МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

высшего образования

«КРЫМСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ им. В. И. ВЕРНАДСКОГО»

ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ

Кафедра компьютерной инженерии и моделирования

Представление сигналов в ортонормированном базисе

Отчет по лабораторной работе 1

по дисциплине «**Обработка сигналов**»

студента 3 курса группы ИВТ-222(2)

Чудопалова Богдана Андреевича

Направления подготовки 09.03.01«Информатика и вычислительная техника»

Симферополь, 2025

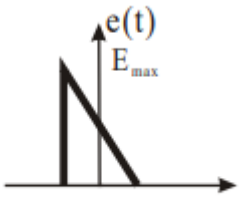
Лабораторная работа №1

Тема: Представление сигналов в ортонормированном базисе

Цель работы: разработать программное обеспечение, осуществляющее апроксимацию импульс системой ортонормированных функций Уолша, определяющее норму импульса и энергию импульса через систему ортонормированных функций Уолша. Провести цикл вычислительных экспериментов, в котором определить количество коэффициентов при разложении по функциям Уолша исходя из потери относительной энергии сигнала (10 %, 5 %, 2 %, 1 %, 0,1 %).

Ход работы

Вариант2: E_max = 90; t_и = 4



б)

Теоретические сведения, используемые в ходе выполнения работы

Произвольный сигнал s(t) может быть представлен в виде обобщенного ряда Фурье в ортонормированном базисе:

$$s(t)=\sum_{i=0}^{\infty} c_i \cdot u_i(t) \qquad (1.1),$$

Норма и энергия сигнала определяются следующими выражениями:

$$\|s\|=\sqrt{\int_{-\infty}^{\infty} s^2(t)\cdot dt}\ ;\ E_s=\|s\|^2=\int_{-\infty}^{\infty} s^2(t)\cdot dt$$

Разложение произвольного сигнала с конечной энергией, заданного на временном интервале [-T/2, T/2], в обобщенный ряд Фурье по функциям Уолша имеет вид:

$$s(t)=\sum_{k=0}^{\infty} c_k \cdot wal(k, t/T) \qquad (1.7).$$

Коэффициенты в разложении по функциям Уолша определяются следующим выражением:

$$c_k=\int_{-l/2}^{l/2} s(\vartheta)\cdot wal(k,\vartheta)\cdot d\vartheta \qquad (1.8).$$

Энергия произвольного сигнала, аппроксимированного ортонормированной системой функций Уолша, определяется следующим выражением:

$$E_w=\sum_{k=0}^{\infty} c_k^2 \qquad (1.9).$$

Количество коэффициентов в разложении по функциям Уолша определяется из следующего выражения:

$$\frac{|E_s-E_w|}{E_s}\cdot 100\% \le \delta \qquad (1.10),$$

Вывод моей программы имеет вид:

```
Норма исходного сигнала: 104.08
Энергия исходного сигнала: 1.08e+04 Дж
```

Рис. 1. Вывод численных данных.

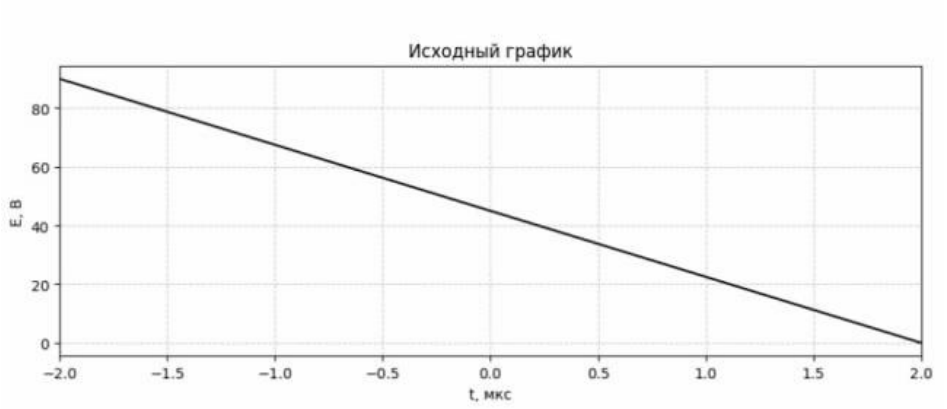


Рис. 2. Отображение графика исходной функции.

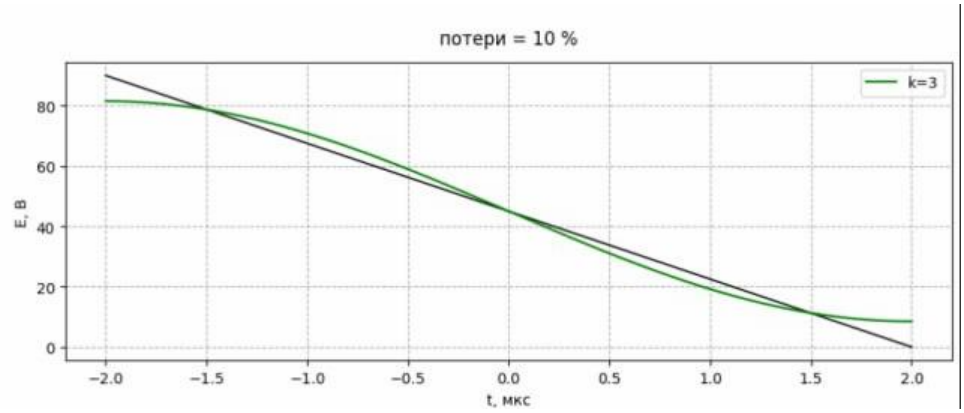


Рис. 3. Отображение аппроксимированной функции про потерях 10%

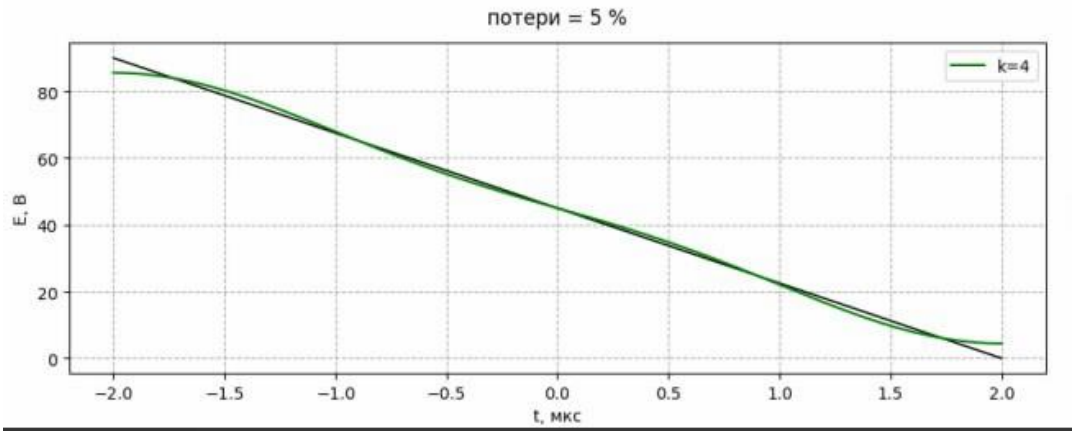


Рис. 4. Отображение аппроксимированной функции про потерях 5%

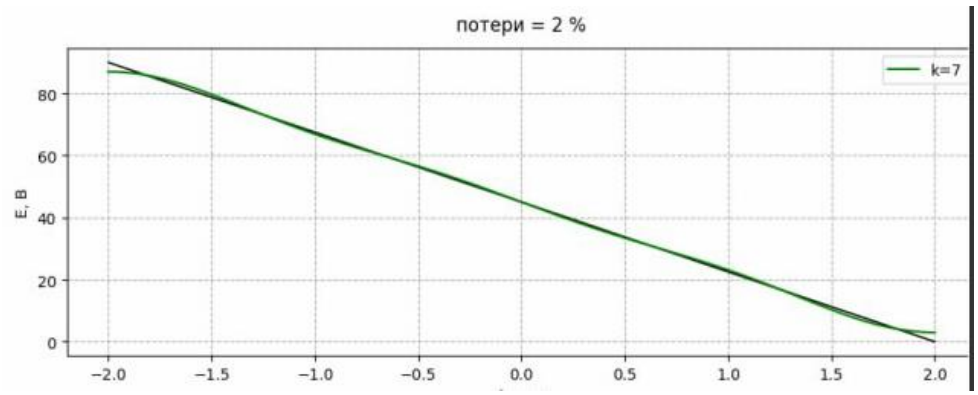


Рис. 5. Отображение аппроксимированной функции про потерях 2%

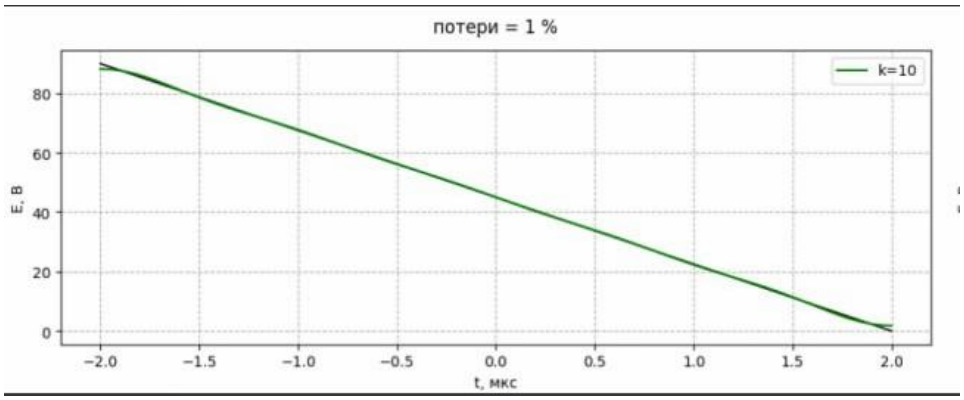


Рис. 6. Отображение аппроксимированной функции про потерях 1%

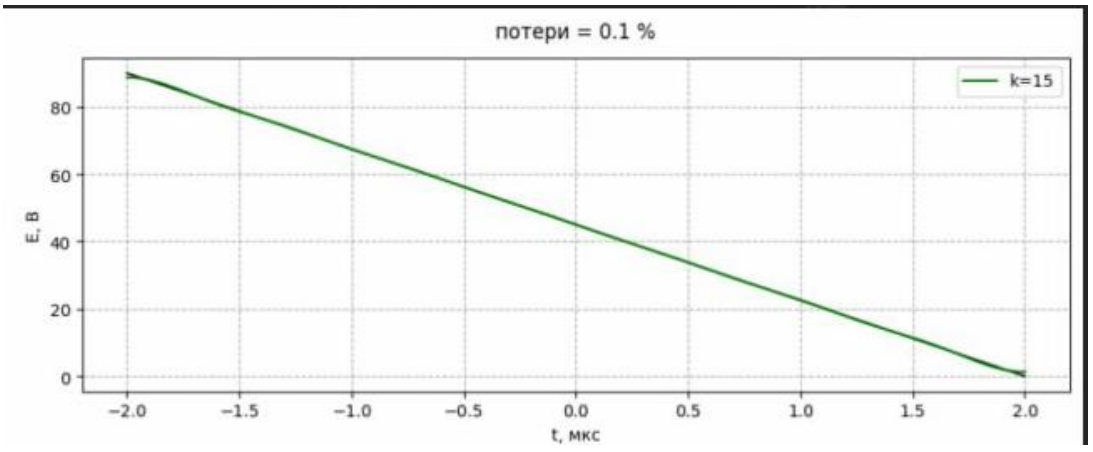


Рис. 7. Отображение аппроксимированной функции про потерях 0.1%

Вывод: В ходе выполнения лабораторной работы были достигнуты следующие результаты:

1. Разработано программное обеспечение для аппроксимации импульса системой ортонормированных функций Уолша.
2. Определена норма импульса и энергия сигнала.
3. Проведен цикл вычислительных экспериментов, в ходе которого было определено количество коэффициентов разложения по функциям Уолша для различных потерь относительной энергии сигнала (10 %, 5 %, 2 %, 1 %, 0,1 %).
4. Графически представлены исходный и аппроксимированный импульсы для разного количества коэффициентов.

Приложение:

Реализация основных функций и формул программы:

```
# Определение обобщенного ряда Фурье
def fourier_series(coefficients, walsh_functions, t):
    series_sum = np.zeros_like(t)
    for i, coef in enumerate(coefficients):
        series_sum += coef * walsh_functions[i](t)
    return series_sum
```

Рис. 8. Функция 1.

```
# Норма и энергия сигнала
norm_signal = np.sqrt(np.trapz(signal**2, t))
energy_signal = np.trapz(signal**2, t)
```

Рис. 9. Формулы для вычисления нормы и энергии сигнала.

```
# Функция для вычисления функций Уолша
def walsh(n, x):
    if n == 0:
        return np.ones_like(x)
    else:
        p = n % 2
        n_half = n // 2
        return np.where((x >= -0.5) & (x < 0),
                        (-1)**p * walsh(n_half, 2*x + 1),
                        (-1)**(p + 1) * walsh(n_half, 2*x - 1))
```

Рис. 10. Функция 2.

```
# Вычисление коэффициентов ряда Фурье
coefficients = []
for k in range(max_coefficients):
    wal_func = walsh_function(k, t/T)
    c_k = np.trapz(signal * wal_func, t)  # Используем метод трапеций для интегрирования
    coefficients.append(c_k)
```

Рис. 11. Функция 3.