A blue and white logo

AI-generated content may be incorrect.

Ministry of Education and Research of the Republic of Moldova

Technical University of Moldova

Department of Software and Automation Engineering

**REPORT**

Individual work No. 1

**Discipline**: Signal Processing

Elaborated: Berco Andrei, FAF - 221

Checked: Railean Serghei

Chișinău 2025

**Purpose of the work**

Generation of noise and its filtering using the Discrete-Time System "M-point Moving Average System.”

**Theoretical Notes**

A system is defined as any device or algorithm that performs operations on a signal.

A **Discrete-Time System** is any device or algorithm that influences a discrete-time signal, called the **Input Signal** or **Excitation – x(n)**, according to well-defined rules to obtain another discrete-time signal called the **Output Signal – y(n)** or **Response**.

The **input-output** relationship consists of mathematical expressions or fixed rules that define the connection between the input and output signals. The exact internal structure of the system is often unknown or ignored.

A system is called **Static** or **Memoryless** if the output signal at any moment **n** depends only on the input signal at the same moment **n**, and not on previous or future moments. Otherwise, the system is **Dynamic** or **with memory**. If the output signal at a given moment **n** depends on the input signal over the interval **n-N** to **n**, then the system has memory of duration **N**. If **N = 0**, the system is **Static**.

A system is called **time-invariant** if its input-output characteristics do not depend on time. If we shift the input signal by **k** units – **x(n-k)** – the output will also be shifted by **k** units: **y(n-k)**.

A system is called **linear** if it satisfies the **superposition principle** – meaning that the system's response to the sum of multiple input signals is equal to the sum of the system's responses to each individual input signal.

**Practical Tasks**

1. Study of Random Processes
   1. **White noise** with a Gaussian dependence is generated using the rand procedure. Generate a random process as follows

A screen shot of a computer program

AI-generated content may be incorrect.

Figure 1: Code for task 1.1

A graph of a white noise

AI-generated content may be incorrect.

Figure 2: Result for 1.1

Parameters:

*Ts = 0.01* sets the sampling interval to 0.01 seconds.

*t = np.arange(0, 5, Ts)* creates an array of time values ranging from 0 to 5 seconds, with a step of 0.01 seconds.

*x1 = np.random.rand(len(t))* generates an array of random values (white noise) with the same length as the time array t. The values are uniformly distributed between 0 and 1.

The resulting plot shows a random signal fluctuating rapidly over time, which is characteristic of white noise. The x-axis represents time in seconds (0 to 5), and the y-axis represents amplitude (0 to 1).

This plot visually demonstrates the concept of white noise, which has equal intensity at different frequencies. White noise is commonly used in various fields such as signal processing and communications.

* 1. Replace the plot function with hist to represent the histogram of the generated noise. Before doing this, change the time range to 1 and swap the variables t and x1.

A screen shot of a computer program

AI-generated content may be incorrect.

Figure 1.3: Code for task 1.2

A graph of blue lines

AI-generated content may be incorrect.

Figure 1.4: result for task 1.2

*plt.hist(x1, bins=100, density=True)* creates a histogram with 100 bins. The density=True parameter normalizes the histogram, so the area under the histogram sums to 1, turning the count into a probability density function.

The resulting histogram shows the distribution of the amplitudes of the white noise signal. The x-axis represents amplitude values ranging from 0 to 1, and the y-axis represents the frequency (normalized) of these amplitude values.

The histogram reveals that the white noise has a relatively uniform distribution of amplitudes between 0 and 1. The bars of the histogram are fairly evenly distributed, indicating that the white noise is uniformly random. This uniformity is a characteristic property of white noise, where each amplitude value within the specified range is equally likely to occur. The grid lines make it easier to visualize the distribution pattern.

* 1. Repeat Step 1.1 for Ts = 0.001 and Generate a New Noise Signal x2.

A screen shot of a computer program

AI-generated content may be incorrect.

Figure 1.5: Code for task 1.3

A blue sound wave with white text

AI-generated content may be incorrect.

Figure 1.6: result for task for 1.3

**Parameters**:

* *Ts2 = 0.001* sets the sampling interval to 0.001 seconds, which is ten times smaller than the previous sampling interval.
* *t2 = np.arange(0, 5, Ts2)* creates an array of time values ranging from 0 to 5 seconds, with a step of 0.001 seconds.

*x2 = np.random.rand(len(t2))* generates an array of random values (white noise) with the same length as the time array t2. The values are uniformly distributed between 0 and 1

The resulting plot shows a much denser distribution of amplitude values over time, which is characteristic of white noise. The x-axis represents time in seconds (0 to 5), and the y-axis represents amplitude (0 to 1).

The plot demonstrates a random signal fluctuating rapidly over time with a higher resolution due to the smaller sampling interval. This higher resolution allows for more detailed analysis of the white noise properties. The increased density of data points shows a continuous and smooth pattern of randomness.

* 1. Represent the Histogram of the Generated Noise x2 from Step 1.3.

A computer screen shot of a code

AI-generated content may be incorrect.

Figure 1.7: Code for task 1.4

A graph with blue lines

AI-generated content may be incorrect.

Figure 1.8: Result for task 1.4

The histogram reveals that the white noise has a relatively uniform distribution of amplitudes between 0 and 1. The bars are fairly evenly distributed, indicating that the white noise is uniformly random, a characteristic property of white noise. The normalized density allows for a clear visualization of the distribution pattern.

The higher number of bins (500) provides a more detailed view of the distribution, giving a smoother representation of the underlying random process. This detailed histogram helps in understanding the statistical properties of the white noise generated with the higher sampling rate

* 1. Design a Second-Order Digital Filter with a Natural Frequency of 1 Hz, Apply It to Signal x1, and Display the Output Signal.

A screen shot of a computer program

AI-generated content may be incorrect.

Figure 1.9: Code for task 1.5

A blue and white sound wave

AI-generated content may be incorrect.

Figure 1.10: Original White Noise for task 1.5

A graph of noise with blue lines

AI-generated content may be incorrect.

Figure 1.11: Filtered result for task 1.5

The filtered signal (y1) is smoother than the original white noise signal. This indicates that the filter is effectively reducing the random fluctuations. The amplitude variations are more controlled and less erratic compared to the unfiltered noise. This suggests the filter is attenuating the noise.

The plot effectively illustrates how a second-order digital filter processes white noise, resulting in a smoother and more stable signal.

* 1. Repeat p 1.5 for Ts = 0.001 and generated noise x2

A screen shot of a computer program

AI-generated content may be incorrect.

Figure 1.12: Code for task 1.6

A blue graph with white text

AI-generated content may be incorrect.  
Figure 1.13: Original White Noise for task 1.6

A graph showing noise with blue lines

AI-generated content may be incorrect.

Figure 1.14: Filtered result for task 1.6

The filtered signal is significantly smoother than the original white noise. The filter reduces the random fluctuations, resulting in a more stable signal. The amplitude variations are more controlled, and the signal appears less erratic compared to the original noise. The grid and legend provide a clear visualization of the filtered signal.

The filtered noise plot demonstrates the effectiveness of the second-order filter in smoothing and attenuating the noise, producing a signal with reduced amplitude variations over time. This illustrates the filter's capability to enhance signal quality by reducing noise.

1. Filtering noise-affected signals using a MAF filter
   1. Generate an original signal not affected by noise s(m)

A screenshot of a computer program

AI-generated content may be incorrect.

Figure 2.1: Code for task 2.1

A graph of a signal

AI-generated content may be incorrect.

Figure 2.2: Result doe task 2.2

* The x-axis represents the **time index** m, ranging from 0 to 49.
* The y-axis represents the **amplitude** of the signal s.
* **Initial Increase**: The amplitude starts at 0 for m=0 and increases as m increases.
* **Peak Amplitude**: The signal reaches its peak amplitude around m=10.
* **Exponential Decay**: After reaching the peak, the amplitude gradually decreases due to the exponential decay factor (0.9m).  
  1. Generate a noise, using the rand function by adding to point 2.1 the noise d=rand(1,length(m))-0.5.

A screen shot of a computer program

AI-generated content may be incorrect.

Figure 2.3: Code for task 2.2

A graph showing a number of signals

AI-generated content may be incorrect.

Figure 2.4: Result for task 2.2

The plot is a visual representation of a random signal, showcasing how the amplitude varies over different time indices.

* 1. Represent both of these signals in continuous form on a single graph, using the plot function.

A screen shot of a computer code

AI-generated content may be incorrect.

Figure 2.5: Code for task 2.3

A graph with a blue line

AI-generated content may be incorrect.

Figure 2.6: Result for task 2.3

The graph visually compares the behaviour of an original signal with a random signal over time, highlighting the differences in their patterns and amplitudes. The original signal shows a clear, structured rise and fall pattern, while the random signal displays irregular and unpredictable fluctuations.

* 1. Represent the sum of these two signals x = s + d and plot the resulting signal x and the initial signal s on a single graph, using the plot function.

A screen shot of a computer program

AI-generated content may be incorrect.

Figure 2.7: Code for task 2.4

A graph with blue and orange lines

AI-generated content may be incorrect.

Figure 2.8: Result for task 2.4

**Original Signal (Blue Line)**: This line represents the original signal without any noise. It has a smooth, sinusoidal shape that increases initially, reaches a peak, and then decreases gradually over time.

**Noisy Signal (Orange Line)**: This line represents the same signal but with added noise. The noisy signal fluctuates around the original signal, showing irregular and random variations due to the noise. The overall shape follows the original signal's trend, but with added randomness.

This visual comparison helps in understanding the impact of noise on signal processing and the importance of filtering or noise reduction techniques.

* 1. Design a moving average filter (MAF) with the parameters y = filter(b,1,x), b = ones(M,1)/M, and specify M = 3 in advance. Filter the signal affected by noise. Plot the filtered signal y, the noisy signal x, and the initial signal s on a single graph, using the plot function

A screenshot of a computer program

AI-generated content may be incorrect.

Figure 2.9: Code for task 2.5

A graph with lines and numbers

AI-generated content may be incorrect.

Figure 2.10: Result for task 2.5

**Original Signal (Blue Line)**: Displays the original data points without any noise.

**Noisy Signal (Orange Line)**: Shows the original signal with added noise, leading to fluctuations and irregularities.

**Filtered Signal (Green Line)**: Demonstrates the effect of applying a moving average filter, which smooths out the noise, making the signal resemble the original one more closely.

The plot demonstrates the effectiveness of the moving average filter in reducing noise while preserving the overall shape of the original signal. This is relevant for signal processing applications where noise reduction is essential for accurate data analysis and interpretation.

* 1. Repeat step 2.5 for M = 5 and M = 10. Compare the results obtained.

A screen shot of a computer program

AI-generated content may be incorrect.

Figure 2.11: Code for task 2.6

A graph with lines and numbers

AI-generated content may be incorrect.

Figure 2.12.1: Result for task 2.6 with M = 5

A graph with lines and numbers

AI-generated content may be incorrect.

Figure 2.12.2: Result for task 2.6 with M = 10

Increasing the value of M from 5 to 10 in the moving average filter results in a smoother filtered signal, reducing more noise and better preserving the original signal's shape.

The plot with M=10 demonstrates better noise reduction compared to the plot with M=5, showing the trade-off between the filter window size and the smoothness of the signal.

These plots effectively illustrate the impact of different moving average filter sizes on a noisy signal, highlighting the benefits of using a larger filter window for more effective noise reduction.

* 1. Repeat step 2.5 for another signal – s = 2 \* sawtooth(3 \* pi \* m + pi / 6) and change the time step to a smaller one – m = 0:0.001:R (R = 1) and set M = 20. Plot the filtered signal y, the noisy signal x, and the initial signal s on a single graph, using the plot function.

A screenshot of a computer program

AI-generated content may be incorrect.

Figure 2.13: Code for task 2.7

A graph with orange and green lines

AI-generated content may be incorrect.

Figure 2.14: Result for task 2.7

**Original Signal (Blue Line)**:

* The original signal is a sawtooth wave, generated using the sawtooth function from the scipy.signal module.
* It exhibits a linear rise and a sharp fall, characteristic of a sawtooth waveform.

**Noisy Signal (Orange Line)**:

* The original sawtooth wave with added random noise.
* The noise causes irregular fluctuations, making the signal less smooth.

**Filtered Signal (Green Line)**:

* This is the result of applying a Moving Average Filter with a window size of M=20 to the noisy signal.
* The filter smooths out the noise, making the signal closely resemble the original sawtooth wave.

This plot visually demonstrates the effectiveness of a Moving Average Filter in reducing noise and preserving the underlying structure of the original signal. By averaging the data points within a window size of M=20, the filter smooths out random fluctuations and enhances the clarity of the signal.

* 1. Repeat step 2.7 for M = 50 and M = 100. Compare the results obtained.

A screen shot of a computer program

AI-generated content may be incorrect.

Figure 2.15: Code for task 2.8

A graph with orange lines and green lines

AI-generated content may be incorrect.

Figure 2.16.1: Result for task 2.8 for M = 50

A graph with orange lines and green lines

AI-generated content may be incorrect.

Figure 2.16.2: Result for task 2.8 for M = 100

**Plot 1: Original, Noisy, and Filtered Signals (M=50)**

* **Original Signal (Blue Line)**: Represents the unfiltered sawtooth signal.
* **Noisy Signal (Orange Line)**: Displays the sawtooth signal with added random noise.
* **Filtered Signal (Green Line, M=50)**: Shows the result of applying a moving average filter with a window size of 50. The filtered signal is smoother than the noisy signal, with reduced random fluctuations, but some noise still persists.

**Plot 2: Original, Noisy, and Filtered Signals (M=100)**

* **Original Signal (Blue Line)**: Again, represents the unfiltered sawtooth signal.
* **Noisy Signal (Orange Line)**: Shows the noisy version of the original signal.
* **Filtered Signal (Green Line, M=100)**: Shows the result of applying a moving average filter with a window size of 100. This filtered signal is significantly smoother compared to the one with M=50M=50, closely resembling the original signal with minimal noise.

Increasing the window size MM results in a smoother filtered signal. The larger the window size, the more points are averaged, which better reduces noise.

While larger MM values yield smoother signals, they may also slightly lag behind rapid changes in the original signal due to the increased averaging.