

Московский государственный технический университет им. Н.Э.  
Баумана

Факультет «Радиоэлектроника и лазерная техника (РЛ)»

Кафедра «Технология приборостроения (РЛ6)»

---

Лабораторная работа №4  
по дисциплине «Цифровая обработка сигналов»

Выполнил ст. группы РЛ6-71

Филимонов С. В.

Преподаватель Дмитриев Д. Д.

Москва, 2023

ЦЕЛЬ РАБОТЫ – изучение методов определения основных свойств случайных сигналов, таких, как мощность, спектральная плотность мощности (СПМ) и автокорреляционная функция (АКФ), получение практических навыков анализа случайных сигналов в среде Matlab.

### Задание № 1

Разработайте в Matlab функцию `[P, P_dB] = calc_power(x)` для вычисления мощности сигнала по формуле

$$P_a = \frac{1}{N} \sum_{n=0}^{N-1} x^2(n).$$

Функция также должна выдавать значение мощности в децибелах

$$P_{dB} = 10 \log_{10} P_a.$$

Посчитайте мощность следующих сигналов при  $n = 0, 1, \dots, 127$ :

$$\begin{aligned} x_1(n) &= 0,5 \sin\left(2\pi \frac{3}{128}n\right), \\ x_2(n) &= 0,5 \sin\left(2\pi \frac{30}{128}n\right), \\ x_3(n) &= 0,3 \cos\left(2\pi \frac{6}{128}n\right) + 0,4 \cos\left(2\pi \frac{10}{128}n\right). \end{aligned}$$

В ходе выполнения задания нам надо было написать функцию, которая считает мгновенную мощность сигнала сигнала  $x(t)$ . А так же вычисления мощности в децибелах. После чего в функцию необходимо передать сигналы  $x_1, x_2, x_3$ . Функция написанная на языке Python:

```
def calc_power(x):  
    P = (sum(i * i for i in x) / len(x))  
    P_dB = 10 * np.log10(P)  
    return [P, P_dB]
```

Результат:

Мощность для первого сигнала [0.125, -9.030899869919436]

Мощность для второго сигнала [0.12499999999999999, -9.03089986991944]

Мощность для третьего сигнала [0.12499999999999996, -9.030899869919438]

## Задание № 2

Разработайте в Matlab функцию для вычисления периодограммы сигнала

$$P_{xx}(k) = \frac{1}{N} |X(k)|^2, k = 0, 1, \dots, N-1.$$

Постройте периодограммы сигналов из задания 4.2.1. Вычислите мощность сигналов из полученных периодограмм по формуле

$$P_a = \frac{1}{N^2} \sum_{k=0}^{N-1} |X(k)|^2.$$

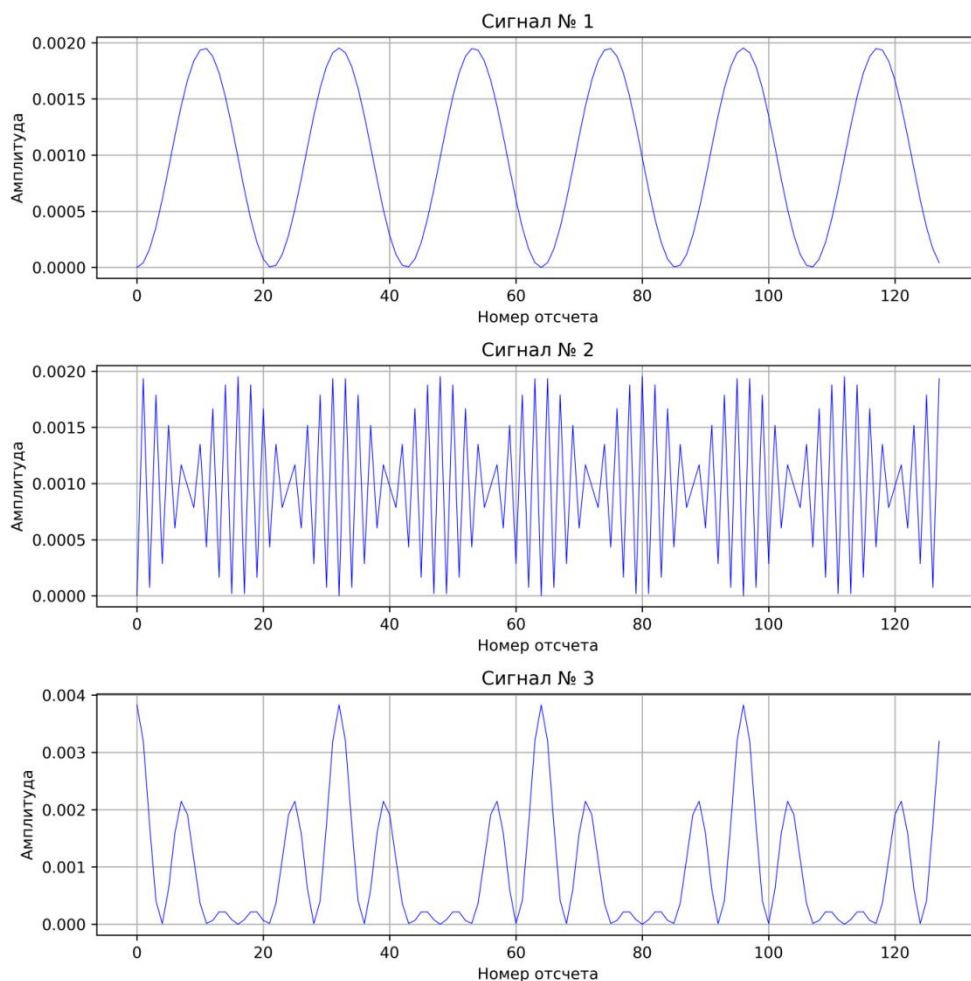
По заданию необходимо вычислить периодограммы сигнала, по формуле 4.3. Сигнала из первого задания. А так же необходимо вычислить мощность полученной периодограммы. Напишем функцию в Python:

```
def calc_pxx(x):  
    Pxx = [(np.fabs(i) ** 2) / len(x) for i in x]  
    Pa = calc_power(Pxx)  
    return [Pxx, Pa]
```

Полученные мощности сигналов:

```
Мощность для первого сигнала [1.4305114746093748e-06, -58.44508654223943]  
Мощность для второго сигнала [1.4305114746093752e-06, -58.44508654223943]  
Мощность для третьего сигнала [2.089691162109375e-06, -56.799178940337185]
```

Полученные периодограммы:



### Задание № 3

Предположим, что зарегистрированный сигнал представляет собой сумму полезного сигнала и белого шума ( $n = 0, 1, \dots, 511$ ):

$$x(n) = 0.4 \cos(2\pi \frac{9}{128} n) + A \cdot r(n).$$

Определите максимальное отношение сигнал/шум (ОСШ), при котором по периодограмме  $x(n)$  можно обнаружить в нем наличие полезного сигнала. Отношение сигнал/шум (англ. signal-to-noise ratio, SNR) – безразмерная величина, равная отношению мощности полезного сигнала к мощности шума:

$$\text{SNR(dB)} = 10 \log_{10} \frac{P_{\text{signal}}}{P_{\text{noise}}}.$$

Для определения требуемого ОСШ необходимо выполнить несколько экспериментов, плавно изменяя параметр  $A$  и наблюдая изменение периодограммы сигнала. Будем считать, что сигнал можно обнаружить, если уровень полезного сигнала на 10 дБ больше уровня компонент спектра, образуемых шумовым сигналом. Белый шум  $r(n)$  можно получить в Matlab при помощи функции.

По заданию необходимо сделать симуляцию реальной задачи, в которой присутствует сигнал с информацией и белый шум. Для этого надо определить оптимальное отношение сигнал/шум. Коэффициент, который определяет это отношение это SNR, в децибеллах. Код написанный на Python:

```
def calc_Ps(x):
    Ps = []
    for i in x:
        Ps.append(calc_power(i)[0])
    return Ps

n = np.arange(0, 512, 1)
A = [0.01, 0.05, 0.1, 0.25, 0.3, 0.4, 0.5, 1, 5, 10, 50, 100, 1000]
rng = np.random.default_rng()
r = rng.integers(low=-1, high=1, size=512)
x = 0.4 * np.cos(2 * np.pi * 9 / 128 * n)
Psignal = calc_power(x)[0]
Ar = []
for a in A:
    Ar.append(a * r)
Pnoise = calc_Ps(Ar)
SNR = 10 * np.log10(Psignal / Pnoise)
print(SNR)
```

[ 32.10959407	18.13019399	12.10959407	4.1507939	2.56716898
0.06839425	-1.86980601	-7.89040593	-21.86980601	-27.89040593
-41.86980601	-47.89040593	-67.89040593		

Переменная  $A$  это амплитуда сигнала шума. Чем выше коэффициент SNR, тем тем меньше искажение. Если коэффициент отрицательный, то сигнал не разобрать.

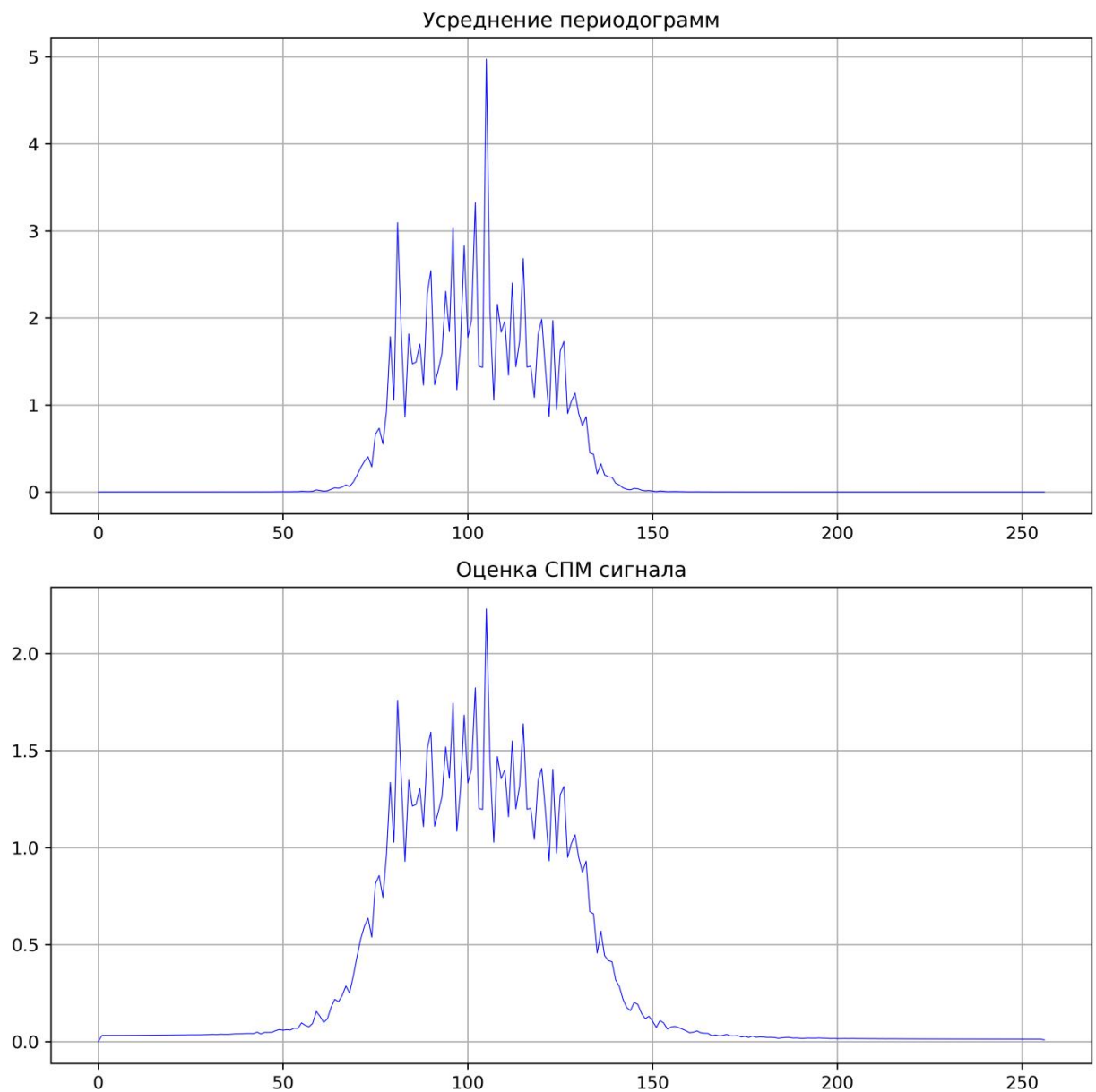
## Задание № 4

Оцените СПМ сигнала на выходе фильтра при помощи метода усреднения периодограмм. Параметры *NSAMP* и *NSHIFT* подберите самостоятельно. На вход фильтра поступает 4096 отсчетов белого шума. Коэффициенты фильтра

$b = [0.00482434 \ 0 \ -0.0192973 \ 0 \ 0.02894606 \ 0 \ -0.0192973 \ 0 \ 0.00482434];$

$a = [1 \ -2.06958023 \ 3.99771255 \ -4.3894077 \ 4.45285533 \ -2.9060422 \ 1.75168470 \ -0.5862147 \ 0.18737949];$

В ходе задания необходимо оценить спектральную плотность сигнала, в границах *NSAMP*, со сдвигом *NSHIFT*. Для начала применили к белому фильтру с параметрами *b* и *a*. После вычислили периодограммы и усреднили их (1 график). После чего провели оценку СПМ(2 график).



## Задание № 5

Не удалось найти нужные файлы.

## Задание № 6

Разработайте в Matlab функцию для отделения вокализованных участков речи от невокализованных:  $[x_v, x_u] = vu\_separate(x, N, threshold)$  Сепарация (разделение) происходит на основе анализа кратковременной мощности сигнала (см. (4.7)). На вход функции поступает речевой сигнал  $x$ , длина окна анализа на котором считается мощность  $N$  и значение порога  $threshold$ . Если мощность сигнала в момент времени  $n$  больше порога, то считается, что отсчет  $x(n)$  является вокализованным, в противном случае – невокализованным. Сепарация выполняется следующим образом:

а) для входного сигнала  $x(n)$  рассчитывается кратковременная мощность  $P(n)$  по выражению (4.7);

б) рассчитывается признак вокализованности:  $v(n) = \{1, P(n) > threshold; 0, P(n) \leq threshold\}$ ;

в) вычисляется сепарация сигнала:  $x_v(n) = x(n) \cdot v(n)$ ,  $x_u(n) = x(n) \cdot (1 - v(n))$ .

Удобно мощность сигнала посчитать в децибелах:  $P_{dB}(n) = 10 \log_{10} P(n)$  и порог для сепарации также подбирать в децибелах.

По заданию необходимо написать функции, алгоритм функции изложен в пунктах а, б, в.

```
def vu_separate(x, N, threshold):  
    P = np.abs(x) ** 2  
    P_dB = 10 * np.log10(P)  
    v = P_dB > threshold  
    x_v = x * v  
    x_u = x * (1 - v)  
    return x_v, x_u
```

Вывод:

Я научился определять свойства случайных параметров, считать мощность таких сигналов и СПМ.