|  |
| --- |
| МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ |
| ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ |
| **«Национальный исследовательский ядерный университет «МИФИ»** |
| ИНСТИТУТ ИНТЕЛЛЕКТУАЛЬНЫХ КИБЕРНЕТИЧЕСКИХ СИСТЕМ |
| КАФЕДРА «КОМПЬЮТЕРНЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ» (№12) |

**Отчет по ПЗ №1**

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Тема:** Генерация дискретного сигнала | | | |
| Вариант: 2 | | | |
| Студент: | Башев Григорий Алексеевич | Группа | С19-501 |
|  | 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 =

sampling\_freq \* duration = 100000 \* 0,1 = 10000

phase = np.deg2rad(10) = 0.17453292519943295 Рад

### ГРАФИЧЕСКОЕ ПРЕДСТАВЛЕНИЕ СИГНАЛА

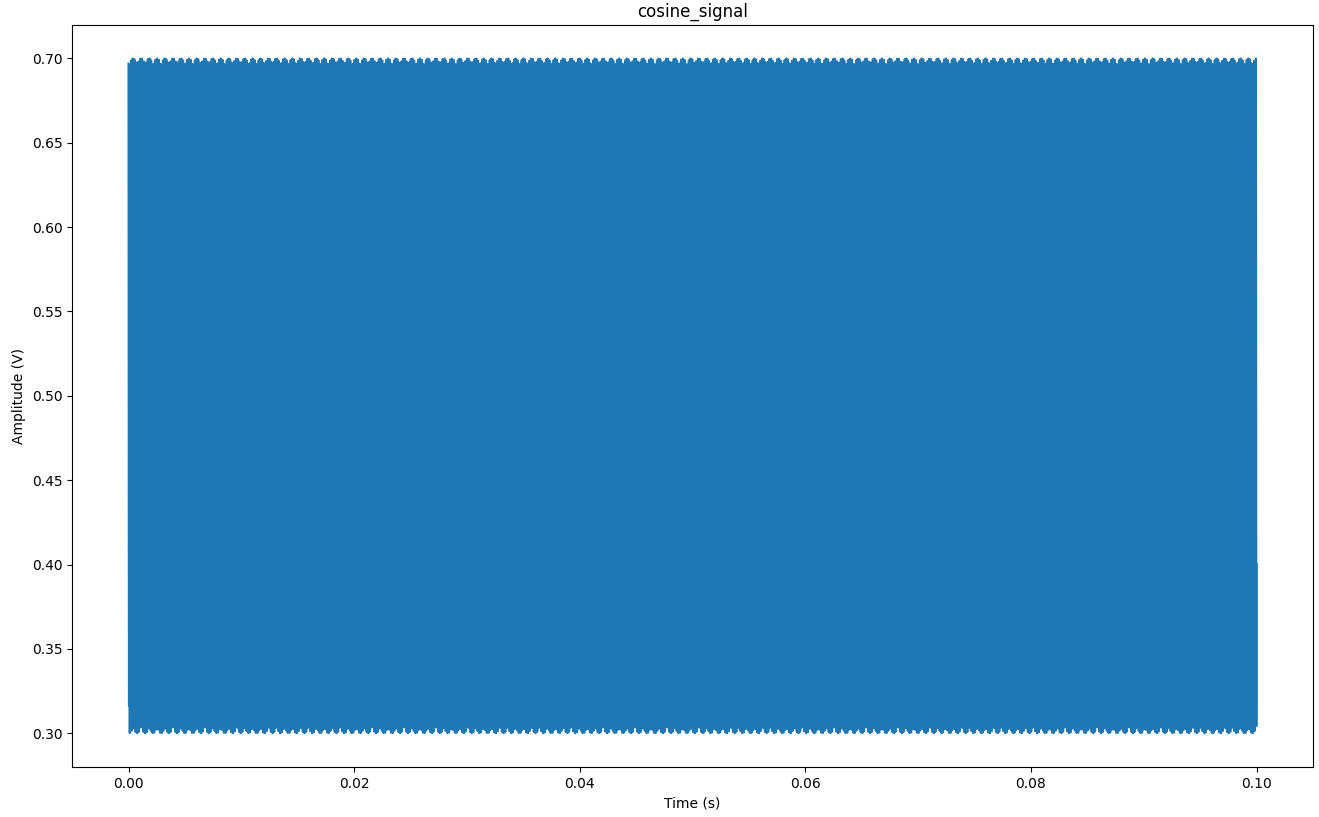


Рис 1 График всего сигнала

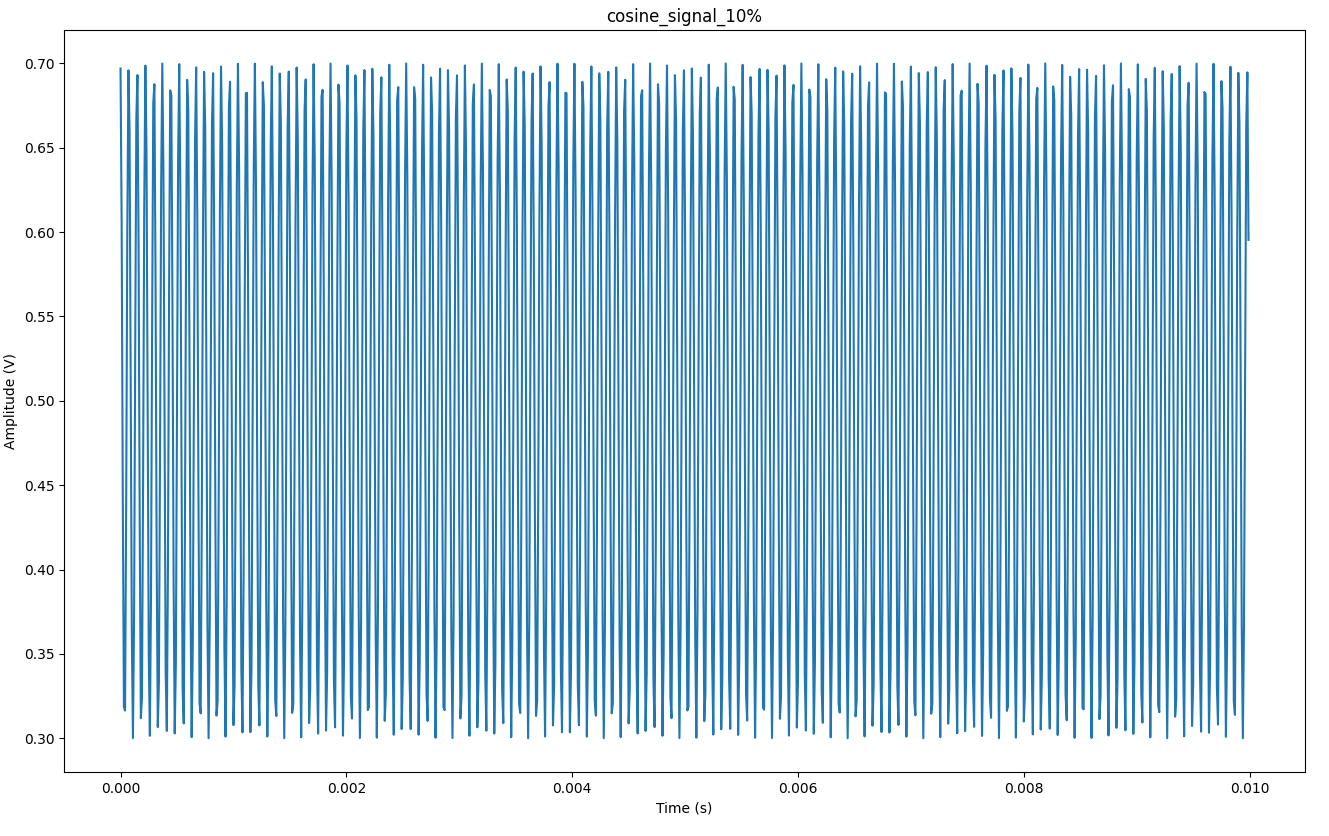


Рис 2 График 10% сигнала

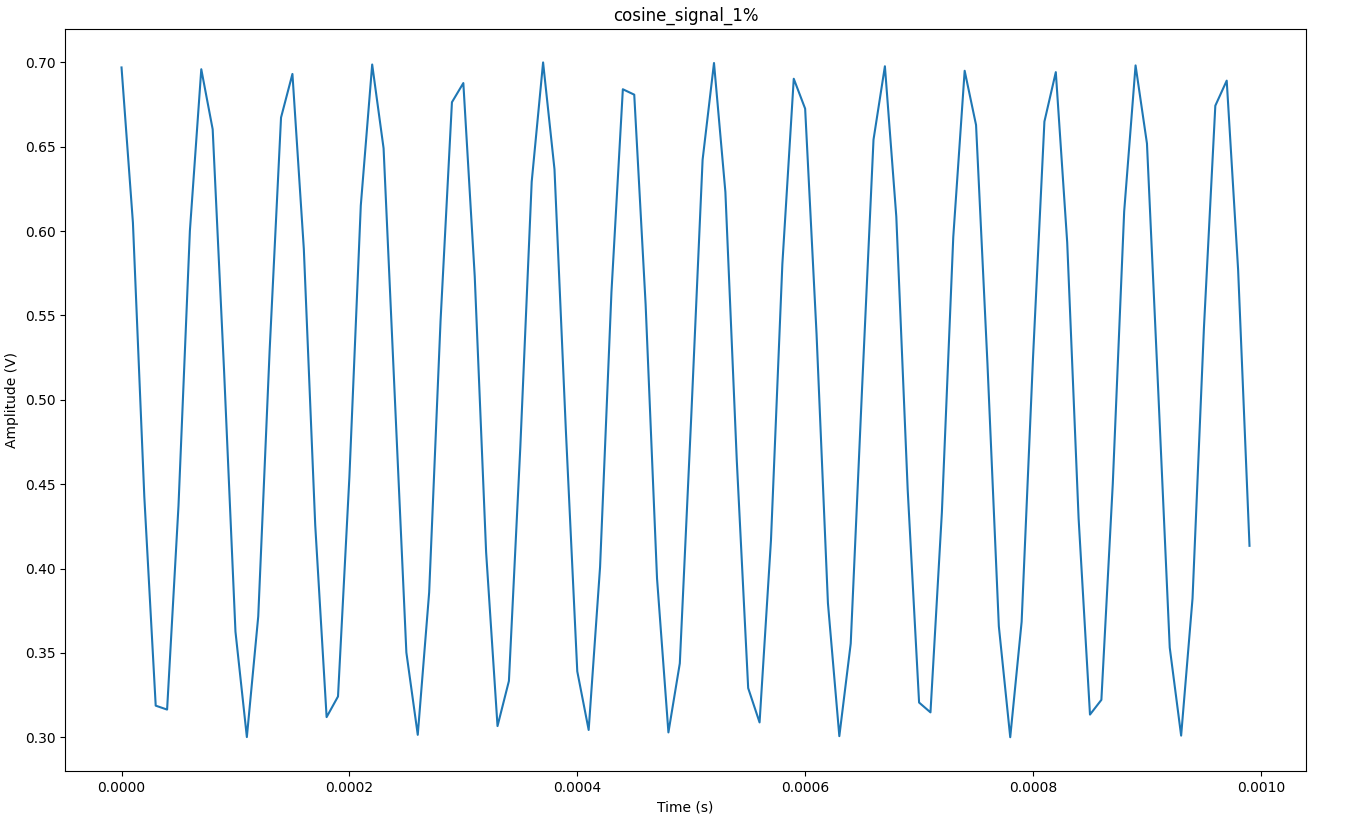


Рис 3 График 1% сигнала

### НАХОЖДЕНИЕ СРЕДНЕГО ЗНАЧЕНИЯ

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

>>> 0.49998167757202866

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

**import nump**y **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:")

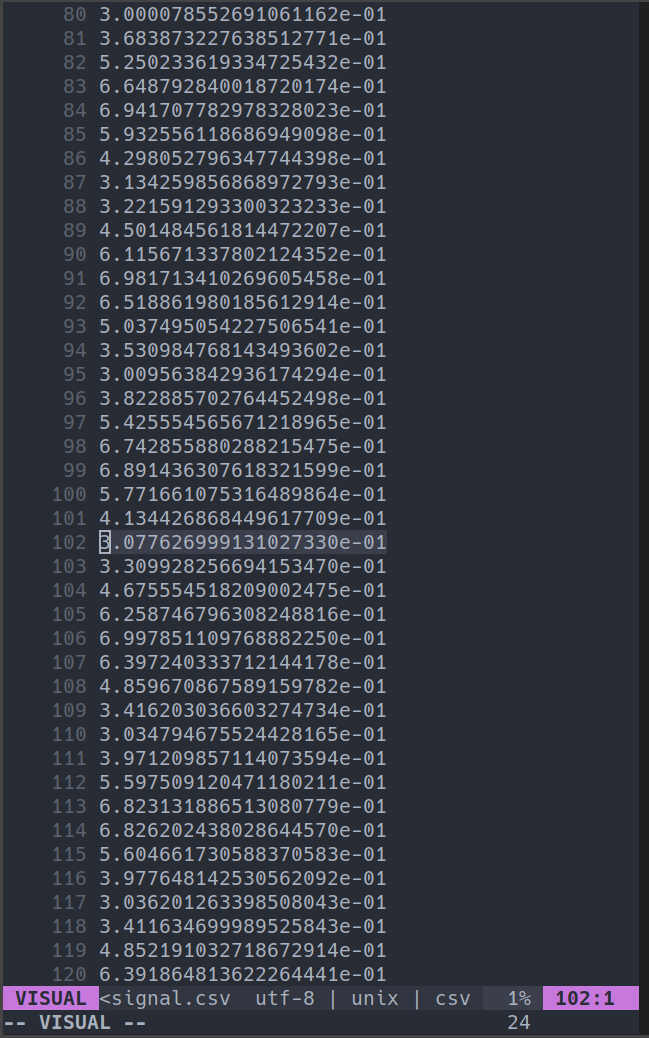
print(np.mean(signal))

### ЗАКЛЮЧЕНИЕ

В ходе выполнения лабораторной работы был промоделирован дискретный сигнал. Так же с помощью библиотеки Matplotlib был построены графики 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 теоритеческим значением



= 0.30776269991310273

0.3077626999131027330

### 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