МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ «ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра математического обеспечения и применения ЭВМ

ОТЧЕТ

по лабораторной работе №2

по дисциплине «Цифровая обработка сигналов»

Тема: Моделирование и анализ линейных комбинаций дискретных сигналов

Студент гр. 9381	________________	Колованов Р.А.
Студент гр. 9381	________________	Семенов А.Н.
Преподаватель	________________	Середа А.-В. И.

Санкт-Петербург

2022

Цель работы.

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

Основные теоретические положения.

Сигнал – это изменяющаяся во времени физическая величина, описываемая функцией времени. Один из параметров этой функции содержит информацию о другой физической величине.

Аналоговый сигнал – сигнал данных, у которого каждый из представленных параметров описывается функцией времени и непрерывным множеством возможных значений (рис. 0, а).

Дискретный сигнал – сигнал, который является прерывистым (в отличие от аналогового) и который изменяется во времени и принимает значения из некоторого конечного дискретного множества (рис. 0, б).

Дискретное время рассматривает значения переменных как происходящие в различных, отдельных точках времени. Зачастую эти точки равноудалены друг от друга и временное расстояние между ними называется периодом. То есть речь идет о значениях nT, где T - период дискретизации.

Дискретное нормированное время – дискретное время, образованное равноудаленными точками с единичным временным расстоянием (с единичным периодом дискретизации). То есть речь идет о значениях n (рис. 0, в).

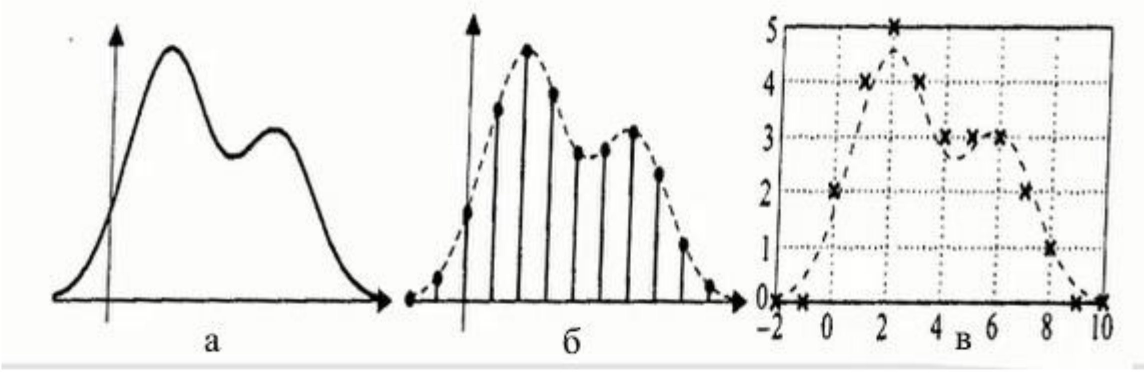

Рисунок 0. Графики сигналов.

Специальные виды детерминированных дискретных сигналов:

1. Единичный цифровой импульс:

$$\delta_d(k) = \begin{cases} 1, & k = 0 \\ 0, & k \neq 0 \end{cases}, \quad k \in \mathrm{Z}$$

2. Дискретный единичный скачок:

$$\sigma_d(k) = \begin{cases} 1, & k \geq 0 \\ 0, & k < 0 \end{cases}, \quad k \in \mathrm{Z}$$

3. Дискретная экспоненциальная функция:

$$s_1(k) = \begin{cases} 0, & k < 0 \\ a^k, & k \geq 0 \end{cases}, \quad k \in \mathrm{Z}$$

4. Дискретный комплексный гармонический сигнал:

$$s_2(k) = C \cdot \exp(j\hat{\omega}_0 k), \quad k \in \mathrm{Z}$$

5. Дискретный прямоугольный импульс:

$$s_3(k) = \begin{cases} U, & n_0 \leq k \leq n_0 + n_{imp} - 1 \\ 0, & \text{иначе} \end{cases}, \quad k \in \mathrm{Z}$$

Характеристики сигналов:

1. Среднее значение:

$$\bar{U} = \frac{1}{T}\int_0^T s(t)dt$$

2. Энергия сигнала:

$$E = \int_0^T s^2(t)dt$$

3. Мгновенная мощность сигнала:

$$p(t) = s^2(t)$$

4. Средняя мощность сигнала:

$$P_{cp} = \frac{1}{T}\int_0^T s^2(t)dt$$

Постановка задачи.

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

Выполнение работы.

Исходные данные для лабораторной работы:

Переменная	**Назначение**	**Значение**
$N_{бр}$	Номер бригады	4
N	Длина последовательности	34
a	Основание экспоненты	0.82
C	Амплитуда гармонического сигнала	5
$\widehat{\omega}_0$ (рад)	Частота гармонического сигнала	$\frac{\pi}{10}$
U	Амплитуда импульса	4
n_0	Начальный момент импульса	7
n_{imp}	Длина импульса	9
B_1	Амплитуды гармонических сигналов	5.5
B_2		1.7
B_3		6.2
$\widehat{\omega}_1$	Частоты гармонических сигналов	$\frac{\pi}{8}$
$\widehat{\omega}_2$		$\frac{\pi}{12}$
$\widehat{\omega}_3$		$\frac{\pi}{20}$
a_1	Коэффициенты линейной комбинации гармонических сигналов	– 2.5
a_2		4.7
a_3		5.4

1. Смоделировать дискретный прямоугольный импульс на основе дискретного единичного скачка $\sigma_d(k)$ из л/р №1 с выводом графика на интервале времени $n \in [0, N-1]$. Пояснить, как выполняется моделирование импульса.

Формула для прямоугольного импульса:

$$s_1(k) = \begin{cases} U, & n_0 \le k \le n_0 + n_{imp} - 1 \\ 0, & \text{иначе} \end{cases}, \quad k \in \mathrm{Z}$$

$$s_1(k) = \begin{cases} 4, & 7 \le k \le 15 \\ 0, & \text{иначе} \end{cases}, \quad k \in \mathrm{Z}$$

Формула для дискретного единичного скачка:

$$\sigma_d(k) = \begin{cases} 1, & k \ge 0 \\ 0, & k < 0 \end{cases}, \quad k \in \mathrm{Z}$$

Формула для дискретного прямоугольного импульса, выраженная через дискретный единичный скачок:

$$s_1(k) = U \cdot \left(\sigma_d(k-7) - \sigma_d(k-16)\right)$$

$$s_1(k) = 4 \cdot \left(\sigma_d(k-7) - \sigma_d(k-16)\right)$$

Формула была составлена из следующих умозаключений: дискретный единичный скачок дает нам сигнал, равный 1, с момента времени 0. Соответственно, если из дискретного единичного скачка вычесть смещенный дискретный единичный скачок, то получится некоторый дискретный прямоугольный импульс. Выбрав правильные смещения для дискретных единичных скачков, можно получить дискретный прямоугольный импульс любой длительности.

График дискретного прямоугольного импульса на основе дискретного единичного скачка представлен на рис. 1.

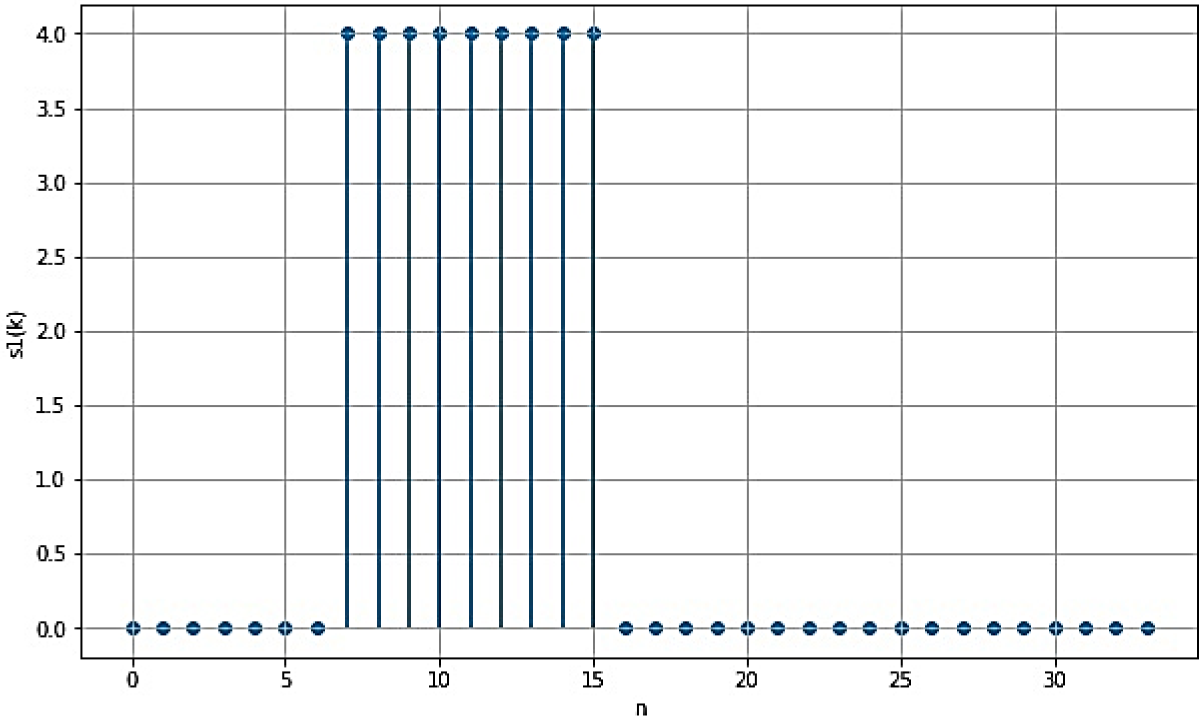

Рисунок 1 – График прямоугольного импульса на основе дискретного единичного скачка.

2. Смоделировать линейную комбинацию дискретных гармонических сигналов:

$$s_2(k) = a_1 x_1(k) + a_2 x_2(k) + a_3 x_3(k),$$

где

$$x_i(k) = B_i \sin(\hat{\omega}_i k),\ i = 1, 2, 3$$

с выводом графиков последовательностей $x_i(k)$ и $s_2(k)$ на интервале времени $n \in [0,\ 5N - 1]$. Вычислить среднее значение, энергию и среднюю мощность последовательности $s_2(k)$. Пояснить:

- операции при моделировании линейной комбинации сигналов $s_2(k)$;
- как определяют указанные характеристики.

Графики последовательностей $x_i(k)$ и $s_2(k)$ на интервале времени $n \in [0,\ 5N - 1]$ приведены на рисунках 2-6:

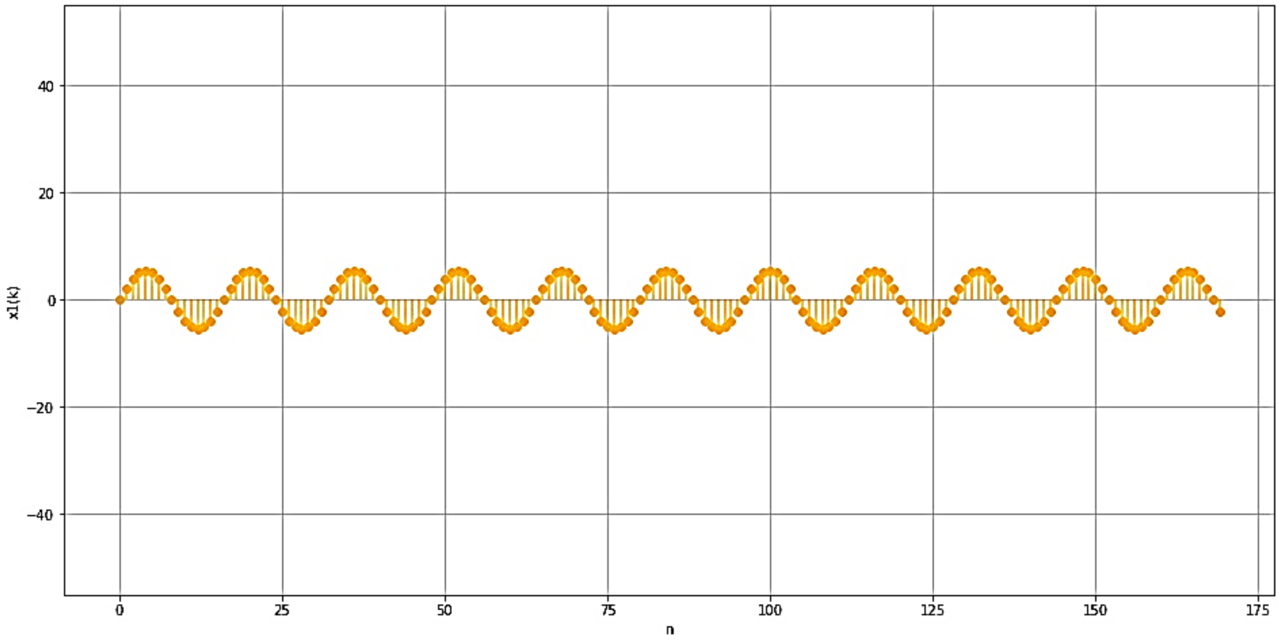

Рисунок 2 – График последовательности $x_1(k)$.

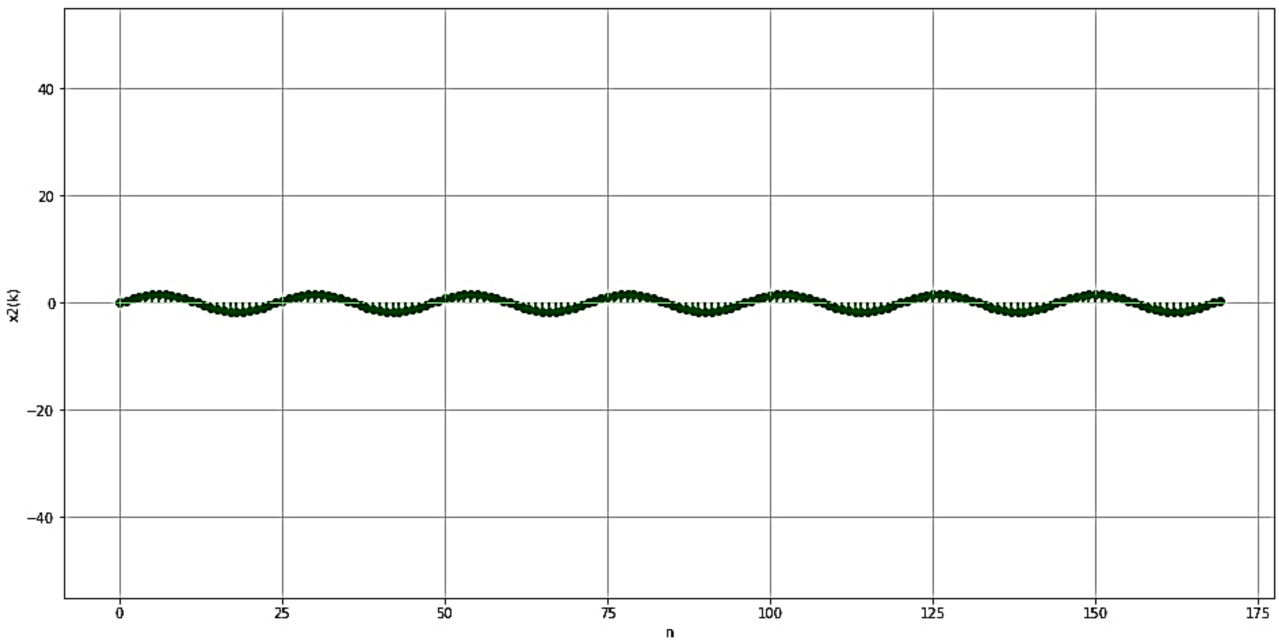

Рисунок 3 – График последовательности $x_2(k)$.

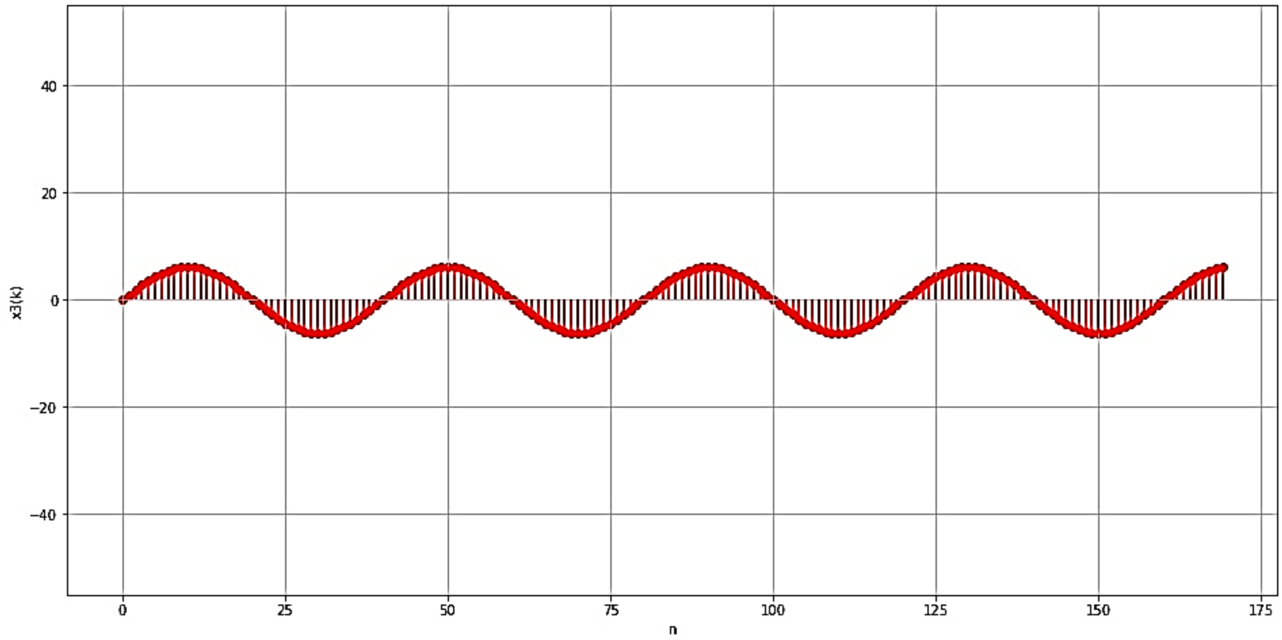

Рисунок 4 – График последовательности $x_3(k)$.

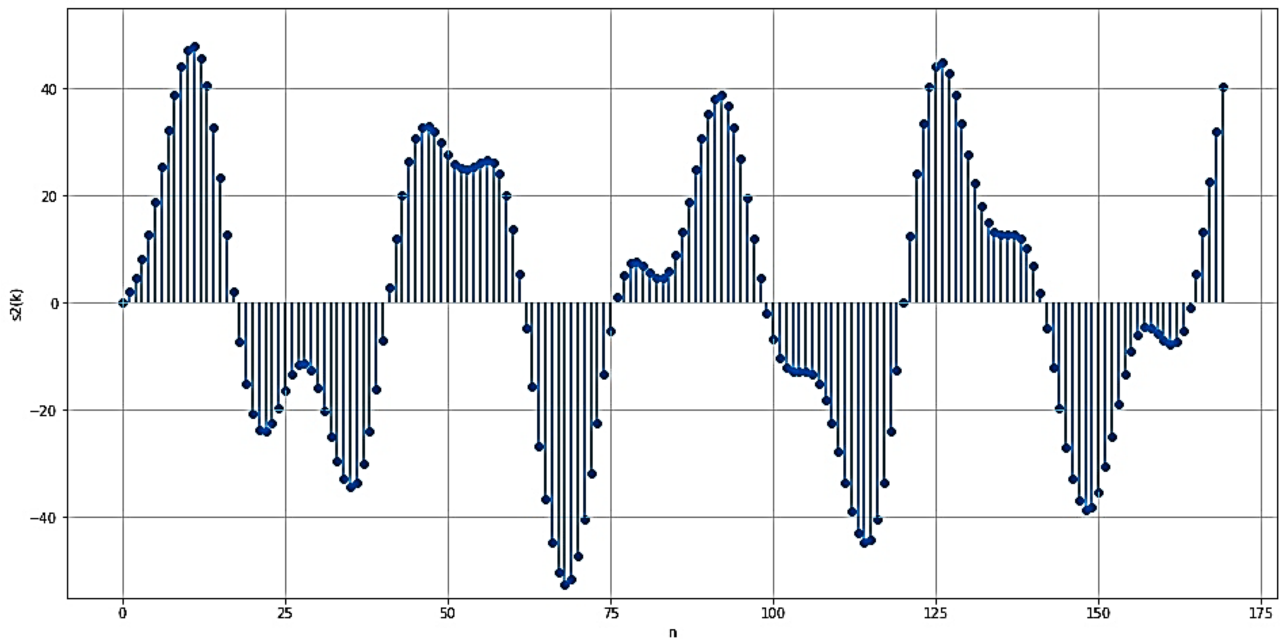

Рисунок 5 – График последовательности $s_2(k)$.

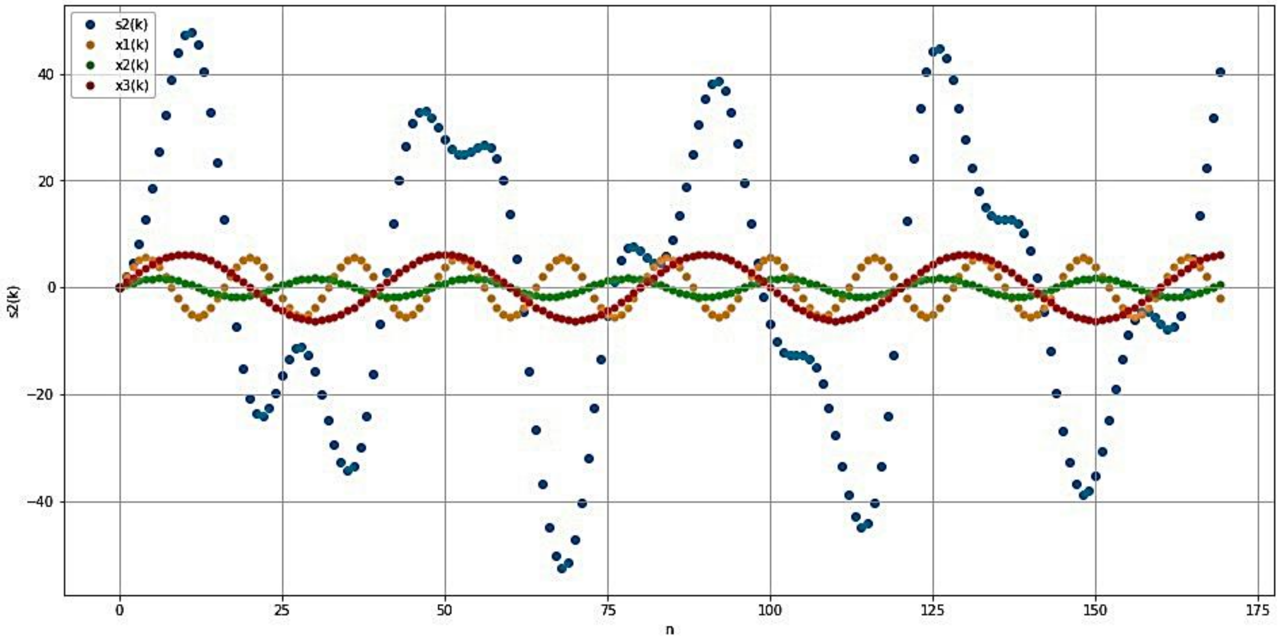

Рисунок 6 – Графики последовательностей $x_i(k)$ и $s_2(k)$.

Моделирование осуществлялось при помощи вычисления компонент сигналов $a_i x_i(k)$ для требуемых k, после чего полученные компоненты складывались. Были использованы операции сложения, умножения и вычисления синуса. Вычислены средние значение, энергия и мощность последовательности $s_2(k)$ с учетом, что период сигнала $s_2(k)$ равен 240:

$$\bar{U} = 0.0$$

$$E = 164857.56$$

$$P_{cp} = 686.9065$$

Такие характеристики, как среднее значение, энергия и мощность последовательности $s_2(k)$ вычисляются по следующим формулам:

$$\bar{U} = \frac{1}{T}\int_0^T s(t)dt$$

$$E = \int_0^T s^2(t)dt$$

$$P_{cp} = \frac{1}{T}\int_0^T s^2(t)dt$$

3. Вывести график дискретного сигнала $s_3(k)$ представляющего собой дискретный гармонический сигнал $s(k)$:

$$s(k) = C * cos(\widehat{\omega}_0 k)$$

с экспоненциальной огибающей $|a|^k$, на интервале времени $n \in [0, N-1]$. Привести аналитическую формулу дискретного сигнала $s_3(k)$ и пояснить операции при его моделировании.

Формула для дискретного сигнала $s_3(k)$:

$$s_3(k) = |a|^k \cdot C \cdot \cos(\hat{\omega}_0 k)$$

$$s_3(k) = |0.82|^k \cdot 5 \cdot \cos(\frac{\pi k}{10})$$

График дискретного сигнала $s_3(k)$ приведен на рис. 7:

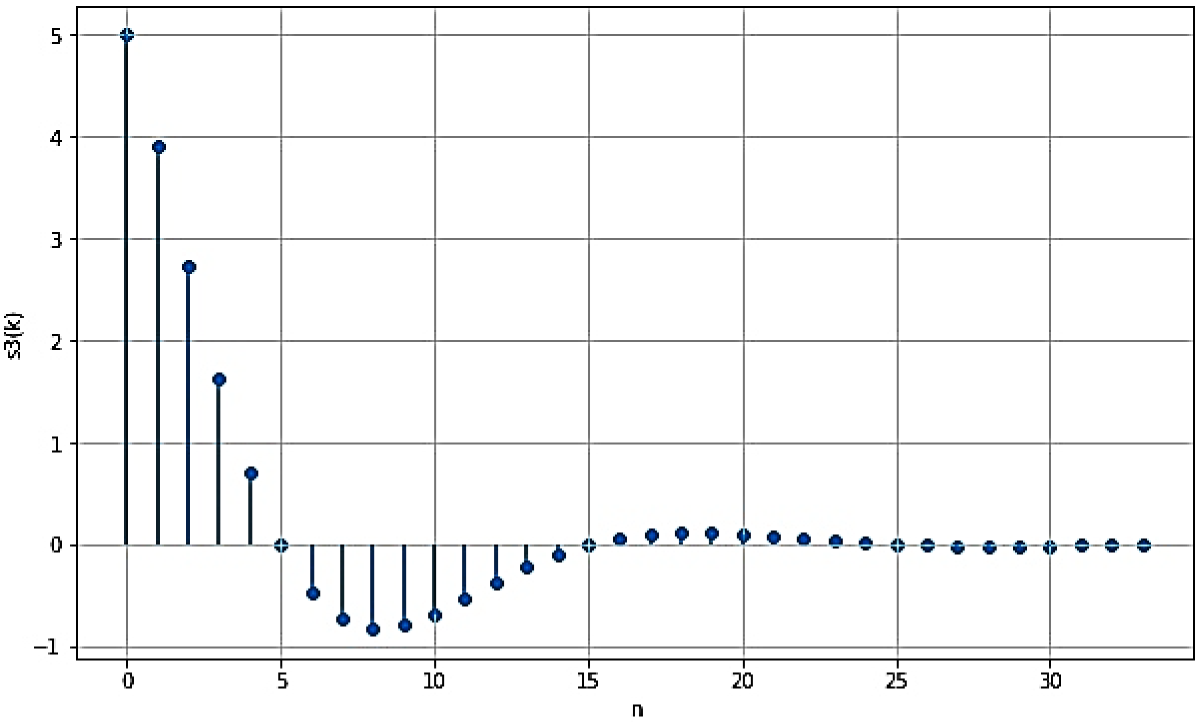

Рисунок 7 – Графики дискретного сигнала $s_3(k)$.

Для моделирования данного сигнала были посчитаны значения дискретного гармонического сигнала, умноженного на затухающую экспоненту.

Были использованы операции умножения, взятия модуля, возведения в степень и вычисление косинуса.

4. Вывести график пяти периодов периодической последовательности $s_4(k)$ дискретных прямоугольных импульсов амплитуды U и длительности n_{imp} с периодом, вдвое большим длительности импульса. Пояснить операции при моделировании периодической последовательности.

Формула для дискретного сигнала $s_4(k)$:

$$s_4(k) = s_1(k \text{ mod } (2 \cdot n_{imp}) + n_0)$$

График пяти периодов периодической последовательности $s_4(k)$ приведен на рис. 8:

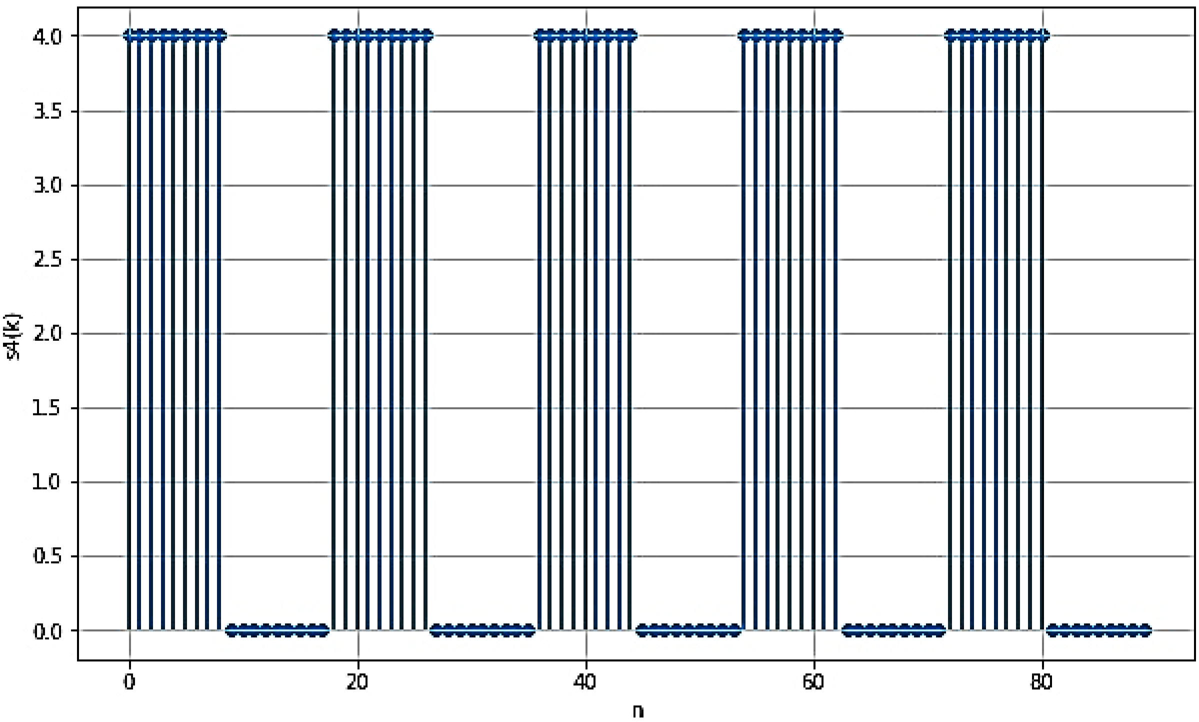

Рисунок 8 – График пяти периодов периодической последовательности $s_4(k)$.

Для моделирования сигнала использовалась уже готовый дискретный сигнал $s_1(k)$. Для того, чтобы сделать сигнал периодичным, использовалась операция взятия остатка от деления, где делитель определяет значение периода.

Для сдвига прямоугольного импульса к моменту времени 0 к аргументу было прибавлено значение n_0.

Выводы.

В процессе выполнения лабораторной работы было рассмотрено математическое описание линейных комбинаций дискретных гармонических сигналов и выполнено моделирование данных сигналов при помощи программных средств, а именно языка программирования Python и библиотек *numpy*, *matplotlib* и *cmath*.

Были рассмотрены характеристики и смоделированы следующие специальные виды дискретных сигналов: дискретный прямоугольный импульс; дискретные гармонические сигналы и их линейная комбинация, а также среднее значение, энергия и средняя мощность соответствующих сигналов; экспоненциальная огибающая дискретного гармонического сигнала; периоды периодической последовательности дискретных прямоугольных импульсов; Были построены и проанализированы графики соответствующих сигналов на интервалах дискретного времени.

ПРИЛОЖЕНИЕ А
ИСХОДНЫЙ КОД ПРОГРАММЫ

```
import math
import cmath
import numpy as np
import matplotlib.pyplot as plt

Nb = 4
N = 30 + Nb % 5
a = ((-1) ** Nb) * (0.8 + 0.005 * Nb)
C = 1 + Nb % 5
w0 = math.pi / (6 + Nb % 5)
U = Nb
n0 = 3 + Nb % 5
n_imp = 5 + Nb % 5
B1 = 1.5 + Nb % 5
B2 = 5.7 - Nb % 5
B3 = 2.2 + Nb % 5
w1 = math.pi / (4 + Nb % 5)
w2 = math.pi / (8 + Nb % 5)
w3 = math.pi / (16 + Nb % 5)
a1 = 1.5 - Nb % 5
a2 = 0.7 + Nb % 5
a3 = 1.4 + Nb % 5

print(Nb, N, a, C, w0, U, n0, n_imp, B1, B2, B3, w1, w2, w3,
a1, a2, a3)

def sigma_d(k):
    return 1  if k >= 0 else 0

def s1(k):
    return U * (sigma_d(k - n0) - sigma_d(k - (n0 + n_imp)))

n_values = np.linspace(0, (N - 1), N)

plt.figure(figsize=(10, 6))
plt.scatter(n_values, [s1(n) for n in n_values])
plt.vlines(n_values, 0, [s1(n) for n in n_values])
plt.ylabel("s1(k)")
plt.xlabel("n")
plt.grid()
plt.show()
```

```
def x1(k):
    return B1 * math.sin(w1 * k)

def x2(k):
    return B2 * math.sin(w2 * k)

def x3(k):
    return B3 * math.sin(w3 * k)

def s2(k):
    return a1 * x1(k) + a2 * x2(k) + a3 * x3(k)

n_values = np.linspace(0, 5 * N - 1, 5 * N)

plt.figure(figsize=(16, 8))
plt.scatter(n_values, [s2(n) for n in n_values], label="s2(k)")
plt.scatter(n_values, [x1(n) for n in n_values], label="x1(k)",
linewidths=0.1)
plt.scatter(n_values, [x2(n) for n in n_values], label="x2(k)",
linewidths=0.1)
plt.scatter(n_values, [x3(n) for n in n_values], label="x3(k)",
linewidths=0.1)
plt.xlabel("n")
plt.ylabel("s2(k)")
plt.grid()
plt.legend()
plt.show()

plt.figure(figsize=(16, 8))
plt.scatter(n_values, [s2(n) for n in n_values])
plt.vlines(n_values, 0, [s2(n) for n in n_values])
plt.ylabel("s2(k)")
plt.xlabel("n")
plt.ylim(-55, 55)
plt.grid()
plt.show()

plt.figure(figsize=(16, 8))
plt.scatter(n_values, [x1(n) for n in n_values], c='orange')
plt.vlines(n_values,   0,   [x1(n)   for   n   in   n_values],
colors='orange')
plt.ylabel("x1(k)")
plt.xlabel("n")
plt.ylim(-55, 55)
plt.grid()
```

```
    plt.show()

    plt.figure(figsize=(16, 8))
    plt.scatter(n_values, [x2(n) for n in n_values], c='green')
    plt.vlines(n_values,   0,   [x2(n)   for   n   in   n_values],
colors='green')
    plt.ylabel("x2(k)")
    plt.xlabel("n")
    plt.ylim(-55, 55)
    plt.grid()
    plt.show()

    plt.figure(figsize=(16, 8))
    plt.scatter(n_values, [x3(n) for n in n_values], c='red')
    plt.vlines(n_values,   0,   [x3(n)   for   n   in   n_values],
colors='red')
    plt.ylabel("x3(k)")
    plt.xlabel("n")
    plt.ylim(-55, 55)
    plt.grid()
    plt.show()

    T = math.lcm(int(2 * math.pi / w1), int(2 * math.pi / w2), int(2
* math.pi / w3))
    print(T)

    n_values = np.linspace(0, T - 1, T)
    mean_s2 = sum([s2(n) for n in n_values]) / len(n_values)
    print(round(mean_s2, 5))

    E = sum([s2(n) ** 2 for n in n_values])
    print(E)

    P = sum([s2(n) ** 2 for n in n_values]) / T
    print(round(P, 5))

    n_values = np.linspace(0, N - 1, N)

    def s3(k):
        return (abs(a) ** k) * C * math.cos(w0 * k)

    plt.figure(figsize=(10, 6))
    plt.scatter(n_values, [s3(n) for n in n_values])
    plt.ylabel("s3(k)")
    plt.xlabel("n")
    plt.vlines(n_values, 0, [s3(n) for n in n_values])
```

```
plt.grid()
plt.show()

def s4(k):
    return s1(k % (2 * n_imp) + n0)

T = 2 * n_imp
n_values = np.linspace(0, 5 * T - 1, 5 * T)

plt.figure(figsize=(10, 6))
plt.scatter(n_values, [s4(n) for n in n_values])
plt.ylabel("s4(k)")
plt.xlabel("n")
plt.vlines(n_values, 0, [s4(n) for n in n_values])
plt.grid()
plt.show()
```