

Отчет по ПЗ №1

**по дисциплине «Основы теории и применения цифровой
обработки данных»**

Тема: Генерация дискретного сигнала

Вариант: 2

Студент:

Башев Григорий
Алексеевич

Группа

C19-
501

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):

Амплитуда = 0.2 В

Смещение = 0.5 В

Частота = 13427.73438 Гц

Начальная фаза = 10 град

Частота дискретизации = 100 кГц

Время моделирования = 100 мс

Расчёт

Количество отсчетов для моделирования =

$\text{sampling_freq} * \text{duration} = 100000 * 0,1 = 10000$

$\text{phase} = \text{np.deg2rad}(10) = 0.17453292519943295 \text{ Рад}$

ГРАФИЧЕСКОЕ ПРЕДСТАВЛЕНИЕ СИГНАЛА

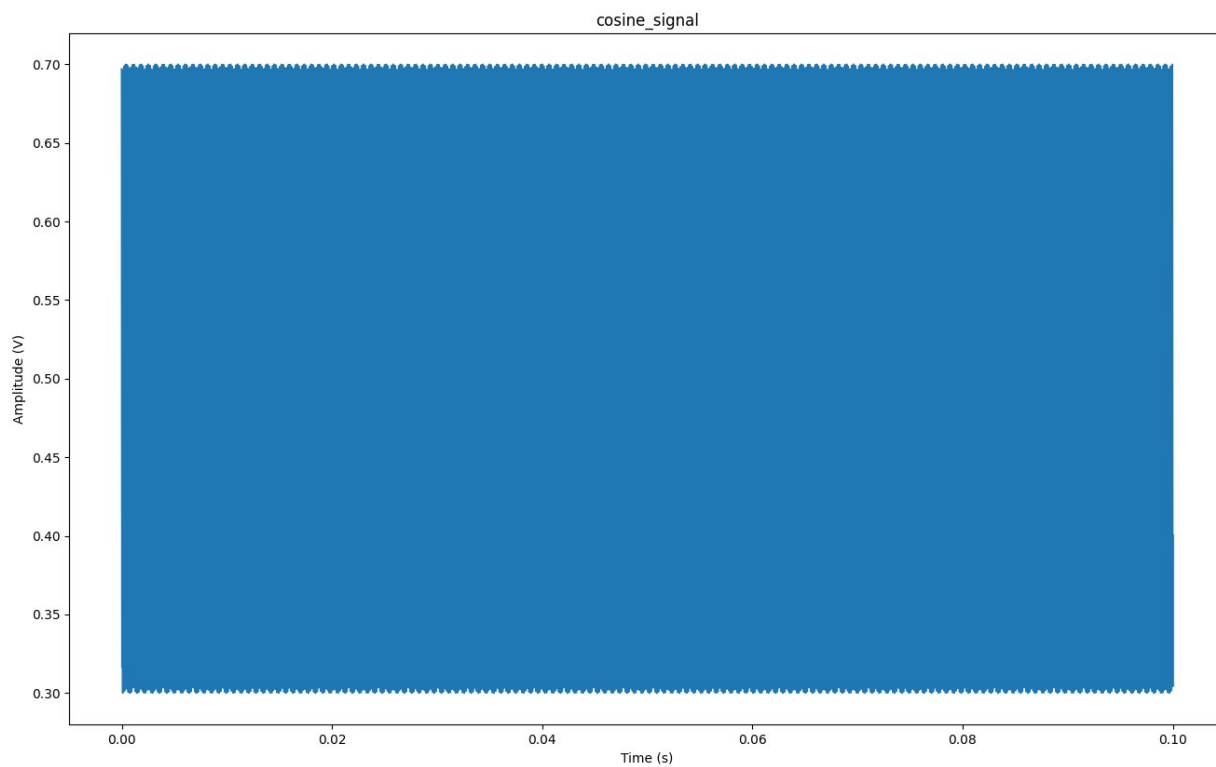


Рис 1 График всего сигнала

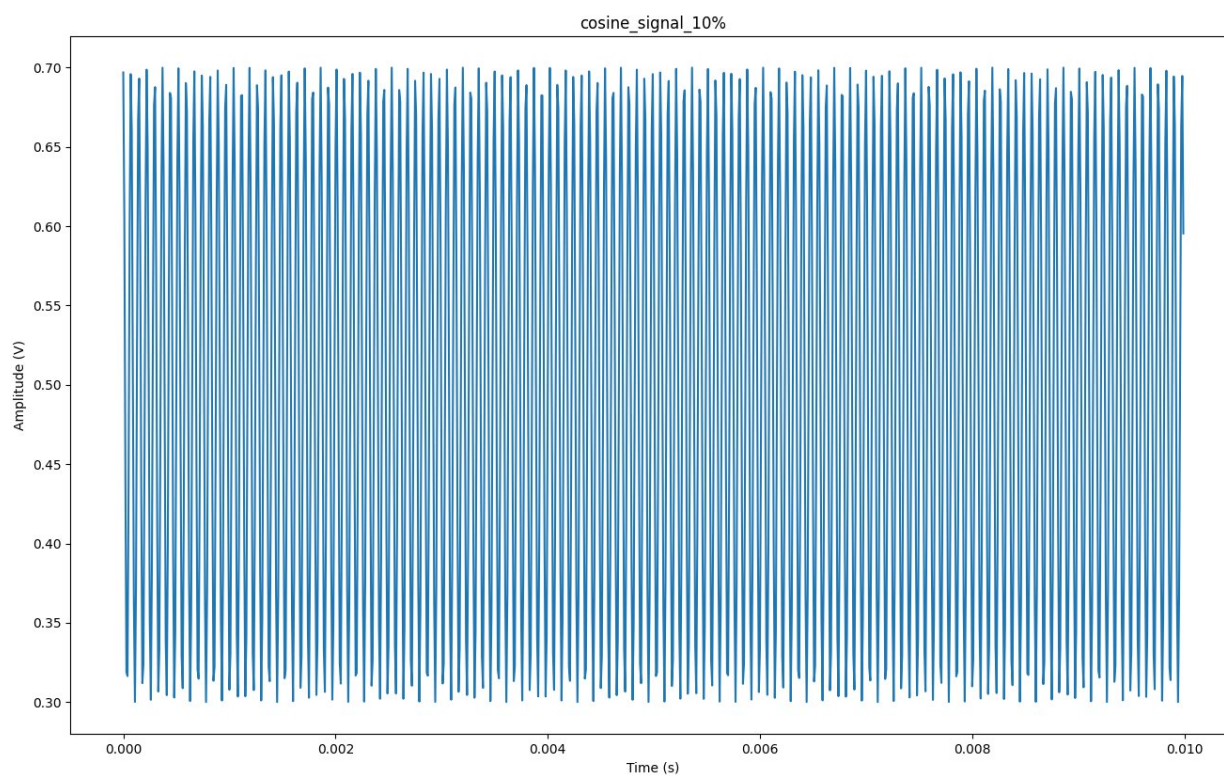


Рис 2 График 10% сигнала

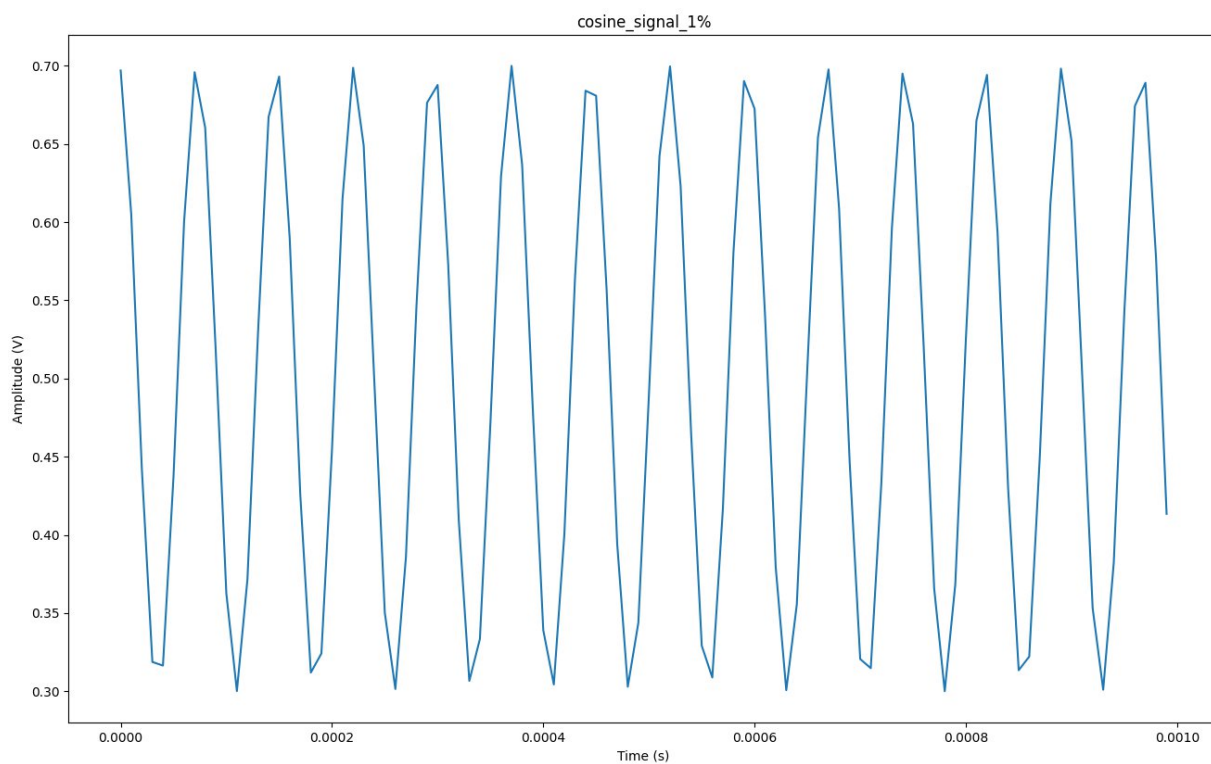


Рис 3 График 1% сигнала

НАХОЖДЕНИЕ СРЕДНЕГО ЗНАЧЕНИЯ

```
>>> print(np.mean(signal))
```

```
>>> 0.49998167757202866
```

Среднее значение сигнала приблизительно равно сдвигу, что и ожидалось для косинуса.

```

import numpy as np # Внедрение зависимостей
import matplotlib.pyplot as plt
import csv

# Начальные данные
amplitude = 0.2
frequency = 13427.73438
phaseDeg = 10
phase = np.deg2rad(phaseDeg)
sampling_freq = 100000
duration = 0.1
offset = 0.5

# Подготовка значений по осям
t = np.arange(0, duration, 1/sampling_freq)
signal = offset + amplitude * np.cos(2*np.pi*frequency*t + phase)

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

# Чтение из файла
with open('cosine_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
t = np.arange(0, duration, 1/sampling_freq)
plt.plot(t, signal)
plt.xlabel("Time (s)")
plt.ylabel("Amplitude (V)")
plt.title("cosine_signal")
plt.show()
plt.plot(t[:int(t.size * 0.1)], signal[:int(signal.size * 0.1)])
plt.xlabel("Time (s)")
plt.ylabel("Amplitude (V)")
plt.title("cosine_signal_10%")
plt.show()
plt.plot(t[:int(t.size * 0.01)], signal[:int(signal.size * 0.01)])
plt.xlabel("Time (s)")
plt.ylabel("Amplitude (V)")
plt.title("cosine_signal_1%")
plt.show()

# Расчёт среднего значения
print("Среднее значение сигнала:")
print(np.mean(signal))

```

ЗАКЛЮЧЕНИЕ

В ходе выполнения лабораторной работы был промоделирован дискретный сигнал. Так же с помощью библиотеки Matplotlib был построен график зависимости уровня сигнала от времени в разных масштабах. Затем посредством numpy было найдено среднее значение сигнала, которое совпало с ожидаемым.

ДОП 1

Задание: Сравнить значение в 1мс, записанное в csv со теоритическим значением

```
80 3.000078552691061162e-01
81 3.683873227638512771e-01
82 5.250233619334725432e-01
83 6.648792840018720174e-01
84 6.941707782978328023e-01
85 5.932556118686949098e-01
86 4.298052796347744398e-01
87 3.134259856868972793e-01
88 3.221591293300323233e-01
89 4.501484561814472207e-01
90 6.115671337802124352e-01
91 6.981713410269605458e-01
92 6.518861980185612914e-01
93 5.037495054227506541e-01
94 3.530984768143493602e-01
95 3.009563842936174294e-01
96 3.822885702764452498e-01
97 5.425554565671218965e-01
98 6.742855880288215475e-01
99 6.891436307618321599e-01
100 5.771661075316489864e-01
101 4.134426868449617709e-01
102 0.3077626999131027330e-01
103 3.309928256694153470e-01
104 4.675554518209002475e-01
105 6.258746796308248816e-01
106 6.997851109768882250e-01
107 6.397240333712144178e-01
108 4.859670867589159782e-01
109 3.416203036603274734e-01
110 3.034794675524428165e-01
111 3.971209857114073594e-01
112 5.597509120471180211e-01
113 6.823131886513080779e-01
114 6.826202438028644570e-01
115 5.604661730588370583e-01
116 3.977648142530562092e-01
117 3.036201263398508043e-01
118 3.411634699989525843e-01
119 4.852191032718672914e-01
120 6.391864813622264441e-01
VISUAL <signal.csv utf-8 | unix | csv 1% 102:1
-- VISUAL -- 24
```

$$a_1 = 0.5 + 0.2 * \cos(2\pi * 13427.73438 * 0.001 + 0.17453292519943295)$$

$$= 0.30776269991310273$$

$$a_2 = 0.3077626999131027330$$

$$a_1 = a_2$$

ДОП 2

Задание: Рассчитать энергию и норму сигнала

```
norm = np.linalg.norm(signal)
print("Норма встроенная функция", norm)

norm = 0
for s in signal:
    norm += s ** 2
norm = np.sqrt(norm)

print("Норма самостоятельный расчет", norm)

energy = np.sum(np.square(signal))
print("Энергия", energy)
print("Корень из энергии", np.sqrt(energy))

Норма встроенная функция 51.95984786396478
Норма самостоятельный расчет 51.959847863964754
Энергия 2699.825790046365
Корень из энергии 51.959847863964775
```