & Engineering Research, Volume 4, Issue 5, May-2013

2. Transform Domain Techniques for Denoising of Signals: A Review, International Journal for Modern Trends in Science and Technology, 7(11): 18-21, 2021

3. Audio Denoising Based on Short Time Fourier Transform, Indonesian Journal of Electrical Engineering and Computer Science, Vol. 9, No. 1, January 2018