

Отчет по ПЗ №2

по дисциплине «Основы теории и применения цифровой обработки данных»

Тема: Генерация шума с равномерным распределением

Вариант: 2

Студент:

Башев Григорий Алексеевич

C19-501

Группа

ФИО

Руководитель:

Заева Маргарита Анатольевна

ФИО

Москва, 2023

ХАРАКТЕРИСТИКА ВЫБРАННЫХ СРЕДСТВ РЕАЛИЗАЦИИ

Для выполнения данной лабораторной работы был выбран язык Python, и библиотеки Matplotlib, numpy, csv.

ВЫВОД РАСЧЁТНЫХ ХАРАКТЕРИСТИК

Исходные данные(вариант 2):

Частота дискретизации = 100 кГц

Время моделирования = 100 мс

Диапазон от $x_{\min} = -1$ до $x_{\max} = 1$

Распределение нормальное

Расчёт

$$M = (x_{\min} + x_{\max}) / 2 = 0$$

$$D = (x_{\max} - x_{\min}) * (x_{\max} - x_{\min}) / 12 = 0.3333333333333333$$

ГРАФИЧЕСКОЕ ПРЕДСТАВЛЕНИЕ СИГНАЛА

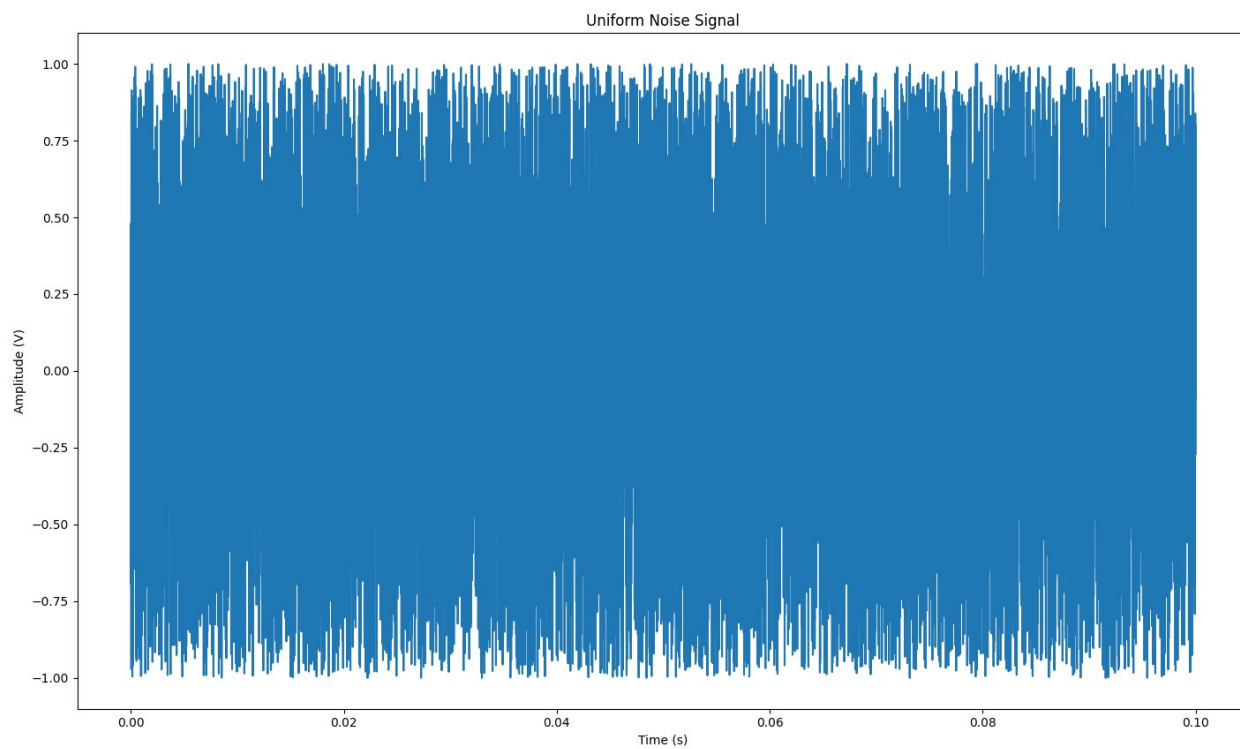


Рис 1 График всего сигнала

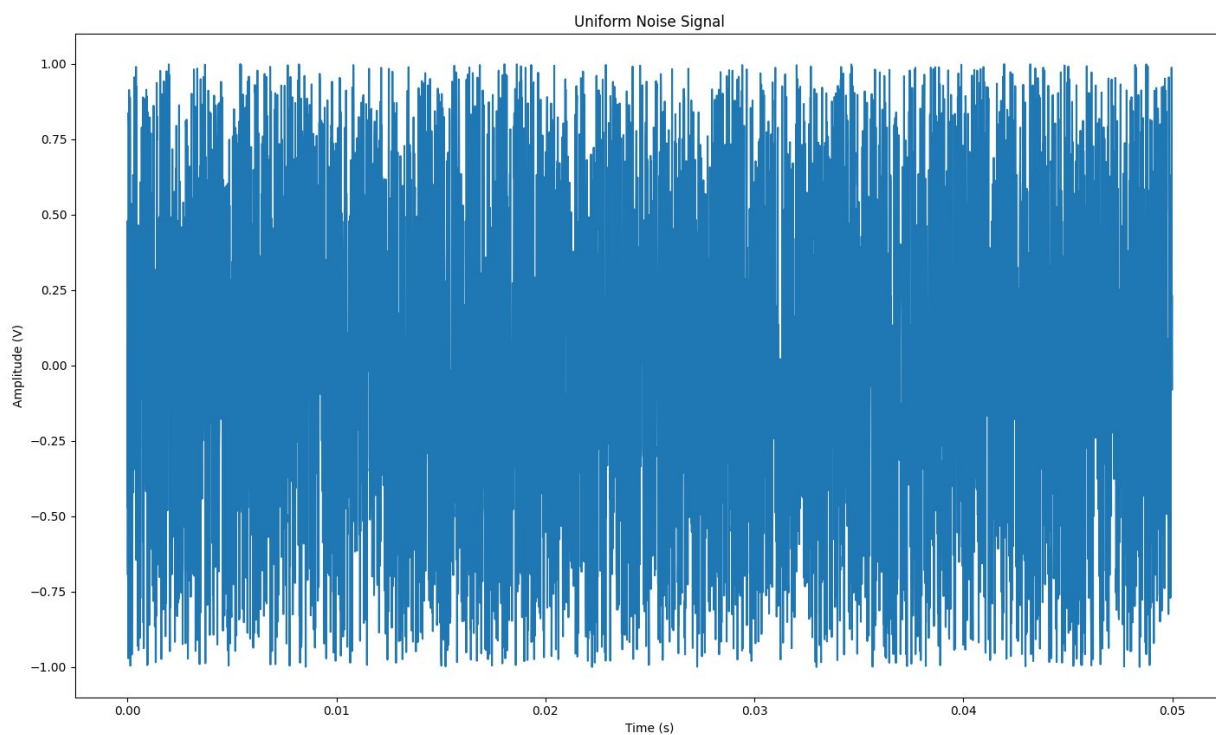


Рис 2 График 10% сигнала

МАТОЖИДАНИЕ И ДИСПЕРСИЯ

```
# расчёт мат ожидания и дисперсии
mean = np.mean(signal)
variance = np.var(signal)
print("Mean: ", mean)
print("Variance: ", variance)
Mean: -0.003983652721449422
Variance: 0.32972212347869934
```

Полученный результат совпадает с теоретическими расчетами

```

import numpy as np
import matplotlib.pyplot as plt
import csv
sampling_freq = 100000
duration = 0.1
# Задам зерно случайности
np.random.seed(12)
# Генерируем сигнал
num_samples = int(sampling_freq * duration)

signal = np.random.uniform(low=-1, high=1, size=num_samples)

t = np.arange(0, duration, 1/sampling_freq)

with open('signal.csv', 'w') as f:
    writer = csv.writer(f)
    writer.writerow([duration, sampling_freq])
    np.savetxt(f, signal, delimiter=",")

# Чтение из файла
with open('signal.csv', 'r') as f:
    reader = csv.reader(f)
    row1 = next(reader)
    duration = float(row1[0])
    sampling_freq = float(row1[1])
    signal = np.genfromtxt(f)

# Графики
plt.plot(t, signal)
plt.xlabel("Time (s)")
plt.ylabel("Amplitude (V)")
plt.title("Uniform Noise Signal")
plt.show()
plt.plot(t[:int(t.size * 0.5)], signal[:int(signal.size * 0.5)])
plt.xlabel("Time (s)")
plt.ylabel("Amplitude (V)")
plt.title("Uniform Noise Signal")
plt.show()

# расчёт мат ожидания и дисперсии
mean = np.mean(signal)
variance = np.var(signal)
print("Mean: ", mean)
print("Variance: ", variance)

```

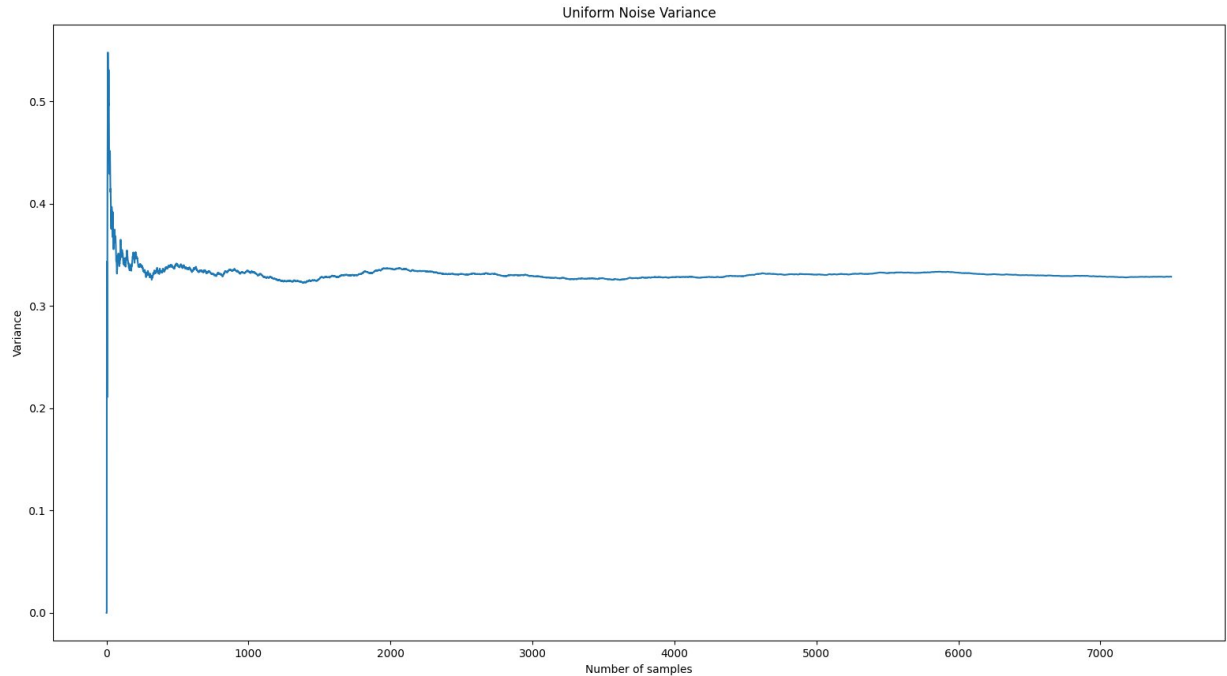
ЗАКЛЮЧЕНИЕ

В ходе выполнения лабораторной работы был промоделирован шум с нормальным распределением. Так же с помощью библиотеки Matplotlib был построены графики зависимости уровня сигнала от времени в разных масштабах. Затем посредством numpy было найдено среднее значение сигнала и дисперсия сигнала, которое совпало с теоретическим.

ДОП 1

Задание: График зависимости дисперсии от длины выборки

После 10000 отсчетов значение дисперсии устанавливается в теоретическое значение.



```
x = np.arange(100000)
var = np.zeros(100000)

for i in range(1, 100000):
    var[i] = np.var(signal[:i])

plt.plot(np.arange(7500), var[:7500])
plt.xlabel("Number of samples")
plt.ylabel("Variance")
plt.title("Uniform Noise Variance")
plt.show()
```