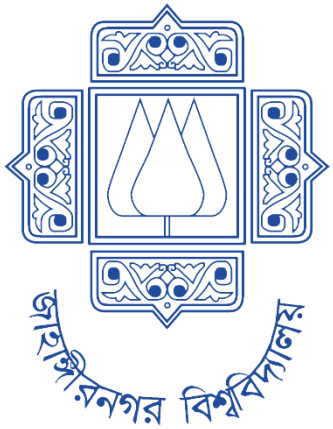
Institute of Information Technology (IIT)

Jahangirnagar University



Lab Report: 07

Submitted by:

Name: Md. Shakil Hossain

Roll No: 2023

Lab Date: 28-08-2023

Submission Date: 04-09-2023

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

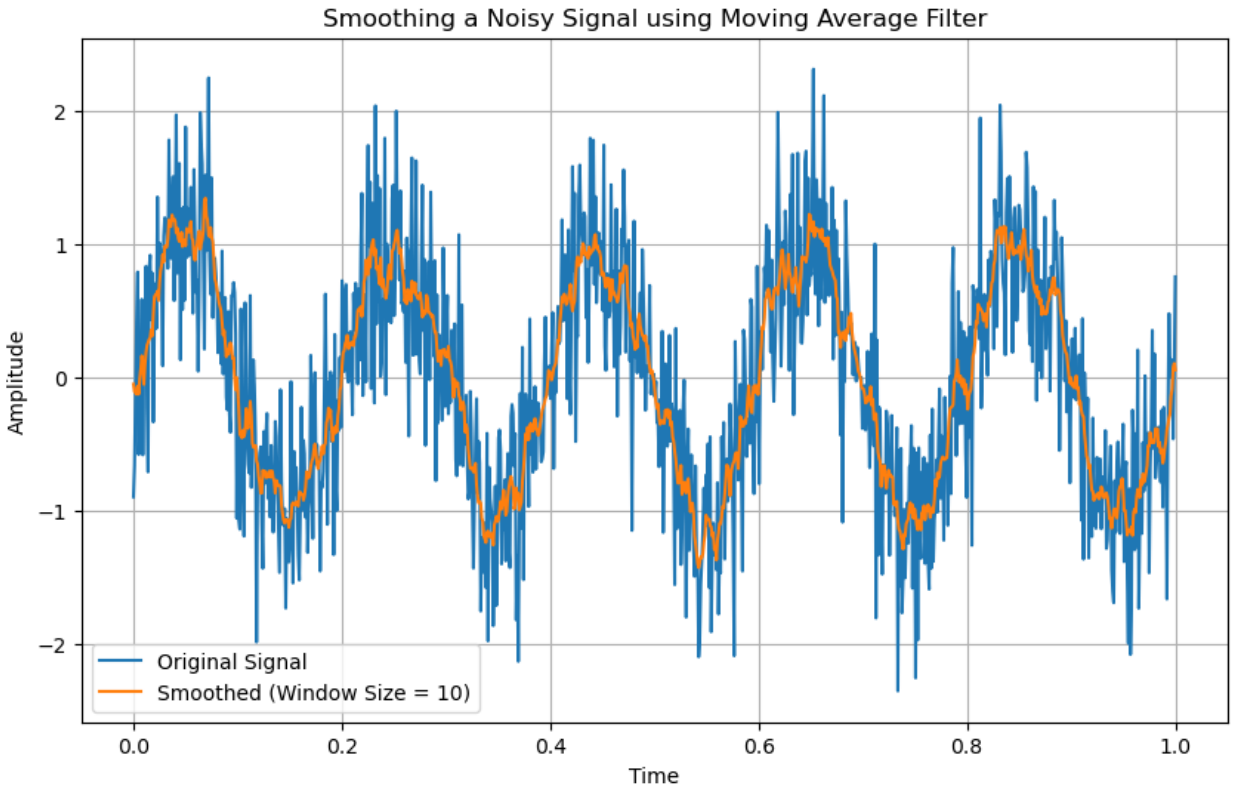


Fig: Smoothing a Noisy Signal using Moving Average Filter

DISCUSSION

The experiment focuses on the applications of processing noisy data with moving average filters. The experiment demonstrates the importance of choosing the right window size in order to strike the ideal balance between noise reduction and signal preservation. As seen, larger window sizes significantly lower noise, resulting in a smoother signal. This could result in the loss of precise signal nuances, though. Smaller window sizes, on the other hand, result in less aggressive noise reduction, which better protects the signal's fine-grained properties. This finding emphasizes how crucial it is to match the filter's parameters to the particular needs of the application.

CONCLUSION

As a result, the code successfully illustrates how to apply a moving average filter to enhance the quality of a noisy signal. It emphasises how crucial it is to pick the right window size for smoothing, with bigger windows offering a more noticeable noise reduction. This method is useful in a variety of domains where noise reduction and feature preservation are important, such as signal processing, audio processing, and data analysis.

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]