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

Лабораторная работа № 2

«Спектральный анализ периодических сигналов»
По дисциплине
«Обработка сигналов»

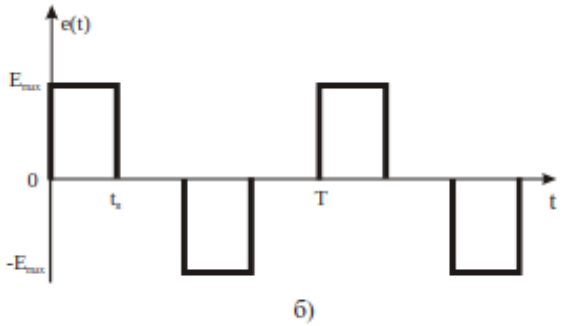
Выполнил:
Студент 3 курса
Направления подготовки
09.03.01 «Информатика и вычислительная техника»
Группы ИВТ-б-о-222
Чудопалов Богдан Андреевич
Проверил:
Таран Е.П.
«_»_____20__г.
Оценка:_____
_____
Подпись:_____
_____

Симферополь, 2025

Лабораторная работа № 2 «Спектральный анализ периодических сигналов»

Цель работы: разработать программное обеспечение, которое будет осуществлять спектральный анализ периодической последовательности импульсов.

Техническое задание: задан произвольный сигнал. Необходимо разработать программное обеспечение, которое будет вычислять коэффициенты разложения сигнала в ряд Фурье, амплитуду и фазу гармоник, строить амплитудные и фазовые спектральные диаграммы, а также строить график аппроксимированного сигнала исходя из потери относительной мощности сигнала. Номер варианта: 2



б)

Рис. 1. Форма сигнала согласно варианту

Таблица 1. Параметры сигнала

№ варианта	2
Сигнал	б
Emax, В	10
$t_и$, мкс	112
T, мкс	448

Ход работы

1. **Теоретический раздел:**

Периодическая последовательность импульсов может быть записана в виде ряда Фурье для периодического сигнала:

$$s(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} (a_n \cdot \cos(n \cdot \omega_1 \cdot t) + b_n \cdot \sin(n \cdot \omega_1 \cdot t)$$

Коэффициенты разложения рассчитываются по следующим формулам:

$$a_0 = \frac{2}{T} \cdot \int_{-T/2}^{T/2} s(t) \cdot dt \,;\; a_n = \frac{2}{T} \cdot \int_{-T/2}^{T/2} s(t) \cdot \cos(n \cdot \omega_1 \cdot t) dt \,;\; b_n = \frac{2}{T} \cdot \int_{-T/2}^{T/2} s(t) \cdot \sin(n \cdot \omega_1 \cdot t) dt$$

Амплитуда и начальная фаза гармоник определяются через коэффициенты ряда Фурье:

$$A_0 = \sqrt{a_n^2 + b_n^2} \quad \operatorname{tg}\varphi_n = \frac{b_n}{a_n}$$

Средняя мощность для периодической последовательности импульсов определяется следующим выражением:

$$P_c = \frac{1}{T} \cdot \int_{-T/2}^{T/2} s^2(t) \cdot dt$$

Мощность, заключенная в сложном периодическом сигнале, может быть рассчитана через коэффициенты ряда Фурье:

$$P_k = \left(a_0 / 2 \right)^2 + \frac{1}{2} \cdot \sum_{n=1}^{\infty} (A_n)^2$$

При учете конечного числа гармоник теряется часть мощности. Выбор максимальной гармоники (определение практической ширины спектра) зависит от отношения:

$$\frac{P_k}{P_c} \geq \delta \,.$$

2. Практический раздел

При запуске программы будут автоматически проведены все необходимые расчёты и выведена информация в доступном для пользователя виде.

Данные которые будут выведены:

1. Исходный сигнал
2. Амплитудный и фазовый спектр
3. Данные (Мощность сигнала, амплитуды и фазы гармоник)
4. 5 экспериментов (10%, 5%, 2%, 1%, 0.1%)

Первым шагом выводится исходный сигнал:

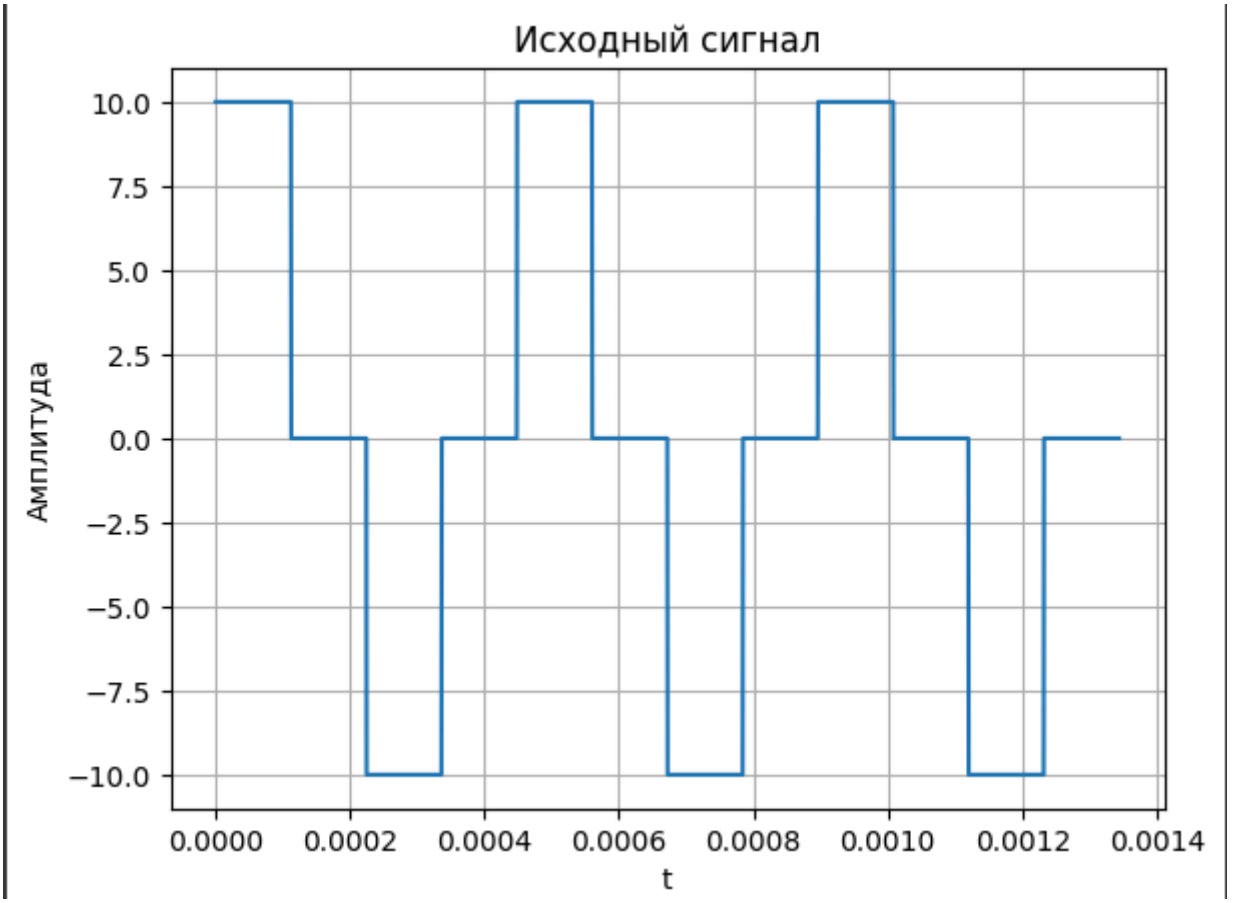


Рис. 2. Форма исходного сигнала.

Далее отображаются амплитудный и фазовый спектр сигнала:

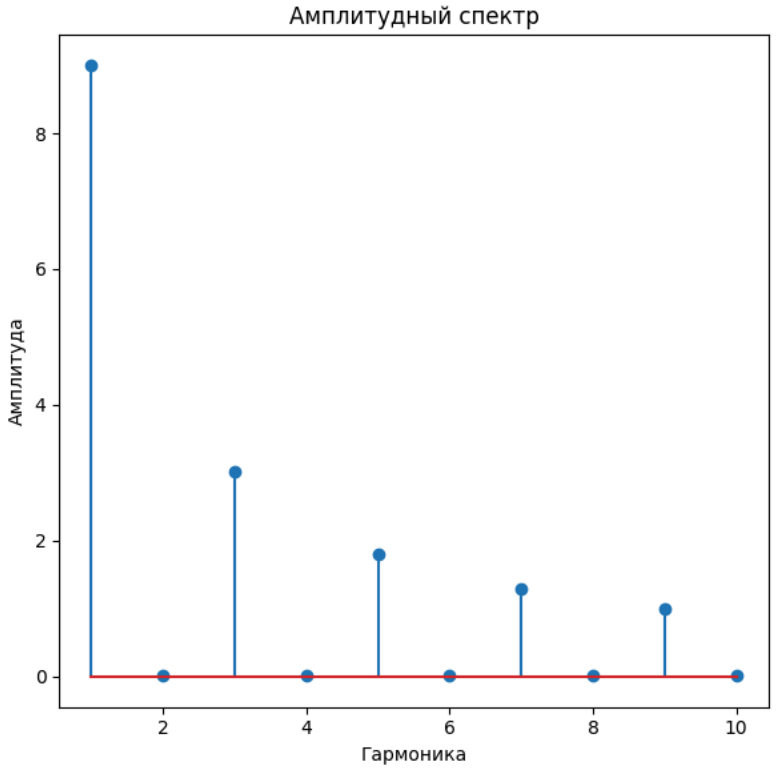


Рис. 3. Амплитудный спектр сигнала.

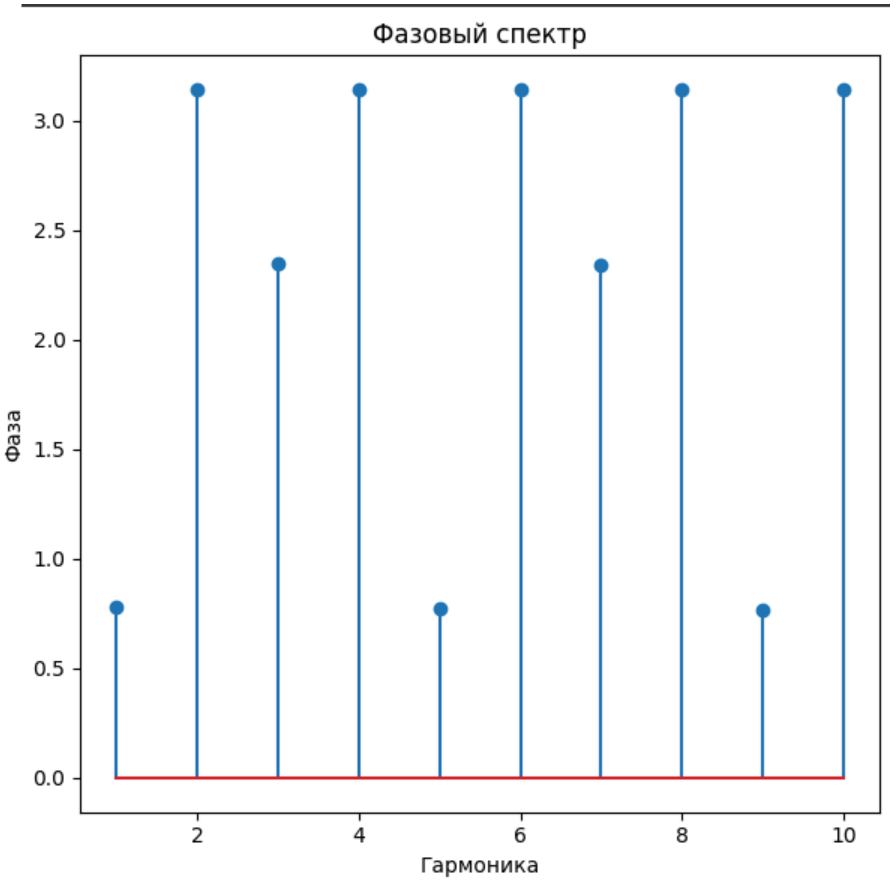


Рис. 4. Фазовый спектр сигнала.

Далее выводятся данные, рассчитанные в процессе работы программы

```
Средняя мощность сигнала: 49.925000000000416
Количество гармоник:
Для 10% потерь: 3
Для 5% потерь: 7
Для 2% потерь: 19
Для 1% потерь: 39
Для 0.1% потерь: 253

Амплитуды гармоник:
n=1: 8.996
n=2: 0.010
n=3: 3.008
n=4: 0.010
n=5: 1.794
n=6: 0.010
n=7: 1.293
n=8: 0.010
n=9: 0.993
n=10: 0.010

Фазы гармоник:
n=1: 0.783
n=2: 3.142
n=3: 2.349
n=4: 3.142
n=5: 0.774
n=6: 3.142
n=7: 2.340
n=8: 3.142
n=9: 0.764
n=10: 3.142
```

Рис. 5. Вывод данных

Далее выводятся аппроксимированные графики нашего сигнала с разным количество гармоник в зависимости от необходимых потерь

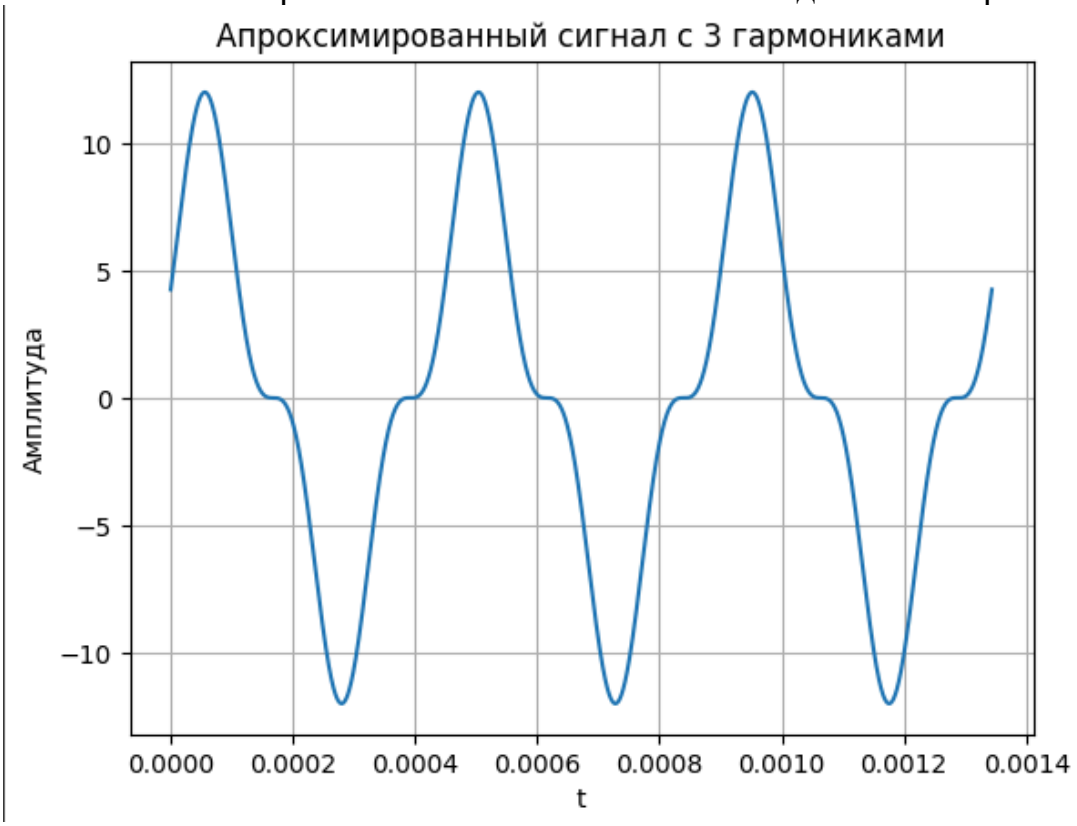


Рис. 6. Для 10% потерь.

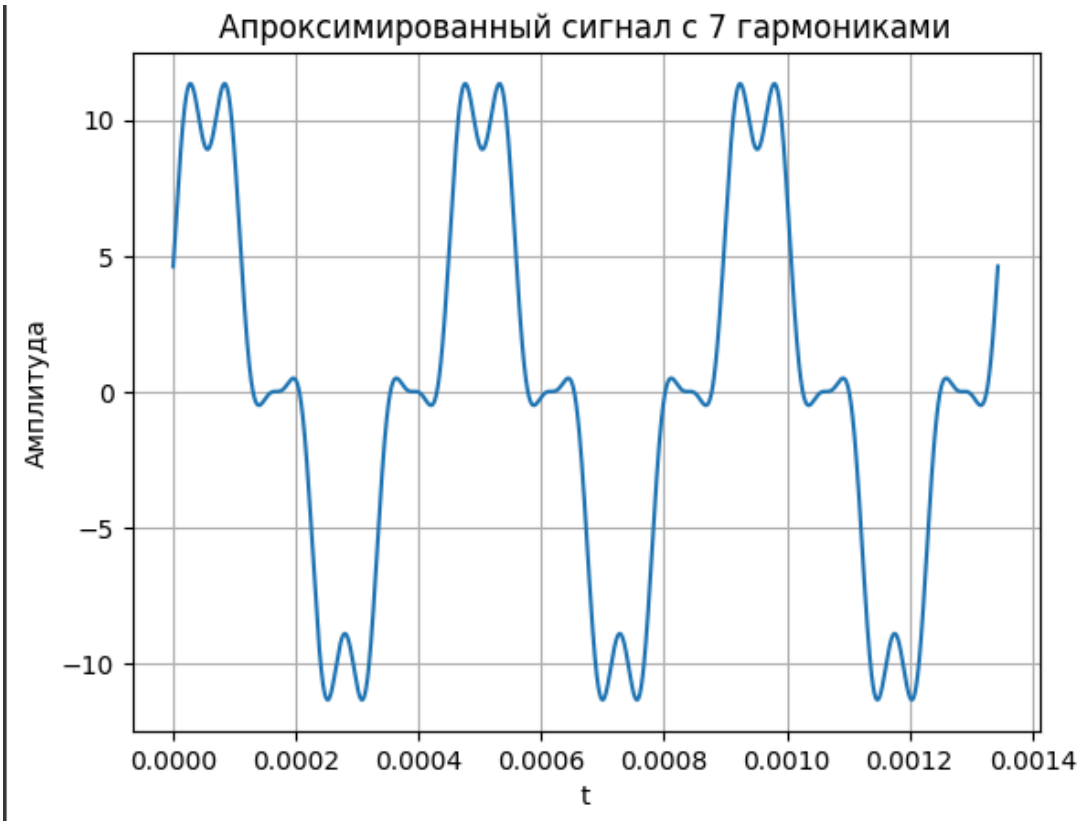


Рис. 7. Для 5% потерь.

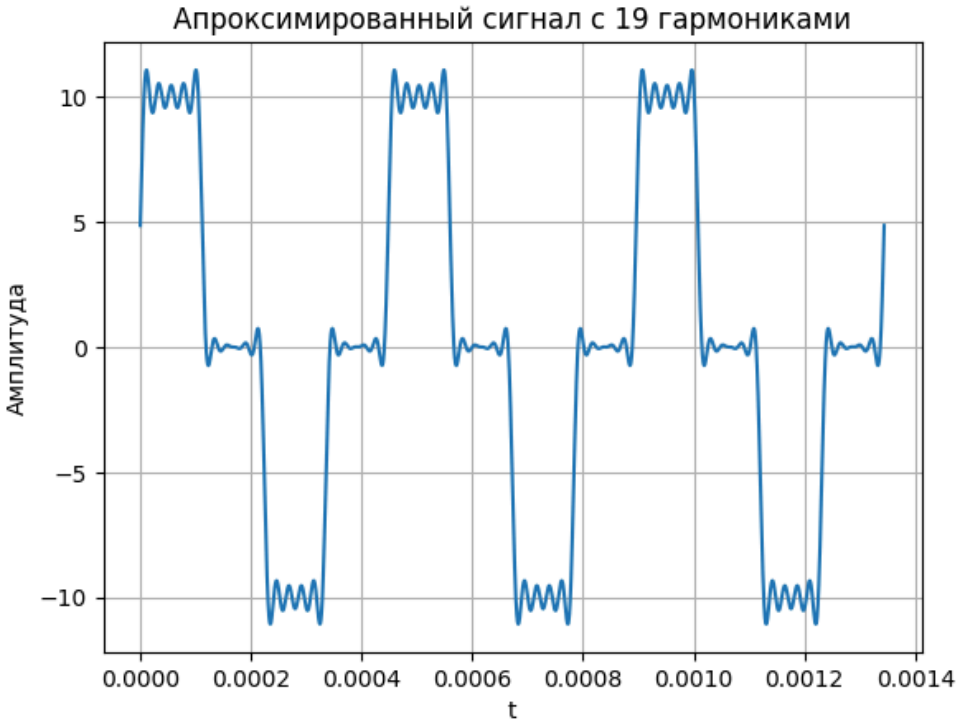


Рис. 8. Для 2% потерь.

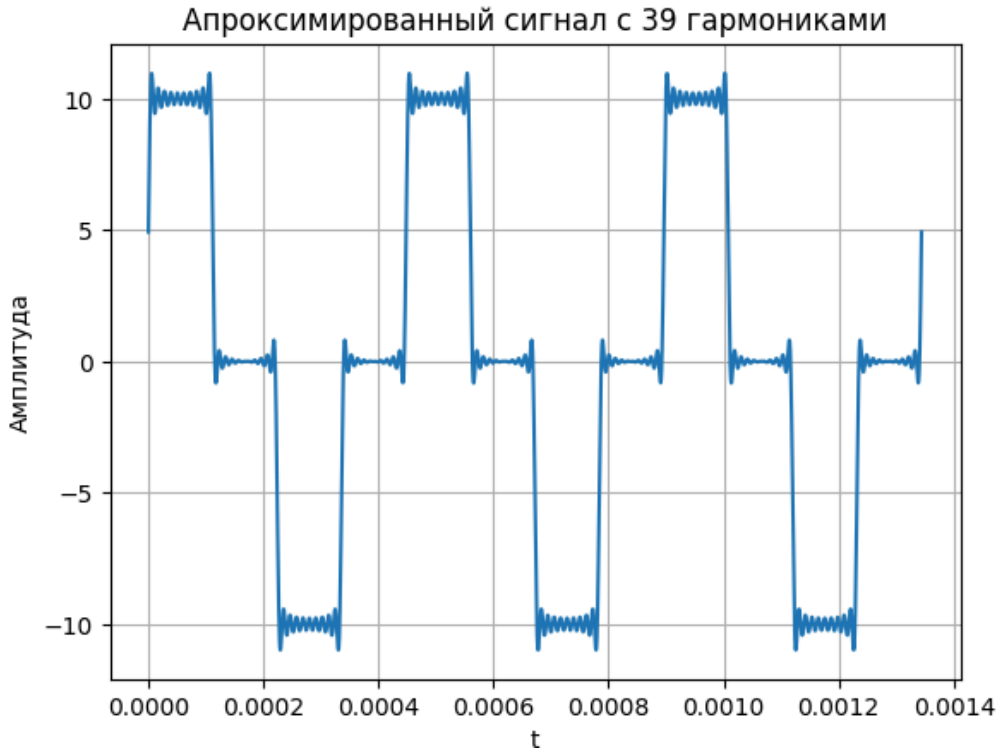


Рис. 9. Для 1% потерь.

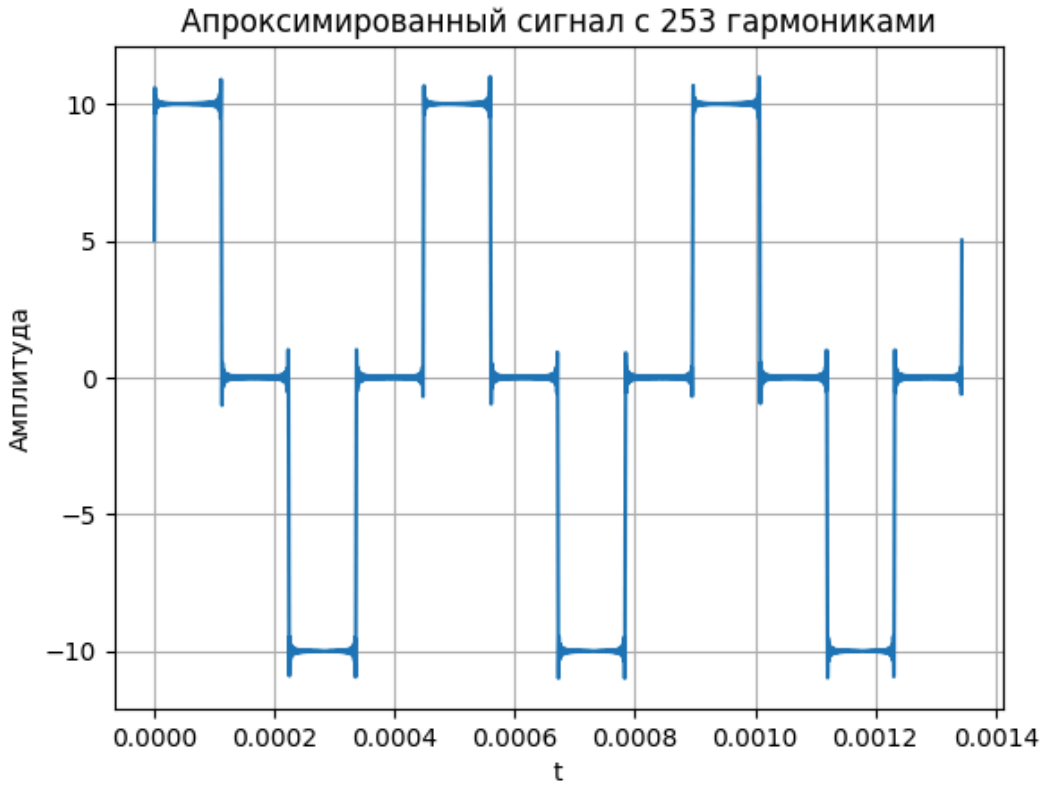


Рис. 10. Для 0.1% потер.

Приложение

```
def modified_rectangular_signal(x):
    while x > T:
        x -= T
    while x < 0:
        x += T

    quarter = T / 4


    if 0 <= x < quarter:
        return Emax
    elif quarter <= x < 2 * quarter:
        return 0
    elif 2 * quarter <= x < 3 * quarter:
        return -Emax
    else:
        return 0
```

1.1. Функция для описания сигнала.

```
def func_a_0():
  a = 0
  b = T
  h = (b - a)/1000
  result = 0
  for i in np.arange(a, b + h, h):
    if i == a or i == b:
      result += modified_rectangular_signal(i)
    else:
      result += 2 * modified_rectangular_signal(i)
  return (2 / T) * result * (h / 2)

def func_a_n(n):
  a = 0
  b = T
  h = (b - a) / 1000
  result = 0
  for i in np.arange(a, b + h, h):
    if i == a or i == b:
      result += modified_rectangular_signal(i) * np.cos(n * w * i)
    else:
      result += 2 * modified_rectangular_signal(i) * np.cos(n * w * i)
  return (2 / T) * result * (h / 2)

def func_b_n(n):
  a = 0
  b = T
  h = (b - a) / 1000
  result = 0
  for i in np.arange(a, b + h, h):
    if i == a or i == b:
      result += modified_rectangular_signal(i) * np.sin(n * w * i)
    else:
      result += 2 * modified_rectangular_signal(i) * np.sin(n * w * i)
  return (2 / T) * result * (h / 2)
```

1.2. Набор функция для расчета коэффициентов Фурье.

```
plt.plot(x_values, y_values)
plt.title("Исходный сигнал")
plt.xlabel("t")
plt.ylabel("Амплитуда")
plt.grid(True)
plt.show()

amplitudes = [np.sqrt(a_n[i] ** 2 + b_n[i] ** 2) for i in range(len(a_n))]
phases = [np.arctan2(b_n[i], a_n[i]) for i in range(len(a_n))]

plt.figure(figsize=(12, 6))
plt.subplot(1, 2, 1)
plt.stem(range(1, 11), amplitudes[:10])
plt.title("Амплитудный спектр")
plt.xlabel("Гармоника")
plt.ylabel("Амплитуда")

plt.subplot(1, 2, 2)
plt.stem(range(1, 11), phases[:10])
plt.title("Фазовый спектр")
plt.xlabel("Гармоника")
plt.ylabel("Фаза")

plt.tight_layout()
plt.show()
```

1.3. Блок кода для отображения функции, амплитудного и фазового спектра.

```
# Расчет средней мощности сигнала
P_c = 0
for i in range(len(amplitudes)):
  P_c += (amplitudes[i] ** 2) / 2

print(f"Средняя мощность сигнала: {P_c}")

# Расчет накопленной мощности
P_k = [0]

for n_max in range(1, 501):
  P_k.append(0)
  for n in range(1, n_max+1):
    P_k[n_max] += a_n[n-1] * a_n[n-1] + b_n[n-1] * b_n[n-1]
  P_k[n_max] *= 0.5
  P_k[n_max] += (func_a_0() ** 2) / 2

# Расчет относительной ошибки
P_final = []
for i in range(1, 501):
  P_final.append(abs(P_k[i] - P_c) / P_c * 100)
```

1.4. Расчет необходимых величин.

```
print("Количество гармоник:")
print(f"Для 10% потерь: {n_chart[0]}")
print(f"Для 5% потерь: {n_chart[1]}")
print(f"Для 2% потерь: {n_chart[2]}")
print(f"Для 1% потерь: {n_chart[3]}")
print(f"Для 0.1% потерь: {n_chart[4]}")

print("\nАмплитуды гармоник:")
for i in range(10):
    print(f"n={i+1}: {amplitudes[i]:.3f}")

print("\nФазы гармоник:")
for i in range(10):
    print(f"n={i+1}: {phases[i]:.3f}")
```

1.5. Вывод величин.

```
for idx in range(5):
  s_values = []
  for i in x_values:
    s = 0
    for n in range(1, n_chart[idx] + 1):
      s += a_n[n-1] * np.cos(n * w * i) + b_n[n-1] * np.sin(n * w * i)
    s += func_a_0() / 2
    s_values.append(s)

  plt.figure()
  plt.plot(x_values, s_values)
  plt.title(f"Апроксимированный сигнал с {n_chart[idx]} гармониками")
  plt.xlabel("t")
  plt.ylabel("Амплитуда")
  plt.grid(True)
  plt.show()
```

1.6. Отображение аппроксимированного графика.

3. Вывод: В ходе выполнения лабораторной работы были рассчитаны амплитуда и начальная фаза n-ой гармоники, средняя мощность периодического сигнала, а также мощность, рассчитанная через коэффициенты ряда Фурье. После этого были построены графики для различного количества коэффициентов, соответствующих разным потерям мощности сигнала.