Institute of Information Technology (IIT)

Jahangirnagar University

Lab Report: 07

<u>Submitted by:</u>

Name: Zannat Hossain Tamim

Roll No: 1970

Lab Date: 28-08-2023

Submission Date: 03-09-2023

EXPERIMENT NO : 07

NAME OF THE EXPERIMENT:

Noise Reduction and Signal Smoothing with Moving Average Filters.

OBJECTIVE:

1.To demonstrate the effectiveness of a moving average filter in reducing noise and smoothing a noisy input signal, emphasizing the impact of varying window sizes on the degree of smoothing.

2.To provide practical experience in implementing signal processing techniques, showcasing the application of Python programming and the use of libraries like NumPy and Matplotlib for signal analysis and visualization.

APPARATUS:

1.Jupyter Notebook

THEORY :

Noise reduction and signal smoothing with moving average filters are fundamental techniques in signal processing and data analysis. Moving average filters work by calculating the average of neighboring data points within a specified window, making them effective tools for reducing random noise and enhancing the clarity of underlying signal trends. The window size is a crucial parameter, where larger windows provide stronger noise reduction but may smooth out finer signal details, while smaller windows offer milder smoothing but preserve finer features. This technique finds wide application in various fields, such as finance, audio processing, and sensor data analysis, where improving data quality and extracting meaningful information from noisy signals are essential tasks.

PROGRAM

```
import numpy as np

import matplotlib.pyplot as plt



def moving_average_filter(input_signal, window_size):

    output_signal = np.convolve(input_signal, np.ones(window_size)/window_size, mode='same')

    return output_signal
```

```
# Generate a noisy input signal (sinusoidal signal with added noise)

t = np.linspace(0, 1, 1000)

input_signal = np.sin(2*np.pi*5*t) + 0.5 * np.random.normal(size=1000)


# Apply a moving average filter with window size 10

window_size = 10

smoothed_signal = moving_average_filter(input_signal, window_size)


# Plot the original signal and the smoothed signal

plt.figure(figsize=(10, 6))

plt.plot(t, input_signal, label='Original Signal')

plt.plot(t, smoothed_signal, label=f'Smoothed (Window Size = {window_size})')

plt.legend()

plt.xlabel('Time')

plt.ylabel('Amplitude')

plt.title('Smoothing a Noisy Signal using Moving Average Filter')

plt.grid(True)

plt.show()
```

RESULT

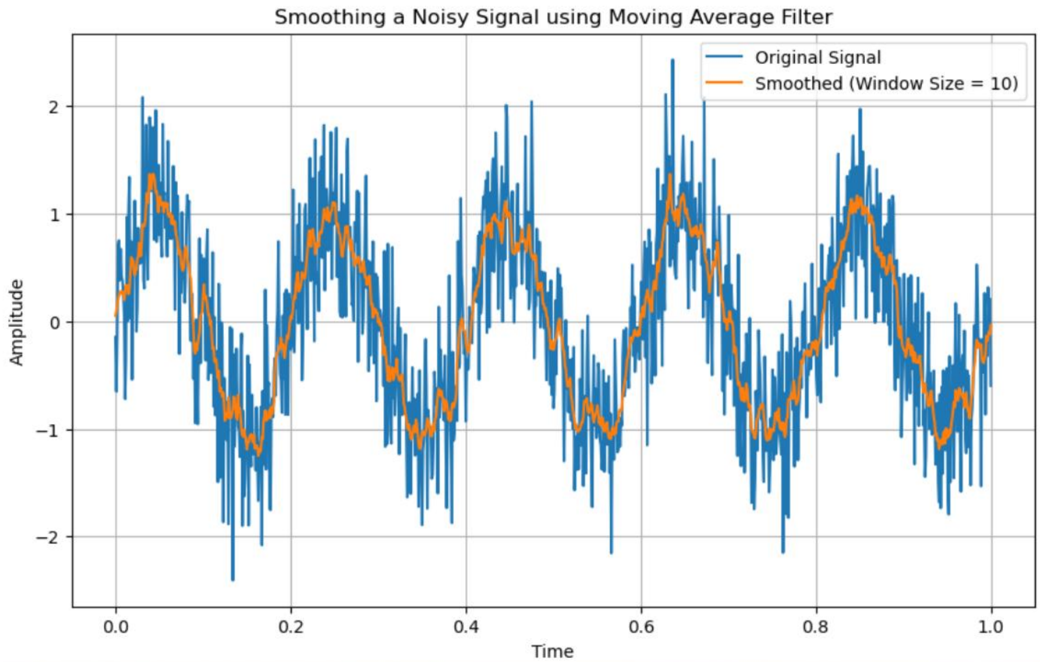

fig:Smoothing a Noisy Signal using Moving Average Filter

DISCUSSION

The experiment revolves around the practical implications of using moving average filters to process noisy signals. The experiment showcases the significance of window size selection in achieving the desired balance between noise reduction and signal preservation. Larger window sizes, as demonstrated, effectively reduce noise, resulting in a smoother signal. However, this comes at the cost of potentially losing intricate signal details. Conversely, smaller window sizes are less aggressive in noise reduction, ensuring better preservation of fine-grained features within the signal. This observation underscores the importance of tailoring the filter's parameters to the specific requirements of the application.

CONCLUSION

In conclusion, the code effectively demonstrates the application of a moving average filter to improve the quality of a noisy signal. It highlights the importance of choosing an appropriate window size for smoothing, with larger windows providing more significant noise reduction. This technique is valuable in various fields, including signal processing, audio processing, and data analysis, where noise reduction and feature preservation are essential.

REFERENCE

[1]The Scientist and Engineer's Guide to Digital Signal Processing, Available: https://www.analog.com/media/en/technical-documentation/dsp-book/dsp_book_ch15.pdf[Accessed: Aug. 31, 2023]