**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра математического обеспечения и применения ЭВМ**

отчет

**по лабораторной работе №1**

**по дисциплине «Цифровая обработка сигналов»**

Тема: Дискретные сигналы

|  |  |  |
| --- | --- | --- |
| Студентка гр. 8383 |  | Гречко В.Д. |
| Студентка гр. 8383 |  | Аверина О.С. |
| Преподаватель |  | Сучков А.И. |

Санкт-Петербург

2021

**Цель работы.**

Изучить математическое описание дискретных сигналов и овладеть программными средствами их моделирования.

**Основные теоретические положения.**

В теории цифровой обработки сигналов (ЦОС) принято разделять операции дискретизации по времени и квантования по уровню. Полагая операцию квантования отсутствующей, изучают дискретные сигналы и линейные дискретные системы (ЛДС), а затем, отдельно, – эффекты нелинейной операции квантования.

Дискретным называют сигнал, дискретный по времени и непрерывный по состоянию (уровню), который описывается последовательностью чисел бесконечной разрядности или , называемой коротко последовательностью. Значения , , называют дискретным временем, где – период дискретизации, а – дискретным нормированным временем.

Цифровым называют сигнал, дискретный по времени и квантованный по состоянию (уровню), который описывается последовательностью чисел конечной разрядности – квантованной последовательностью или . При компьютерном моделировании под дискретным сигналом условно понимают последовательность чисел максимально возможной разрядности, а под цифровым – последовательность чисел заданной разрядности.

**Постановка задачи.**

С помощью программных средств провести моделирование и анализ дискретных последовательностей. Результаты подкрепить соответствующими графиками и выводами.

Исходные данные:

|  |  |  |
| --- | --- | --- |
| Переменная | Назначение | Значение |
|  | **Длина последовательности** |  |
|  | **Период дискретизации** |  |
|  | **Основание экспоненты** |  |
|  | **Амплитуда гармонического сигнала** |  |
|  | **Частота гармонического сигнала** |  |
|  | **Задержка** |  |
|  | **Амплитуда импульса** |  |
|  | **Начальный момент импульса** |  |
|  | **Длина импульса** |  |
|  | **Амплитуды гармонических сигналов** |  |
|  | **Частоты гармонических сигналов** |  |
|  | **Коэффициенты линейной комбинации гармонических сигналов** |  |

**Выполнение работы.**

1. Смоделировать единичный цифровой импульс с выводом графиков на интервале дискретного времени и дискретного нормированного времени . Пояснить взаимосвязь между дискретным и дискретным нормированным временем и различие между цифровым единичным импульсом и функцией Дирака.

Единичный цифровой импульс :

Интервал дискретного времени .

Интервал дискретного нормированного времени . Графики единичного цифрового импульса на интервалах дискретного времени и дискретного нормированного времени представлены на рис. 1.1 и рис. 1.2 соответственно.

|  |  |
| --- | --- |
|  |  |
| *Рисунок 1.1 – Цифровой единичный импульс на интервале дискретного времени* | *Рисунок 1.2 – Цифровой единичный импульс на интервале дискретного нормированного времени* |

Значения называют *дискретным временем*, а — *дискретным нормированным временем* (тождественно *T = 1*).

Функция Дирака представляет собой бесконечно узкий импульс с бесконечной амплитудой, расположенный при нулевом значении аргумента функции. Сигнал в виде дельта-функции невозможно реализовать физически, в то время как единичный цифровой импульс имеет амплитуду равную 1. Поэтому он – её дискретный аналог.

1. Смоделировать дискретный единичный скачок с выводом графиков на интервале дискретного времени и дискретного нормированного времени . Пояснить соответствие между дискретным единичным скачком и функцией Хэвисайда, а также чему равна частота дискретизации дискретного единичного скачка.

Дискретный единичный скачок :

Интервал дискретного времени

Интервал дискретного нормированного времени . Графики дискретного единичного скачка на интервалах дискретного времени и дискретного нормированного времени представлены на рис. 2.1 и рис. 2.2 соответственно.

|  |  |
| --- | --- |
|  |  |
| *Рисунок 2.1 – Цифровой единичный скачок на интервале дискретного времени* | *Рисунок 2.2 – Цифровой единичный скачок на интервале дискретного нормированного времени* |

Дискретный единичный скачок по смыслу полностью соответствует своему аналоговому прообразу (т.е. функции единичного скачка, иначе функции Хэвисайда).

Частота дискретизации дискретного единичного скачка равна

1. Смоделировать дискретную экспоненциальную функцию с выводом графиков на интервале дискретного времени и дискретного нормированного времени . Пояснить соответствие между дискретной и аналоговой экспонентами.

Дискретная экспоненциальная функция :

Интервал дискретного времени

Интервал дискретного нормированного времени . Графики дискретной экспоненциальной функции на интервалах дискретного времени и дискретного нормированного времени представлены на рис. 3.1 и рис. 3.2 соответственно.

|  |  |
| --- | --- |
|  |  |
| *Рисунок 3.1 – Дискретная экспоненциальная функция на интервале дискретного времени* | *Рисунок 3.2 – Дискретная экспоненциальная функция на интервале дискретного нормированного времени* |

Дискретная экспонента – это результат преобразования аналоговой экспоненты в последовательность отсчетов (дискретную систему).

1. Смоделировать дискретный комплексный гармонический сигнал с выводом графиков вещественной и мнимой частей на интервале времени . Записать данный сигнал в виде комбинации двух вещественных последовательностей.

Дискретный комплексный гармонический сигнал:

Интервал дискретного нормированного времени . Дискретный комплексный гармонический сигнал с выводом графиков вещественной и мнимой частей представлен на рис. 4.1 и рис. 4.2 соответственно.

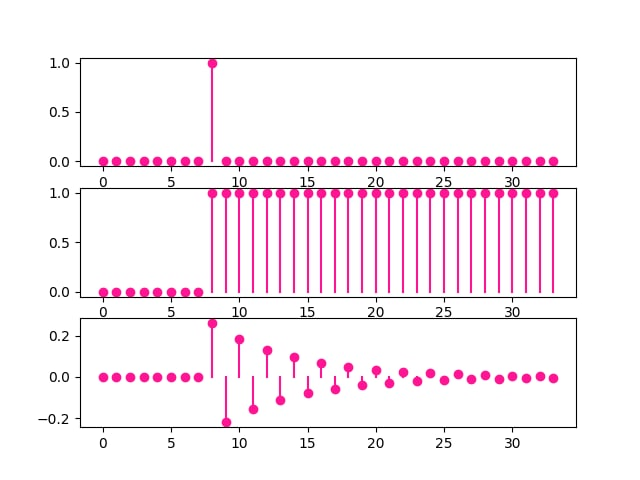
|  |  |
| --- | --- |
|  |  |
| *Рисунок 4.1 – Вещественная часть дискретного комплексного гармонического сигнала* | *Рисунок 4.2 – Мнимая часть дискретного комплексного гармонического сигнала* |

Для записи данного сигнала в виде комбинации двух вещественных последовательностей воспользуемся формулой Эйлера:

Тогда:

1. Вывести графики последовательностей , задержанных на отсчетов, на интервале времени . Записать формулы задержанных последовательностей.

Интервал дискретного нормированного времени . Графики последовательностей , задержанных на отсчетов представлены на рис. 5.



*Рисунок 5 – Графики последовательностей (сверху вниз), задержанных на 9 отсчетов*

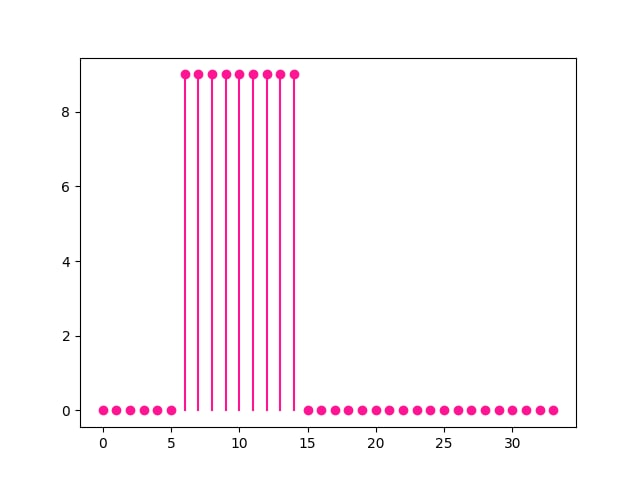
Формулы задержанных последовательностей:

1. Смоделировать дискретный прямоугольный импульс

на основе дискретного единичного скачка с выводом графика на интервале времени . Пояснить как выполняется моделирование импульса.

Дискретный прямоугольный импульс:

Интервал дискретного нормированного времени . Дискретный прямоугольный импульс представлен на рис. 6.



*Рисунок 6 – График дискретного прямоугольного импульса*

Моделирование импульса было реализовано путём задания координат точек, где значения по оси OY в интервале от 7 до 15 были равны 9. Функция отрисовки, использованная при построении графика дискретного единичного скачка, осталась без изменений. Поменялась лишь передача аргументов в данную функцию.

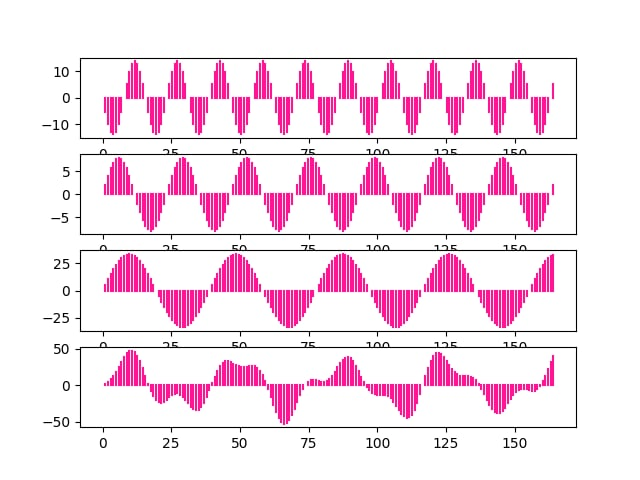
1. Смоделировать линейную комбинацию дискретных гармонических сигналов :

где

с выводом графиков последовательностей и на интервале времени . Вычислить среднее значение, энергию и среднюю мощность последовательности . Пояснить, какие операции при моделировании линейной комбинации сигналов и как определяют указанные характеристики.

Линейная комбинация дискретных гармонических сигналов :

где



*Рисунок 7 – График линейной комбинации дискретных гармонических сигналов с выводом графиков последовательностей (сверху вниз*

Среднее значение

Энергия

Средняя мощность

Для построения графиков 1-3 в каждой точке были подсчитаны значения функции , а затем по этим координатам происходила отрисовка. График 4 был построен путем суммирования рассчитанных координат для графиков 1-3.

Полученные значения и использовались для расчетов указанных характеристик:

среднее значение – сумма значений последовательности, отнесенная к её длине (было реализовано через функцию np.mean)

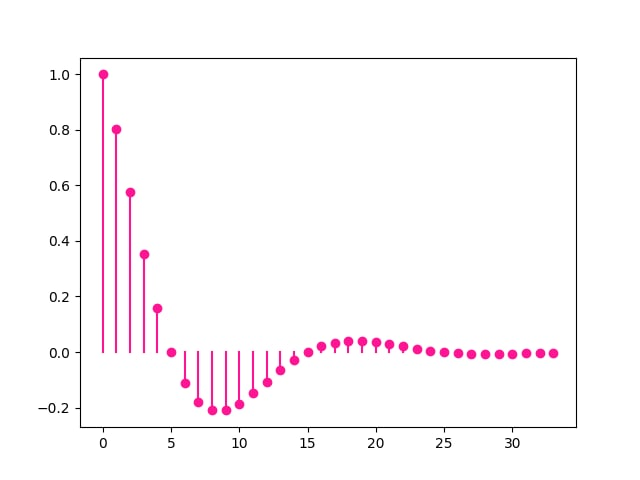
энергия – сумма квадратов ее значений последовательности (было реализовано через функции np.sum и np.power)

средняя мощность — энергия, отнесенная к длине последовательности (было реализовано через деление переменной, хранящей значение энергии, на длину (подсчитывалась через len))

1. Смоделировать дискретную затухающую синусоиду и вывести график на интервале времени . Пояснить операции при моделировании данного сигнала.

Дискретная затухающая синусоида :

Интервал дискретного нормированного времени . Дискретная затухающая синусоида представлена на рис. 8.

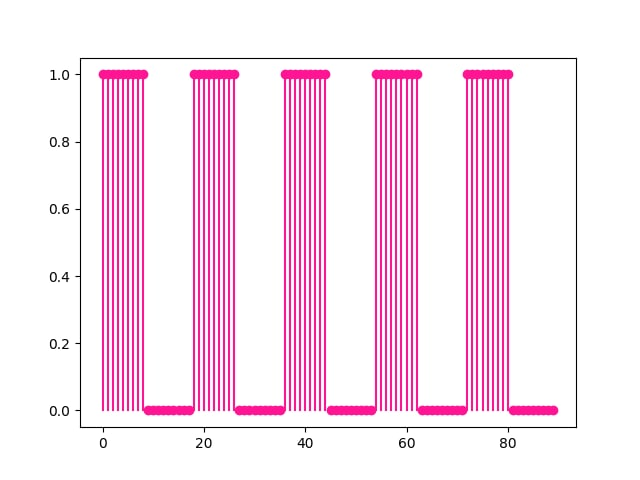


*Рисунок 8 – График дискретной затухающей синусоиды*

Для построения графика были рассчитаны координаты по формуле (1) и были переданы в функцию отрисовки. Расчеты выполнялись с использованием библиотеки numpy.

1. Вывести график пяти периодов периодической последовательности дискретных прямоугольных импульсов амплитуды и длительности с периодом, вдвое большим длительности импульса. Пояснить операции при моделировании периодической последовательности.

График пяти периодов периодической последовательности дискретных прямоугольных импульсов амплитуды и длительности с периодом, вдвое большим длительности импульса представлен на рис. 9.



*Рисунок 9 – График пяти периодов периодической последовательности дискретных прямоугольных импульсов*

Для построения графика была использована реализация дискретного прямоугольного импульса из пункта 1, амплитуда была задана равной 9 и таких периодов было выведено 5 согласно заданию

**Выводы.**

В ходе лабораторной работы были получены навыки моделирования дискретных сигналов средствами языка Python, включая библиотеки numpy и matplotlib, на основе их математического описания, а также были изучены основные теоретические положения по данной теме.

Приложение А

Исходный код программы

import numpy as np

import matplotlib.pyplot as plt

# Смоделировать единичный цифровой импульс δd(k) с выводом графиков на интервале

# дискретного времени nT ∈ [0, (N−1)T] и дискретного нормированного времени n ∈ [0, N−1].

# Пояснить взаимосвязь между дискретным и дискретным нормированным временем и

# различие между цифровым единичным импульсом и функцией Дирака.

nrows, ncols = 2, 1

N = 34

T = 0.0005

a = 0.845

n\_imp = 9

w\_0 = np.pi / 10

# 1)

def func\_1(end\_of\_line, ax, y):

x = np.linspace(0, end\_of\_line, N, endpoint=True)

for i in range(N):

ax.scatter(x[i], y[i], color="deeppink")

ax.plot([x[i], x[i]], [0, y[i]], color="deeppink")

y = [1, \*([0] \* (N - 1))]

# 1.1)

end\_of\_line = (N - 1) \* T

fig = plt.figure()

func\_1(end\_of\_line, plt, y)

plt.show()

# 1.2)

end\_of\_line = (N - 1)

fig = plt.figure()

func\_1(end\_of\_line, plt, y)

plt.show()

# 2)

# Смоделировать дискретный единичный скачок σd(k) с выводом графиков на

# интервале дискретного времени nT∈[0,(N−1)T] и дискретного нормированного

# времени n ∈[0,N-1]. Пояснить соответствие между дискретным единичным

# скачком и функцией Хэвисайда, а также чему равна частота дискретизации

# дискретного единичного скачка.

def func\_2(end\_of\_line, ax):

x = np.linspace(0, end\_of\_line, N, endpoint=True)

for i in range(N):

ax.scatter(x[i], y[i], color="deeppink")

ax.plot([x[i], x[i]], [0, y[i]], color="deeppink")

# 2.1)

y = [1, \*([1] \* (N - 1))]

end\_of\_line = (N - 1) \* T

fig = plt.figure()

func\_2(end\_of\_line, plt)

plt.show()

# 2.2)

end\_of\_line = (N - 1)

fig = plt.figure()

func\_2(end\_of\_line, plt)

plt.show()

# 3)

# Смоделировать дискретную экспоненциальную функцию s1(k) с выводом графиков

# на интервале дискретного времени nT ∈[0,(N−1)T] и дискретного нормированного

# времени n ∈[0,N− 1]. Пояснить соответствие между дискретной и аналоговой экспонентами.

def is\_odd(a):

return bool(a & 1)

def func\_3(end\_of\_line, ax, offset):

x = np.linspace(0, end\_of\_line, N, endpoint=True)

for i in range(offset):

ax.scatter(x[i], 0, color="deeppink")

for i in range(offset, N):

res = np.power(a, i)

if is\_odd(i):

res \*= -1

ax.scatter(x[i], res, color="deeppink")

ax.plot([x[i], x[i]], [0, res], color="deeppink")

# 3.1)

end\_of\_line = (N - 1) \* T

fig = plt.figure()

func\_3(end\_of\_line, plt, 0)

plt.show()

# 3.2)

end\_of\_line = (N - 1)

fig = plt.figure()

func\_3(end\_of\_line, plt, 0)

plt.show()

# 4)

# Смоделировать дискретный комплексный гармонический сигнал s2(k)=C exp(j^ω0k)

# с выводом графиков вещественной и мнимой частей на интервале времени n∈[0,N−1].

# Записать данный сигнал в виде комбинации двух вещественных последовательностей.

def func\_4\_1(end\_of\_line, ax):

x = np.linspace(0, end\_of\_line, N, endpoint=True)

fig = plt.figure()

for i in range(N):

res = 5 \* np.cos(w\_0 \* i)

ax.scatter(x[i], res, color="deeppink")

ax.plot([x[i], x[i]], [0, res], color="deeppink")

def func\_4\_2(end\_of\_line, ax):

x = np.linspace(0, end\_of\_line, N, endpoint=True)

for i in range(N):

res = 5 \* np.sin(w\_0 \* i)

ax.scatter(x[i], res, color="deeppink")

ax.plot([x[i], x[i]], [0, res], color="deeppink")

nrows, ncols = 2, 1

end\_of\_line = (N - 1)

fig = plt.figure()

ax = fig.add\_subplot(nrows, ncols, 1)

func\_4\_1(end\_of\_line, ax)

ax = fig.add\_subplot(nrows, ncols, 2)

func\_4\_2(end\_of\_line, ax)

plt.show()

# 5

# Вывести графики последовательностей δd(k), σd(k) и s1(k), задержанных на m отсчетов,

# на интервале времени n∈[0,N−1]. Записать формулы задержанных последовательностей.

end\_of\_line = (N - 1)

nrows, ncols = 3, 1

offset = 8

fig = plt.figure()

y = [\*([0] \* (offset)), 1, \*([0] \* (N - 8))]

ax = fig.add\_subplot(nrows, ncols, 1)

func\_1(end\_of\_line, ax, y)

y = [\*([0] \* (offset)), 1, \*([1] \* (N - 8))]

ax = fig.add\_subplot(nrows, ncols, 2)

func\_2(end\_of\_line, ax)

ax = fig.add\_subplot(nrows, ncols, 3)

func\_3(end\_of\_line, ax, offset)

plt.show()

# 6

# Смоделировать дискретный прямоугольный импульс s3(k)={U,n0⩽n⩽n0+nimp−1;0,иначе

# на основе дискретного единичного скачка с выводом графика на интервале времени n ∈[0,N−1].

# Пояснить как выполняется моделирование импульса.

y = [\*([0] \* (6)), \*([n\_imp] \* (n\_imp)), \*([0] \* (N - 15))]

end\_of\_line = (N - 1)

fig = plt.figure()

func\_1(end\_of\_line, plt, y)

plt.show()

# 7

end\_of\_line = (5 \* N - 1)

nrows, ncols = 4, 1

offset = 8

def func\_7\_1(end\_of\_line, ax):

x = np.linspace(0, end\_of\_line, 175, endpoint=True)

for i in range(5 \* N):

x\_1 = 5.5 \* np.sin(i \* np.pi/8)

ax\_1 = -2.5 \* x\_1

ax.plot([x[i], x[i]], [0, ax\_1], color="deeppink")

def func\_7\_2(end\_of\_line, ax):

x = np.linspace(0, end\_of\_line, 175, endpoint=True)

for i in range(5 \* N):

x\_2 = 1.7 \* np.sin(i \* np.pi/12)

ax\_2 = 4.7 \* x\_2

ax.plot([x[i], x[i]], [0, ax\_2], color="deeppink")

def func\_7\_3(end\_of\_line, ax):

x = np.linspace(0, end\_of\_line, 175, endpoint=True)

for i in range(5 \* N):

x\_3 = 6.2 \* np.sin(i \* np.pi/20)

ax\_3 = 5.4 \* x\_3

ax.plot([x[i], x[i]], [0, ax\_3], color="deeppink")

def func\_7\_4(end\_of\_line, ax):

x = np.linspace(0, end\_of\_line, 175, endpoint=True)

s4\_list = []

for i in range(5 \* N):

x\_1 = 5.5 \* np.sin(i \* np.pi / 8)

ax\_1 = -2.5 \* x\_1

x\_2 = 1.7 \* np.sin(i \* np.pi / 12)

ax\_2 = 4.7 \* x\_2

x\_3 = 6.2 \* np.sin(i \* np.pi / 20)

ax\_3 = 5.4 \* x\_3

s4 = ax\_1 + ax\_2 + ax\_3

s4\_list.append(s4)

ax.plot([x[i], x[i]], [0, s4], color="deeppink")

return(s4\_list)

fig = plt.figure()

ax = fig.add\_subplot(nrows, ncols, 1)

func\_7\_1(end\_of\_line, ax)

y = [\*([0] \* (offset)), 1, \*([1] \* (N - 8))]

ax = fig.add\_subplot(nrows, ncols, 2)

func\_7\_2(end\_of\_line, ax)

y = [\*([0] \* (offset)), 1, \*([1] \* (N - 8))]

ax = fig.add\_subplot(nrows, ncols, 3)

func\_7\_3(end\_of\_line, ax)

ax = fig.add\_subplot(nrows, ncols, 4)

s4 = func\_7\_4(end\_of\_line, ax)

plt.show()

s4\_mean = np.mean(s4)

s4\_energy = np.sum(np.power(s4, 2))

s4\_power = s4\_energy/len(s4)

print(s4\_mean, s4\_energy, s4\_power)

#8

def func\_8(end\_of\_line, ax):

x = np.linspace(0, end\_of\_line, N, endpoint=True)

fig = plt.figure()

for i in range(N):

res = np.power(abs(a), i) \* np.cos(w\_0 \* i)

ax.scatter(x[i], res, color="deeppink")

ax.plot([x[i], x[i]], [0, res], color="deeppink")

end\_of\_line = (N - 1)

fig = plt.figure()

func\_8(end\_of\_line, plt)

plt.show()

#9

y = [\*([1] \* (n\_imp)), \*([0] \* (n\_imp))] \* 5

end\_of\_line = n\_imp \* 2 \* 5

def func\_9(end\_of\_line, ax):

x = range(end\_of\_line)

for i in range(end\_of\_line):

ax.scatter(x[i], y[i], color="deeppink")

ax.plot([x[i], x[i]], [0, y[i]], color="deeppink")

fig = plt.figure()

func\_9(end\_of\_line, plt)

plt.show()