1) Цель работы

- По заданным параметрам вычислить недостающие значения параметров
- Привести аналитические выражения для всех сигналов из сигнального множества как функций времени
- Вычислить значения энергий всех сигналов
- Построить графики всех сигналов

Вариант **III.9**

Квадратурная амплитудная модуляция:

f_0 = 1800 Гц — несущая частота

V_m = 2400 Бод — модуляционная скорость

V_i = 12000 бит/с — информационная скорость

2) Формулы для вычисления недостающих параметров

- Период следования сигналов

$$T = \frac{1}{V_m}$$

- Количество сигналов

$$q = 2^{V_i T}$$

- Коэффициенты

$$s_{i1} = A\left(1 - \frac{2 i_1}{\sqrt{q} - 1}\right);\ s_{i2} = A\left(1 - \frac{2 i_2}{\sqrt{q} - 1}\right)$$

где A – максимальное по абсолютной величине значение коэффициентов s_{i1} и s_{i2}.

3) Приведение аналитического выражение для всех сигналов из множества как функций времени

$$s_i(t) = \begin{cases} s_{i1}\sqrt{\frac{2}{T}}\cos(2\pi f_0 t) + s_{i2}\sqrt{\frac{2}{T}}\sin(2\pi f_0 t), \text{ если } 0 < t < T \\ 0, \text{ в противном случае} \end{cases}$$

или

$$s_i(t) = \begin{cases} \sqrt{\frac{2E_0}{T}}\cos(2\pi f_0 t - \theta_i), \text{ если } 0 < t < T \\ 0, \text{ в противном случае} \end{cases}, \text{ где } \begin{matrix} E_i = s_{i1}^2 + s_{i2}^2 \\ \theta = \arctan\left(\frac{s_{i2}}{s_{i1}}\right) \end{matrix}$$

4) Вычисление значения энергии сигналов

$$E_i = s_{i1}^2 + s_{i2}^2$$

или

$$E_i = \int_0^T \left(s_{i1}\sqrt{\frac{2}{T}}\cos(2\pi f_0 t) + s_{i2}\sqrt{\frac{2}{T}}\sin(2\pi f_0 t)\right) dt$$

$$\int_0^T \left(S_{i1}\sqrt{\frac{2}{T}}\cos(2\pi f_i T) + S_{i2}\sqrt{\frac{2}{T}}\sin(2\pi f_i T) \right) = E$$

$$\int_0^T \left(S^2{}_{i1}\frac{2}{T}\cos^2(2\pi f_i T) + 2S_{i1}S_{i2}\frac{2}{T}\sin(2\pi f_i T)\cos(2\pi f_i T) + S^2{}_{i2}\frac{2}{T}\sin^2(2\pi f_i T) \right) =$$

$$= S^2{}_{i1}\frac{2}{T}\left(\frac{4\pi\ f_i T + \sin(4\pi f_i T)}{8\pi f_i}\right) + 2S_{i1}S_{i2}\frac{2}{T}\left(\frac{1-\cos(4\pi f_i T)}{4\pi f_i}\right) + S^2{}_{i2}\frac{2}{T}\left(\frac{4\pi\ f_i T - \sin(4\pi f_i T)}{8\pi f_i}\right) =$$

$$= S^2{}_{i1}\frac{2}{T}\left(\frac{4\pi\ \frac{l}{T}T + \sin(4\pi\frac{l}{T}T)}{8\pi\frac{l}{T}}\right) + 2S_{i1}S_{i2}\frac{2}{T}\left(\frac{1-\cos(4\pi\frac{l}{T}T)}{4\pi\frac{l}{T}}\right) +$$

$$+ S^2{}_{i2}\frac{2}{T}\left(\frac{4\pi\ \frac{l}{T}T - \sin(4\pi\frac{l}{T}T)}{8\pi\frac{l}{T}}\right) = S^2{}_{i1}\frac{2}{T}\left(\frac{4\pi\ +0}{8\pi}\right)T + 2S_{i1}S_{i2}\frac{2}{T}*0 + S^2{}_{i2}\frac{2}{T}\left(\frac{4\pi\ -0}{8\pi}\right)T$$

$(\ т.к\ f_i = \frac{l}{T},\ где\ l - любое\ целое\ число)$

$получаем\ что\ E = S^2{}_{i1} + S^2{}_{i2}$

5) Графики

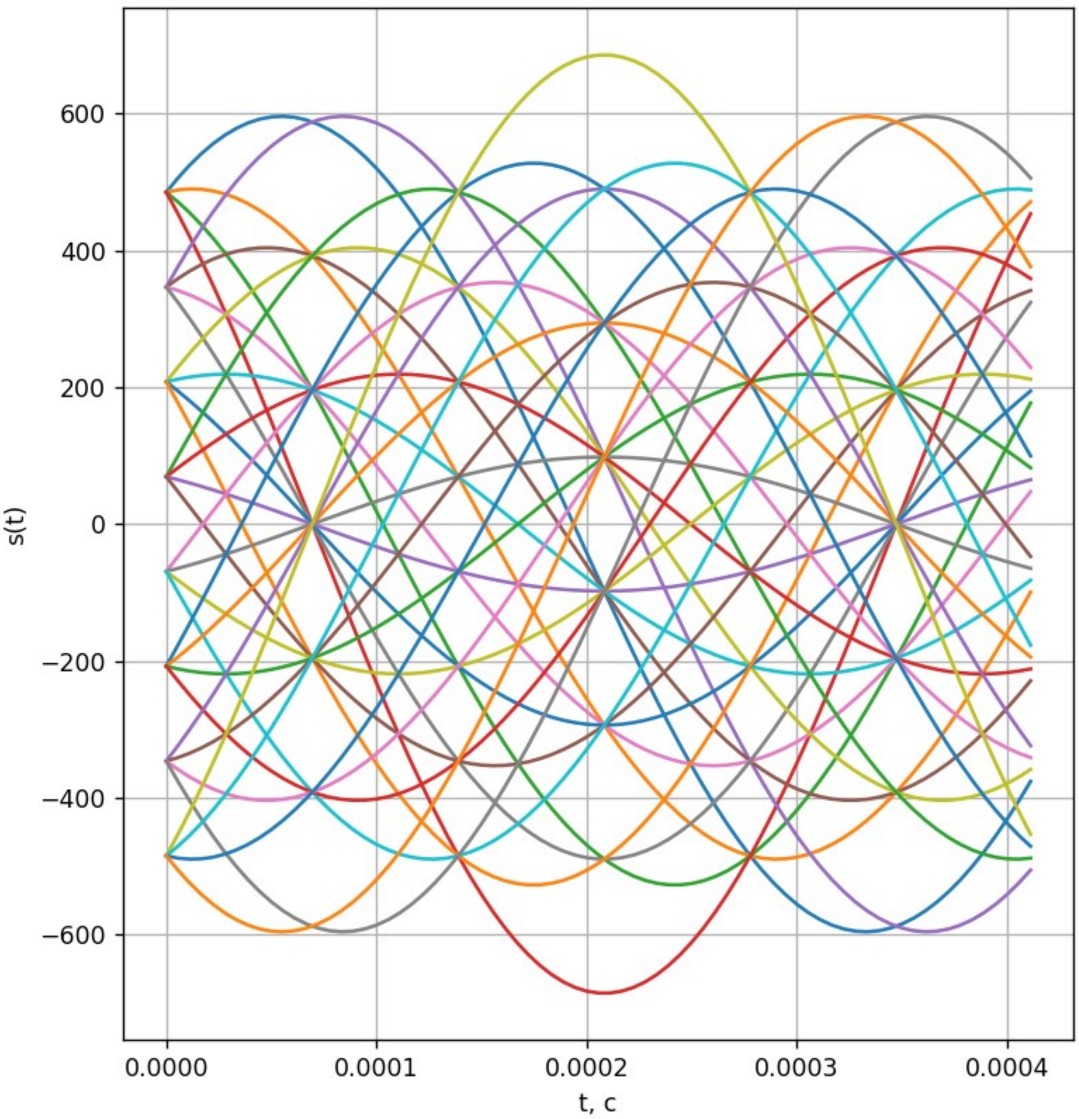

Рис. 1 — Графики сигналов

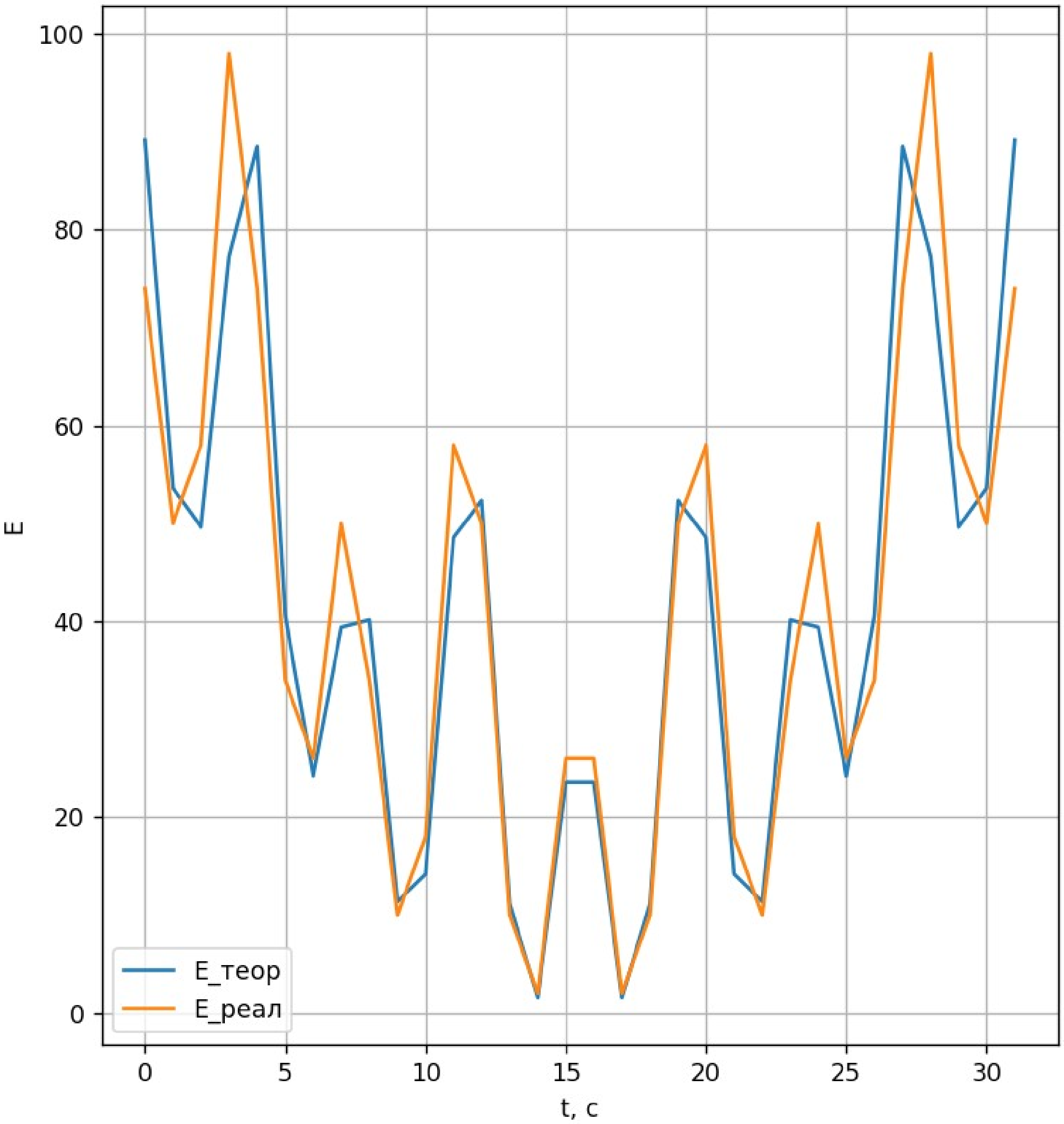

Рис. 2 — Графики энергий

6) Выводы

В ходе лабораторной работы

- Было определено количество сигналов при заданных характеристиках **q = 32**.
- Были приведены аналитические выражения сигналов, подсчитана энергия каждого сигнала.
- Были построены графики сигналов, показанные на рис. 1, и графики энергий, показанные на рис. 2.
- В ходе сравнения энергий было выявлено несовпадение теоретической и реальной энергии, вызванное тем, что модуляционная скорость больше несущей частоты

Приложение 1

Листинг исходного кода на языке *Python*

```
import numpy as np
from matplotlib import pyplot as plt

def getSs(q):
    m = np.log2(q)
    s1 = []
    s2 = []

    t = 0
    step2 = 1
    if m % 2:
        t = np.sqrt(2 * q)
        step2 = 2
    else:
        t = np.sqrt(q)
    A = t - 1

    i1 = 0
    while i1 < t:
        i2 = 0
        if (m % 2) and (i1 % 2 == 0):
            i2 = 1
        while i2 < t:
            s1.append((A * (1 - (2 * i1) / (t - 1))))
            s2.append((A * (1 - (2 * i2) / (t - 1))))
            i2 += step2
        i1 += 1

    return s1, s2, A

def getSi(f0, T, dt, s1, s2):
    x = np.arange(0, T, dt)
    s = []
    for i in range(len(s1)):
        temp = []
        t1 = s1[i] * np.sqrt(2 / T)
        t2 = s2[i] * np.sqrt(2 / T)
        for ii in range(x.size):
            temp.append(t1 * np.cos(2 * np.pi * f0 * x[ii]) + t2 * np.sin(2 * np.pi * f0 * x[ii]))
        s.append(temp)

    return x, s
```

```
def drawArrays(x, ys, s = '-', xL ='', yL ='', linesLabel=None, sub = 0):
    if sub != 0:
        plt.subplot(sub)

    if linesLabel is None:
        for y in ys:
            plt.plot(x, y, s)
    else:
        i = 0
        for y in ys:
            plt.plot(x, y, s, label=linesLabel[i])
            i += 1
        plt.legend()

    plt.ylabel(yL)
    plt.xlabel(xL)
    plt.grid()

def QAM(f0, Vmod, Vinf):
    T = 1 / Vmod
    q = pow(2, Vinf / Vmod)
    s1, s2, A = getSs(q)

    dt = 1 / (f0 * 100)
    x, s = getSi(f0, T, dt, s1, s2)

    drawArrays(x, s, xL='t, c', yL='s(t)', sub=121)

    e1 = []
    e2 = []
    for i in range(len(s)):
        in1 = 0
        for ii in s[i]:
            in1 += ii*ii
        e1.append(in1 * dt)
        e2.append(s1[i]*s1[i] + s2[i]*s2[i])

    drawArrays(np.arange(0, len(s)), [e1, e2], xL='t, c', yL='E', linesLabel=['E_теор', 'E_реал'], sub=122)

    plt.show()

if __name__ == "__main__":
    QAM(1800, 2400, 12000)
```