

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение высшего образования
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА №25

ОТЧЕТ ЗАЩИЩЕН С ОЦЕНКОЙ _____

ПРЕПОДАВАТЕЛЬ

ассистент

должность, уч. степень, звание

подпись, дата

Н.В. Степанов

инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ № 2

ИССЛЕДОВАНИЕ ДИСКРЕТНЫХ СИГНАЛОВ В ЧАСТОТНОЙ
ОБЛАСТИ

по курсу: ОБЩАЯ ТЕОРИЯ СВЯЗИ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №

3032

номер группы

подпись, дата

В.Д. Кибилев

инициалы, фамилия

Санкт-Петербург
2022

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

$q = 32$ – количество сигналов

1) Цель работы

- Исследовать дискретный сигнал в дискретной области.

2) Вывод выражений спектра отрезка гармоник

Имеется сигнал:

$$s_i(t) = \begin{cases} A \cos(2\pi f_0 t), & \text{если } 0 < t < T \\ 0, & \text{в противном случае} \end{cases}$$

Спектр данного сигнала можно рассчитать следующим образом:

$g(t)$ – некоторая произвольная функция (ограничивающая)

$c(t) = \cos(2\pi f_0 t + \theta)$ – гармонический сигнал (несущая)

По теореме о свертке:

$$S_c(f) = G(f) \cdot C(f), \text{ где } g(t) \Leftrightarrow G(f), \text{ где } C(f) = \frac{\delta(f-f_0) + \delta(f+f_0)}{2}$$

Следовательно:

$$S_c(f) = G(f) \cdot \frac{\delta(f-f_0) + \delta(f+f_0)}{2} = \frac{G(f-f_0) + G(f+f_0)}{2} \quad (1.1)$$

Рассмотрим функцию

$$g(t) = \begin{cases} A, & \text{если } 0 < t < T \\ 0, & \text{в противном случае} \end{cases}$$

Подставим $g(t)$ в формулу прямого преобразования Фурье:

$$G(f) = \int_0^T g(t) e^{-j2\pi ft} dt = \frac{A}{-j2\pi f} (e^{-j2\pi fT} - 1) = \frac{AT \sin(\pi fT)}{\pi fT} e^{-j\pi fT} = AT \operatorname{sinc}(fT) e^{-j\pi fT} \quad (1.2)$$

Подставим в выражение (1.1) формулу (1.2):

$$S_c(f) = \frac{AT}{2} (\operatorname{sinc}((f-f_0)T) + \operatorname{sinc}((f+f_0)T)) e^{-j\pi fT} \quad (1.3)$$

Имеется сигнал:

$$s_i(t) = \begin{cases} A \sin(2\pi f_0 t), & \text{если } 0 < t < T \\ 0, & \text{в противном случае} \end{cases}$$

Спектр данного сигнала можно рассчитать следующим образом:

$g(t)$ – некоторая произвольная функция (огибающая)

$c(t) = \sin(2\pi f_0 t)$ – гармонический сигнал (несущая)

По теореме о свертке:

$$S_c(f) = G(f) \cdot C(f), \text{ где } \begin{matrix} g(t) \Leftrightarrow G(f) \\ c(t) \Leftrightarrow C(f) \end{matrix}, \text{ где } C(f) = \frac{\delta(f-f_0) + \delta(f+f_0)}{2j}$$

Следовательно:

$$S_c(f) = G(f) \cdot \frac{\delta(f-f_0) + \delta(f+f_0)}{2j} = \frac{G(f-f_0) + G(f+f_0)}{2j} \quad (1.4)$$

Рассмотрим функцию

$$g(t) = \begin{cases} A, & \text{если } 0 < t < T \\ 0, & \text{в противном случае} \end{cases}$$

Подставим $g(t)$ в формулу прямого преобразования Фурье:

$$G(f) = \int_0^T g(t) e^{-j2\pi ft} dt = \frac{A}{-j2\pi f} (e^{-j2\pi fT} - 1) = \frac{AT \sin(\pi fT)}{\pi fT} e^{-j\pi fT} = AT \operatorname{sinc}(fT) e^{-j\pi fT} \quad (1.5)$$

Подставим в выражение (1.5) формулу (1.4):

$$S_c(f) = \frac{AT}{2j} (\operatorname{sinc}((f-f_0)T) + \operatorname{sinc}((f+f_0)T)) e^{-j\pi fT} \quad (1.6)$$

Выражение для преобразования Фурье:

$$S_c(f) = \sqrt{\frac{ET}{2}} (\operatorname{sinc}((f-f_0)T) + \operatorname{sinc}((f+f_0)T)) e^{-j\pi fT} \quad (1.3)$$

3) Графики

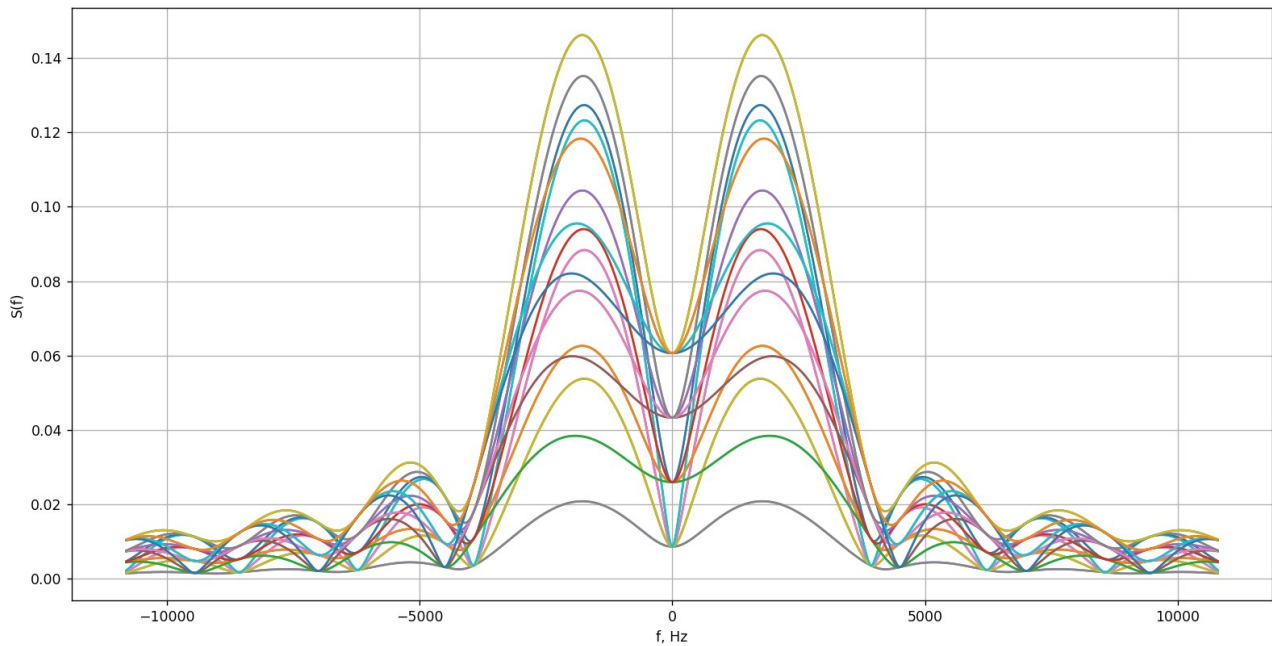


Рис. 1 — График спектров сигналов

Из-за того, что несущая частота меньше информационной скорости, определить точные ширины полос по графику, затруднительно.

Ширина полосы частот = $2V_m = 4800$ Гц

4) Вывод выражения спектра последовательности сигналов

Последовательность сигналов: $s_i(t) = \sum_{l=1}^{N-1} S_{i_l}(t-lT)$

Воспользуемся свойствами преобразования Фурье:

1. Задержка: $s(t-\tau) = S(f)e^{-j2\pi f\tau}$
2. Линейность: $\begin{cases} g(t) \Leftrightarrow G(f) \\ h(t) \Leftrightarrow H(f) \end{cases} \Rightarrow a \cdot g(t) + b \cdot h(t) \Leftrightarrow a \cdot G(f) + b \cdot H(f)$

Следовательно:

$$s_i(t) = \sum_{l=1}^{N-1} S_{i_l}(f) e^{-j2\pi f l T}$$

5) Графики

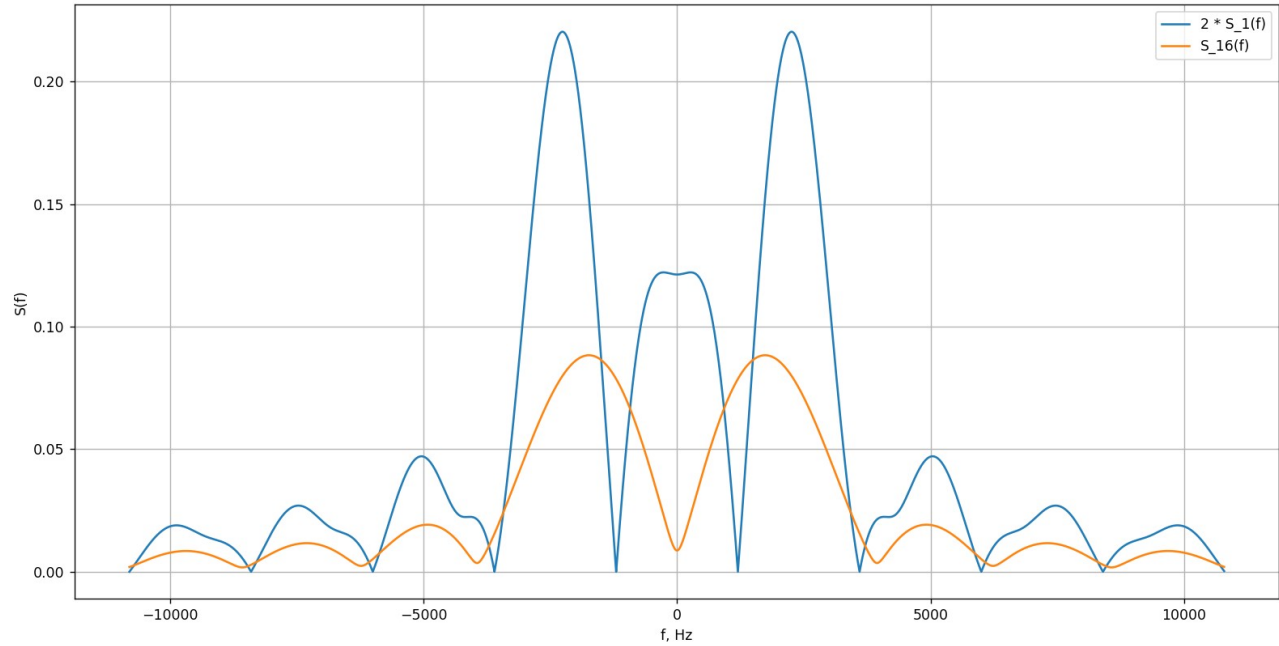


Рис. 2 — Спектр последовательности длиной 2 1-го сигнала

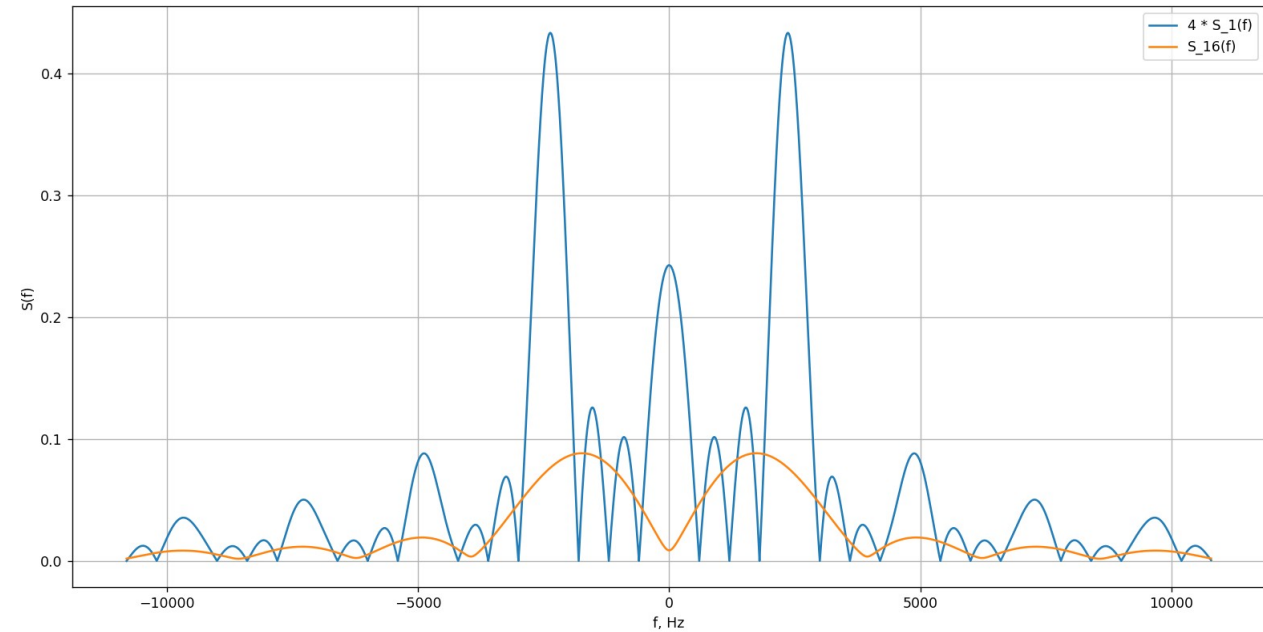


Рис. 3 — Спектр последовательности длиной 4 1-го сигнала

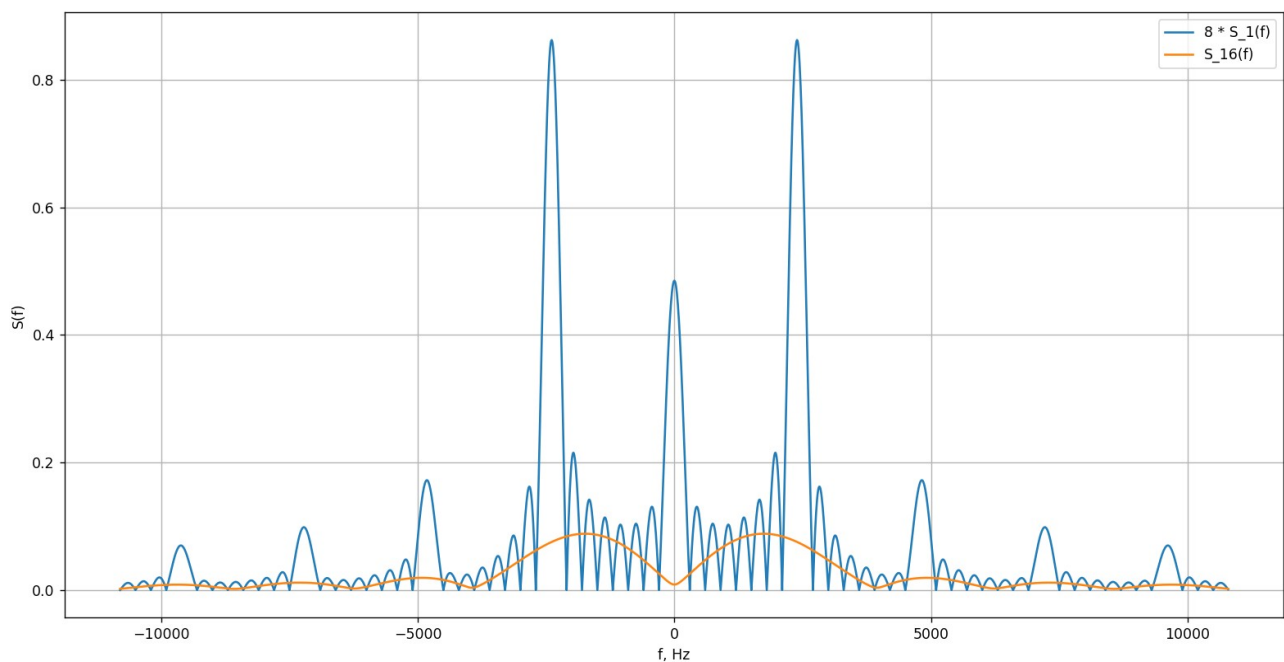


Рис. 4 — Спектр последовательности длиной 8 1-го сигнала

Изменение ширины полосы частот при последовательностях разной длины:

N	s1
2	3600-1200 = 2400
4	3000-1800 = 1200
8	2700-2100 = 600

Исходя из полученных результатов можно сделать вывод, что, задавая спектр последовательности длины N , его ширина полосы частот уменьшается в N раз, относительно ширины полосы частот спектра, заданного одним индексом.

6) Вывод

В ходе лабораторной работы

- Были выведены формулы преобразования Фурье для отрезков гармоник сигналов косинуса (1.3) и синуса (1.6)
- Была выведена формула выражения спектра последовательности сигнала. Было проведено преобразование Фурье для сигнала с частотной модуляцией, определены их амплитудные спектры и ширина полосы частот.

Листинг исходного кода на языке *Python*

```
import random
```

```
import numpy as np
```

```
from matplotlib import pyplot as plt
```

```
import O_L1 as L1
```

```
def ft(q, s1, s2, T, f0):
```

```
    f = np.arange(-6 * f0, 6 * f0, 10)
```

```
    sp = []
```

```
    for i in range(0, len(s1)):
```

```
        si = np.zeros(len(f), dtype=np.complex_)
```

```
        for ii in range(0, len(f)):
```

```
            si[ii] = (s1[i]*np.sqrt(T/2)*(np.sinc((f[ii] - f0) * T) + np.sinc((f[ii] + f0) *  
T))*np.exp(-1j * np.pi * f[ii] * T)
```

```
                    + (s2[i] / 1j)*np.sqrt(T/2)*(np.sinc((f[ii] - f0) * T) - np.sinc((f[ii] + f0) *  
T))*np.exp(-1j * np.pi * f[ii] * T))
```

```
            sp.append(si)
```

```
    return f, sp
```

```
def getModules(sp):
```

```
    out = []
```

```
    for i in sp:
```

```
        t = []
```

```
        for ii in i:
```

```
            t.append(np.sqrt(np.power(ii.real, 2) + np.power(ii.imag, 2)))
```

```
        out.append(t)
```

```
    return out
```

```

def QAM_ft(f0, Vmod, Vinf):
    T = 1 / Vmod
    q = pow(2, Vinf / Vmod)
    s1, s2, A = L1.getSs(q)

    dt = 1 / (f0 * 100)
    x, s = L1.getSi(f0, T, dt, s1, s2)

    freq, sp = ft(len(x), s1, s2, T, f0)
    m = getModules(sp)
    L1.drawArrays(freq, m, xL='f, Hz', yL='S(f)')

    plt.show()

def getI(q, n):
    out = []
    for i in range(n):
        out.append(random.randint(0, q - 1))

    return out

def getSI(s, x, I, T):
    new_x = np.zeros(len(x) * len(I))
    for i in range(len(x)):
        for ii in range(0, len(I)):
            new_x[len(x) * ii + i] = x[i] - T * (len(I) - 1 - ii)

    new_s = np.zeros(len(new_x))
    for i in range(len(x)):
        for ii in range(0, len(I)):
            new_s[len(x) * ii + i] = s[I[ii]][i] - T * (len(I) - 1 - ii)

    return new_x, new_s

def getSIF(q, s1, s2, T, f0, I):
    f, sp = ft(q, s1, s2, T, f0)
    out = np.zeros(len(f), dtype=np.complex_)
    for l in range(0, len(I)):
        for ii in range(0, len(f)):
            out[ii] += sp[I[l]][ii] * np.exp(-1j * 2 * np.pi * f[ii] * T * l)

    return f, out

```



```

def Serial_QAM(f0, Vmod, Vinf, signalsNumber = 4):
    T = 1 / Vmod
    q = pow(2, Vinf / Vmod)
    s1, s2, A = L1.getSs(q)

    dt = 1 / (f0 * 100)
    x, s = L1.getSi(f0, T, dt, s1, s2)

    I1 = getI(1, signalsNumber)
    xI1, SI1 = getSI(s, x, I1, T)
    fi1, SIF1 = getSIF(q, s1, s2, T, f0, I1)

    m = getModules([SIF1])
    L1.drawArrays(fi1, m, linesLabel=[(str(signalsNumber) + ' * S_1(f)')])

    freq, sp = ft(len(x), s1, s2, T, f0)
    m = getModules(sp)

    L1.drawArrays(freq, [m[int(len(m) / 2)]], xL='f, Hz', yL='S(f)', linesLabel=['S_16(f)'])
    plt.grid()

    plt.show()

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

    Serial_QAM(1800, 2400, 12000, 2)

```