

МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ
КОММУНИКАЦИЙ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Сибирский государственный университет телекоммуникаций и
информатики»

Кафедра телекоммуникационных систем и вычислительных средств
(ТС и ВС)

Отчет по лабораторной работе №1
по дисциплине
Основы систем мобильной связи

по теме:
ВРЕМЕННАЯ И ЧАСТОТНАЯ ФОРМЫ СИГНАЛОВ.
ПРЕОБРАЗОВАНИЯ ФУРЬЕ. ДИСКРЕТИЗАЦИЯ СИГНАЛОВ

Студент:
Группа ИА-331

Я.А Гмыря

Предподаватель:
Заведующая кафедрой ТВ и ВС

В.Г Дроздова

Новосибирск 2025 г.

СОДЕРЖАНИЕ

ЦЕЛЬ И ЗАДАЧИ	3
1 ЗАДАЧА №1	5
2 ЗАДАЧА 2.....	6
3 ЗАДАЧА 3.....	7
4 ЗАДАЧА 4.....	8
5 ЗАДАЧА 5.....	10
6 ЗАДАЧА 6.....	12
7 ЗАДАЧА 7.....	13
8 ЗАДАЧА 8.....	14
9 ЗАДАЧА 9.....	15
10 ЗАДАЧА 10	16
11 ЗАДАЧА 11	17
12 ЗАДАЧА 12	19
13 ЗАДАЧА 13	21

ЦЕЛЬ И ЗАДАЧИ

Цель: Получить представление о формах радиосигналов, их частотном и временном представлении, а также о преобразованиях Фурье и аналогово-цифровых преобразованиях сигналов, частоте дискретизации сигналов.

Задачи:

1. Необходимо сгенерировать и визуализировать (вывести на график) непрерывный сигнал.
2. Определить максимальную частоту в спектре данного сигнала.
3. Определить минимальную необходимую частоту дискретизации полученного сигнала (теорема Котельникова).
4. Оцифровать сигнал с полученной частотой дискретизации, выбрав требуемое число отсчетов сигнала на длительности 1 секунда, сохранить полученные значения в массив, пока не озадачиваясь разрядностью АЦП, просто выбранные с частотой дискретизации значения с теми уровнями, которые имеет функция в полученных точках.
5. Выполнить прямое дискретное преобразование Фурье для массива временных отсчетов сигнала и оценить ширину данного спектра. А также объем памяти, требуемый для хранения данного массива (тип переменных в массиве – на ваше усмотрение, float, int и пр.).
6. Восстановите оригинальный аналоговый сигнал по массиву имеющихся у вас отсчетов, соединив их непрерывной линией и оцените визуальное сходство оригинального сигнала и восстановленного после оцифровки.
7. Увеличьте частоту дискретизации в 4 раза и проделайте задания из п.4-6.
8. Запишите аудиофайл со своим голосом (например, в формате wav). Проанализируйте визуально спектр голоса. Определите максимальную частоту в спектре данного сигнала и выберите требуемую для оцифровки частоту дискретизации.
9. Используя библиотеки работы со звуком Matlab (или Python) проанализируйте имеющуюся у вас запись голоса.
10. Определите частоту дискретизации, которая была использована при записи голоса на цифровой носитель. Для этого возьмите количество элементов в файле с записью и разделите на длительность записи в секундах. Например,

если у вас файл состоит из 441 000 элементов (отсчетов) и имеет длительность 10 секунд, то частота дискретизации F_s будет равна $441\,000/10 = 44\,100$ Гц.

11. Проредите полученный массив (иными словами – уменьшите частоту дискретизации) и воспроизведите полученный сигнал. Обратите внимание на качество звучания при неверно взятой частоте дискретизации:

12. Выполните прямое дискретное преобразование Фурье для оригинального звучания и для прореженного сигнала, выведите на график амплитудный спектр сигнала, определите его ширину.

13. Оцените влияние разрядности АЦП на спектр сигнала. Для этого нужно написать функцию, которая бы округляла значения отсчетов сигнала, заданного в вашем варианте, до какого-то числа, определяемого разрядностью АЦП. Допустим если у АЦП всего 3 разряда, то диапазон возможных дискретных значений амплитуд временных отсчетов сигнала – это 0..7. Все значения больше 7 округляются до 7. Для результирующего дискретного сигнала требуется выполнить прямое преобразование Фурье. Сравнить полученный спектр со спектром исходной синусоиды, отсчеты которой не подвергались квантованию по уровню. Вывести среднюю ошибку квантования для случаев, когда разрядность АЦП равна 3/4/5/6.

ЗАДАЧА №1

Изобразим следующий сигнал на $t \in [0; 2]$ с шагом 0.05 при помощи MATLAB:

$$y(t) = \sin\left(12\pi ft + \frac{\pi}{11}\right) + \sin(10\pi ft), \quad f = 4$$

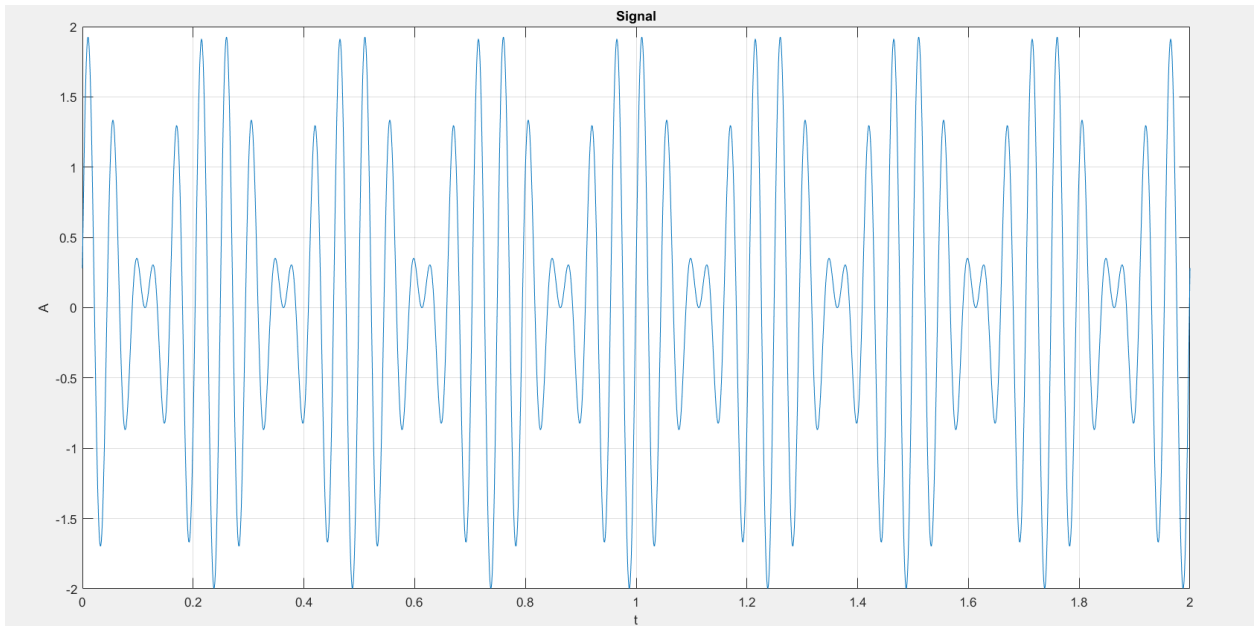


Рисунок 1 — Пример сигнала

ЗАДАЧА 2

Обратим внимание, что наш сигнал состоит из двух компонент (они составляют спектр сигнала):

$$y_1(t) = \sin\left(12\pi ft + \frac{\pi}{11}\right), \quad y_2(t) = \sin(10\pi ft)$$

В общем виде гармонический сигнал записывается как:

$$S(t) = A \sin(\omega t + \varphi),$$

где

- A — амплитуда сигнала (в нашем случае $A = 1$),
- ω — угловая частота,
- φ — фаза (сдвиг).

Угловая частота ω связана с обычной частотой f через уравнение:

$$f = \frac{\omega}{2\pi}.$$

// Найдем частоты компонент при $f = 4$ Гц:

$$f_1 = \frac{12\pi \cdot 4}{2\pi} = 24 \text{ Гц}, \quad f_2 = \frac{10\pi \cdot 4}{2\pi} = 20 \text{ Гц}.$$

Максимальная частота в спектре данного сигнала составляет 24 Гц.

ЗАДАЧА 3

Теорема Котельникова утверждает, что минимальная необходимая частота дискретизации должна быть как минимум вдвое больше максимальной частоты в спектре сигнала:

$$F_s = 2F_{\max}.$$

Для нашего сигнала имеем $F_{\max} = 24$ Гц, следовательно:

$$F_s = 2 \cdot 24 \text{ Гц} = 48 \text{ Гц}.$$

ЗАДАЧА 4

Необходимо оцифровать сигнал с частотой дискретизации $F_s = 48$ на длительности 1 секунда. Это значит, что нужно найти значения сигнала на промежутке $t \in [0; 1]$ с шагом $1/48$ и записать в массив.

Визуально это можно представить себе так:

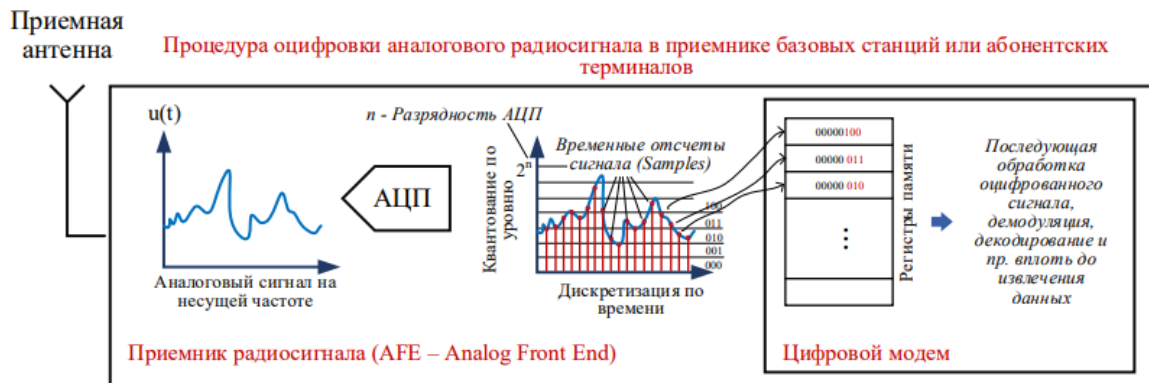


Рис. 2. Аналого-цифровое преобразование радиосигнала.

Рисунок 2 — Пример оцифровки сигнала

Реализация:

Создаем массив с $t \in [0; 1]$ с шагом $1/48$. Создаем массив семплов с размером, равным кол-ву семплов (размеру массива t). Перебираем значения t , считаем размер сигнала и пишем в массив с семплами.

```
%digitization
t = 0:1/48:1;

samples = zeros(1, length(t));
pos = 0;

for t_k = 1:length(t)
    samples(pos) = sin(12*pi*f*t_k+pi/11) + sin(10*pi*f*t_k);
    pos = pos + 1;
end
```

Рисунок 3 — Реализация оцифровки сигнала

Посмотрим, что хранит массив семплов после расчетов:



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1	0.2817	0.2183	-0.5843	0.7183	-0.5843	0.2183	0.2817	-0.7817	1.1478	-1.2817	1.1478	-0.7817	0.2817	0.2183	-0.5843	0.7183	-0.5843	0.2183	0.2817	-0.7817	1.1478
2																					

Рисунок 4 — Пример семплов

ЗАДАЧА 5

Необходимо провести прямое дискретное преобразование Фурье над массивом семплов, полученных на прошлом шаге.

Прямое дискретное преобразование Фурье имеет вид:

$$F(x) = \sum_{n=0}^{N-1} f(n) e^{-j2\pi \frac{xn}{N}}, \quad (1)$$

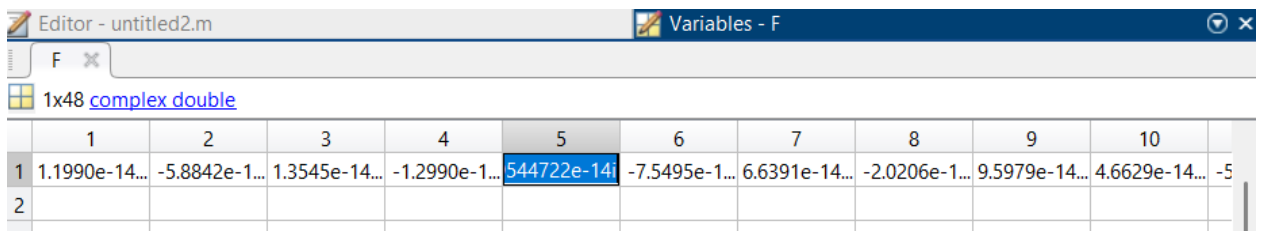
где

- N — размерность преобразования Фурье,
- $F(x)$ — сигнал в частотной области,
- $f(n)$ — дискретизированный сигнал во временной области.

```
25 %DFT
26 N = length(samples);
27 F = zeros(1, N);
28
29 for x = 0:N-1
30     sum = 0;
31     for n = 0:N-1
32         sum = sum + samples(n+1) * exp(-1j * 2 * pi * x * n / N);
33     end
34     F(x+1) = sum;
35 end
36
37
```

Рисунок 5 — Реализация ППФ

Посмотрим на массив F после выполнения программы:



The screenshot shows the MATLAB Editor with a file named 'untitled2.m' and the Variables window displaying the array F. The array F is a 1x48 complex double array. The first row of the array is visible, showing 10 columns of values. The 5th element is highlighted in blue.

	1	2	3	4	5	6	7	8	9	10	
1	1.1990e-14...	-5.8842e-1...	1.3545e-14...	-1.2990e-1...	544722e-14i	-7.5495e-1...	6.6391e-14...	-2.0206e-1...	9.5979e-14...	4.6629e-14...	-5
2											

Рисунок 6 — Результат ППФ

Получили массив комплексных чисел типа `complex double`, где реальная и мнимая часть являются типом `double`, который занимает 8 байт, т.е один `complex double` занимает 16 байт. Всего в массиве 48 элементов, а это значит, что он занимает в памяти 768 байт.

Вычислим ширину спектра сигнала

????????????????????????????????????

ЗАДАЧА 6

Восстановим сигнал после оцифровки:

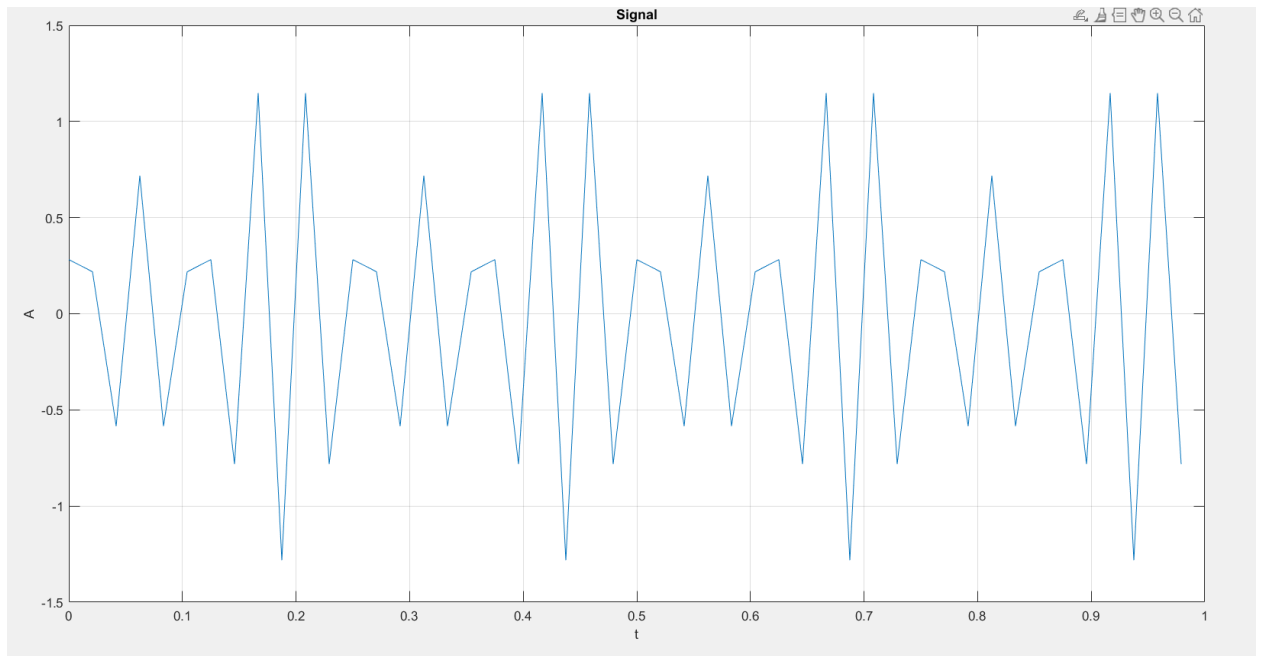


Рисунок 7 — Сигнал после оцифровки ($F_s = 48$)

Данный сигнал слабо похож на оригинал, т.к мы выбрали самую минимальную частоту дискретизации и 48 точек слишком мало для корректного построения графика.

ЗАДАЧА 7

Увеличим частоту дискретизации в 4 раза (до 192Гц) и посмотрим, что получится:

Поскольку теперь у нас 192 элемента `complex double`, то теперь результат ППФ будет занимать 3072 байта (в 4 раза больше, чем раньше).

График теперь выглядит так:

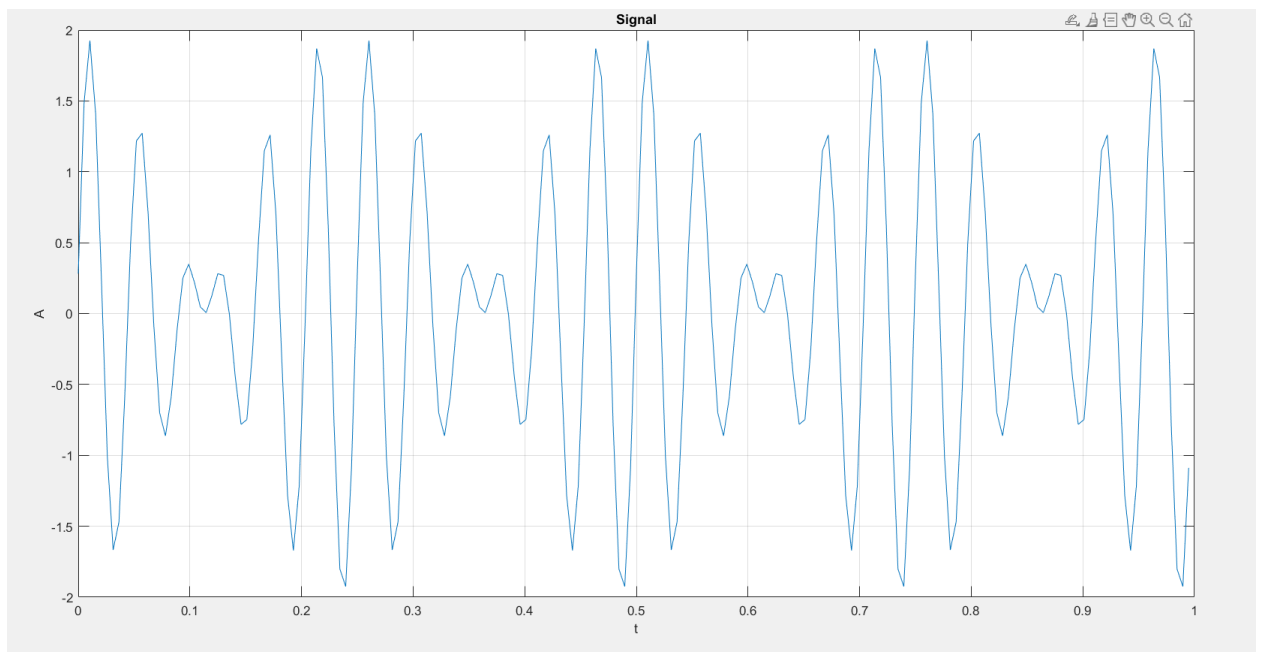


Рисунок 8 — Сигнал после оцифровки ($F_s = 192$)

Теперь график намного больше похож на оригинал, но все-таки недостаточно плавный.

ЗАДАЧА 8

Я записал свой голос и через приложение построил график его спектра:

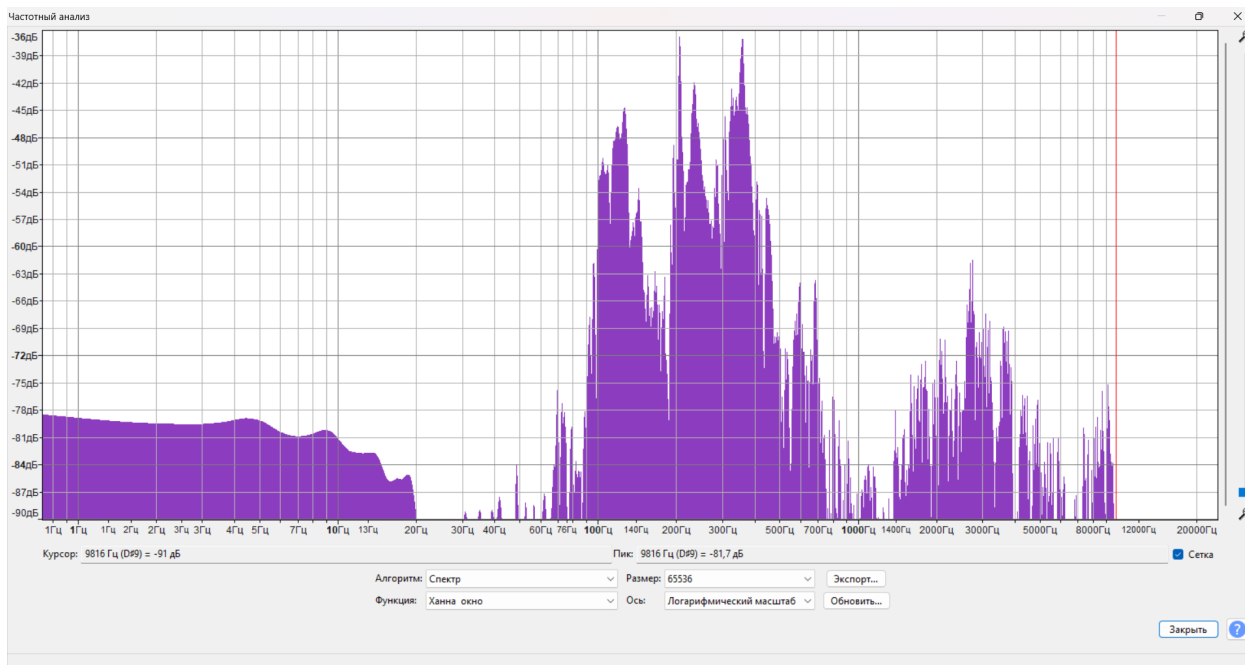


Рисунок 9 — Спектр голоса

Максимальная частота в спектре моего голоса - 9816Гц, минимальная частота дискретизации для этой записи будет составлять $2 * 9816\text{Гц} = 19632\text{Гц}$

ЗАДАЧА 9

С помощью встроенных функции MATLAB откроем аудиофайл с голосом и узнаем частоту дискретизации (F_s) и кол-во отсчетов (y):

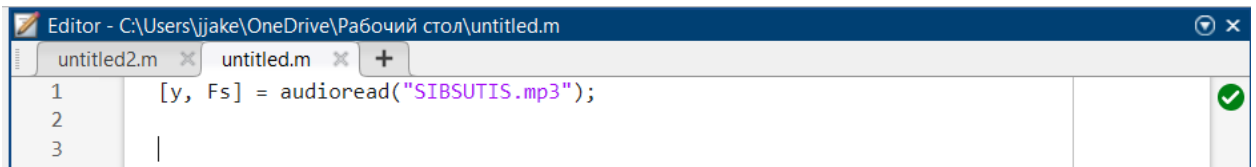
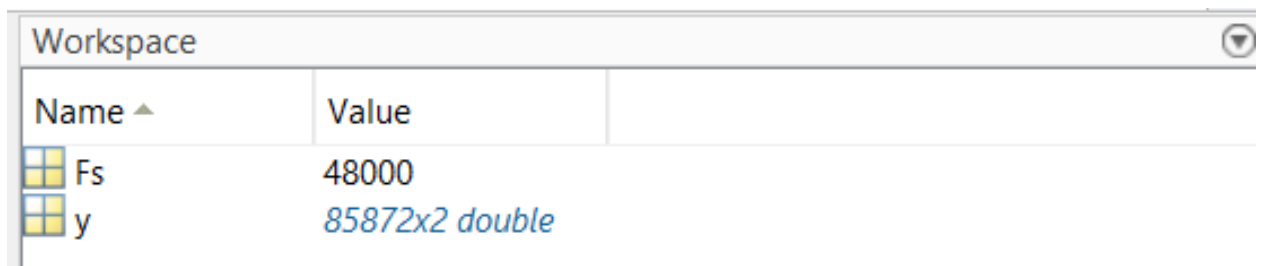


Рисунок 10 — Пример работы с аудиофайлами в MATLAB

Результаты:

The image shows the MATLAB Workspace window. It contains a table with two columns: 'Name' and 'Value'.

Name ▲	Value
Fs	48000
y	85872x2 double

Рисунок 11 — Результаты чтения аудиофайла

Матлаб сделал 85872 отсчета с частотой дискретизации 48000Гц

ЗАДАЧА 10

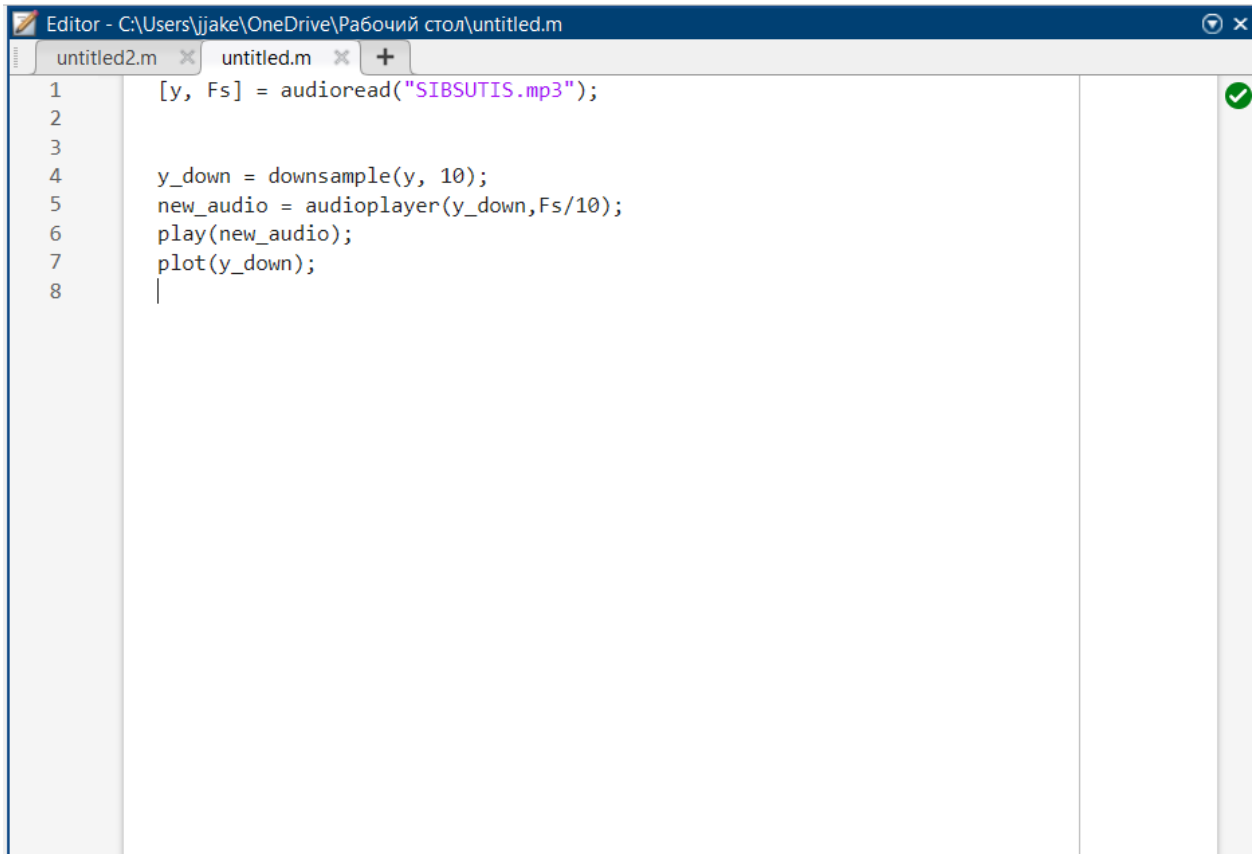
Найдем частоту дискретизации, с которой был записан аудиофайл на носителе. Для этого разделим кол-во элементов после оцифровки (у из прошлого шага) на длительность записи (1.8сек)

$$F_s = \frac{85872}{1.8} = 47707$$

На прошлом шаге мы выяснили, что MATLAB выбрал для файла частоту дискретизации в 48000Гц, что совпадает с частотой дискретизации при записи аудиофайла на носителе.

ЗАДАЧА 11

Уменьшим частоту дискретизации для записи и воспроизведем ее. Для этого уберем каждый 10-ый элемент из отсчетов (y) и уменьшим F_s в 10 раз:

A screenshot of a MATLAB editor window titled "Editor - C:\Users\jjake\OneDrive\Рабочий стол\untitled.m". The window has two tabs: "untitled2.m" and "untitled.m". The code in the editor is as follows:

```
1 [y, Fs] = audioread("SIBSUTIS.mp3");
2
3
4 y_down = downsample(y, 10);
5 new_audio = audioplayer(y_down, Fs/10);
6 play(new_audio);
7 plot(y_down);
8 |
```

A green checkmark icon is visible in the top right corner of the editor window.

Рисунок 12 — Пример искажения записи

Качество записи сильно упало, но все-таки можно разобрать, что было сказано.

График амплитуды сигнала от времени выглядит так:

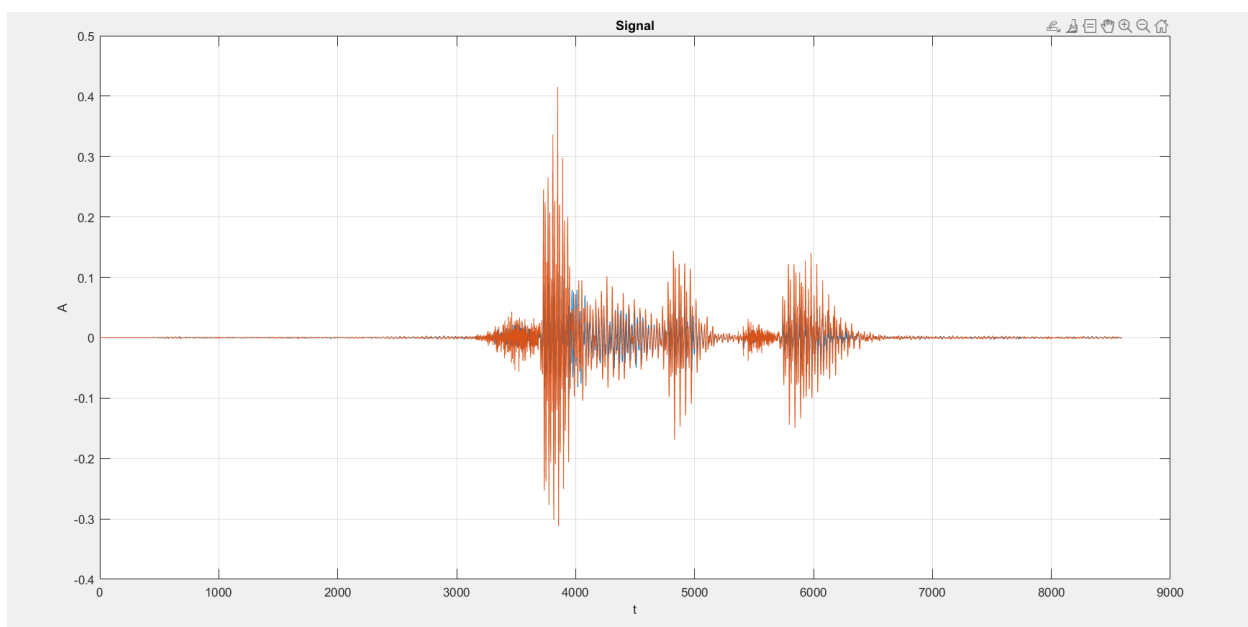


Рисунок 13 — График голоса

ЗАДАЧА 12

Произведем прямое дискретное преобразование Фурье для оригинального и искаженного сигнала. Построим графики амплитудного спектра каждого сигнала:

```
22
23
24     %fft
    y_fft = fft(y);
    y_down_fft = fft(y_down);
25
26
27     %build plots|
    figure(2);
    semilogx(abs(y_fft));
28
29     xlabel('f');
30     ylabel('A');
31     title('Signal');
32     grid on;
33
34
35     figure(3);
36     semilogx(abs(y_down_fft));
37
38     xlabel('f');
39     ylabel('A');
40     title('Signal');
41     grid on;
```

Рисунок 14 — График голоса

Результат

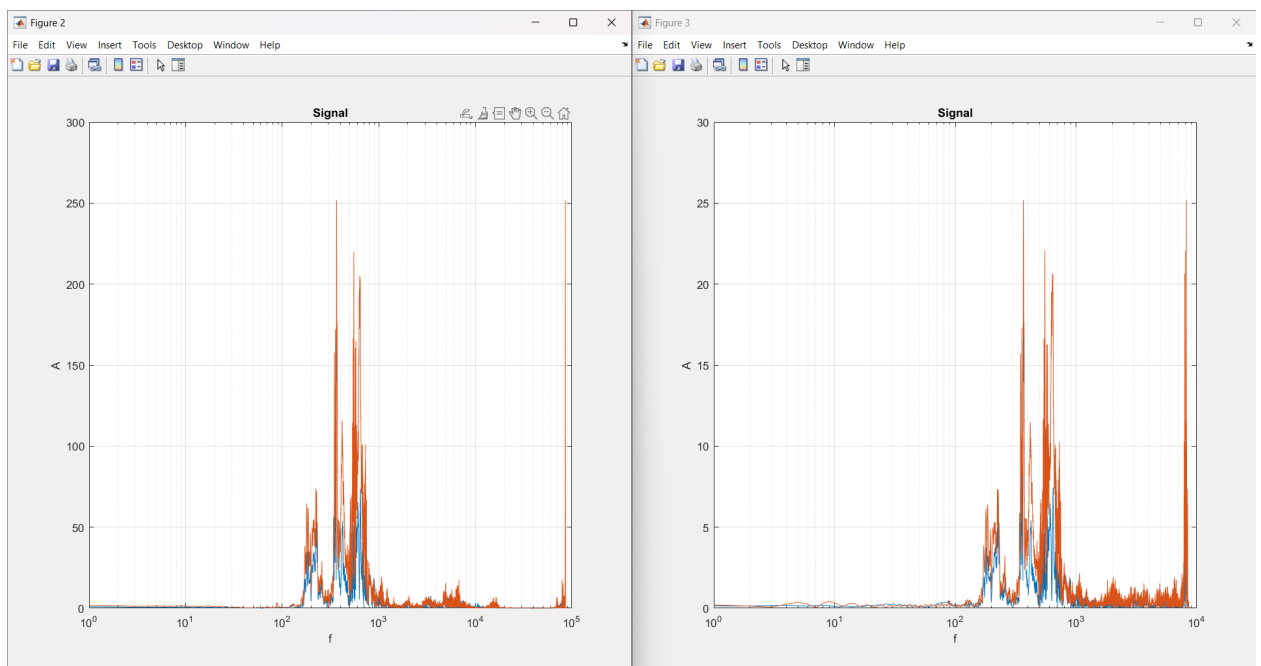


Рисунок 15 — График голоса

Слева амплитудный спектр оригинального сигнала, справа - искаженного.

Ширина амплитудного спектра - ???

ЗАДАЧА 13

Оценим влияние разрядности АЦП на спектр сигнала. Для этого будем семплировать сигнал с разрядностью АЦП из отрезка [3;6], т.е. если разрядность, равна 4, то максимальное значение функции будет $2^4 - 1 = 15$, и если значение функции в точке будет превышать 15, то оно будет округляться до 15. Далее будем создавать Фурье-образ по набору семплов и строить график амплитуды от частоты и наблюдать за тем, как разрядность АЦП будет влиять на частотное представление сигнала.

P.S : В моем случае значения функции очень малы и при разных разрядностях результат будет одинаков, поэтому я увеличил амплитуду функции в 50 раз (домножил на 50 обе компоненты), чтобы можно было увидеть разницу.

Реализация:

```
48 %Test ADC capacity
49 for ADC_capacity = 3:6
50     max_value = 2^ADC_capacity - 1;
51
52     t = 0:1/Fs:1-(1/Fs);
53
54     samples = zeros(1, length(t));
55
56     for k = 1:length(t)
57         tmp = 50 * (sin(12*pi*f*t(k)+pi/11) + sin(10*pi*f*t(k)));
58         if tmp > max_value
59             samples(k) = max_value;
60         else
61             samples(k) = tmp;
62         end
63     end
64
65     F = fft(samples);
66
67     figure;
68     plot(abs(F));
69     xlabel('f');
70     ylabel('A');
71     label = sprintf("Signal ADC_capacity = %d", ADC_capacity);
72     title(label);
73 end
```

Рисунок 16 — График голоса

Результат:

Обратим внимание, что с увеличением разрядности АЦП на графике становится меньше пиков, и он становится больше похож на график, на ко-

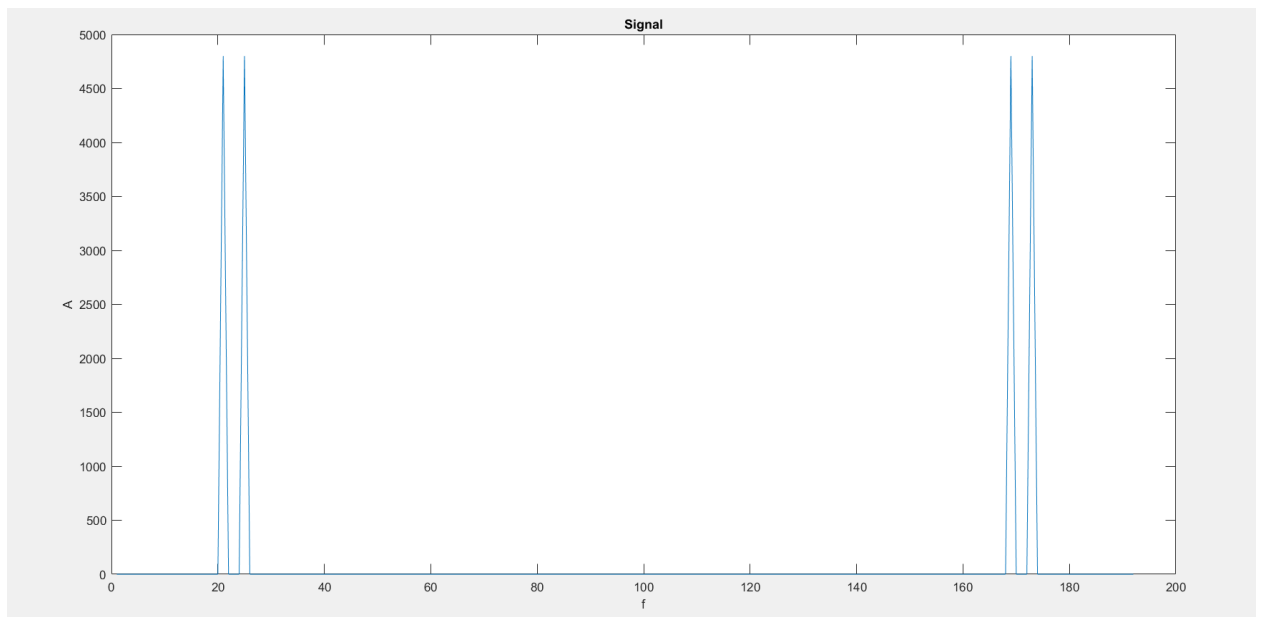


Рисунок 17 — Спектр без ограничений на разрядность

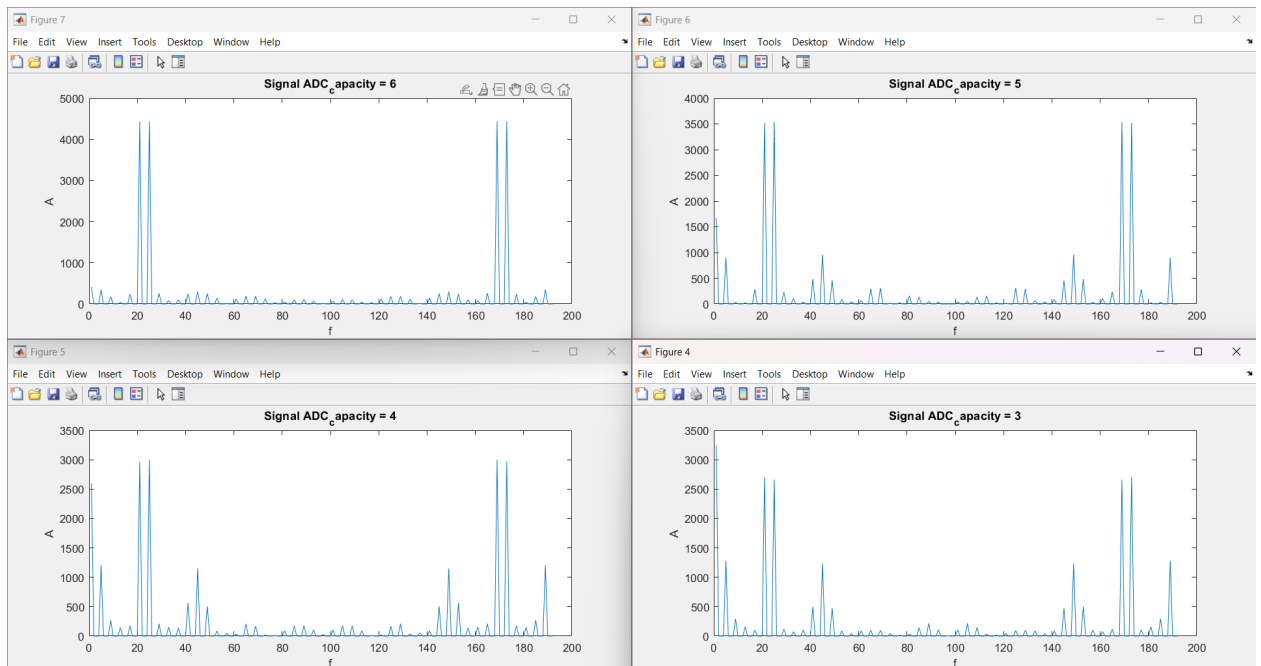


Рисунок 18 — Спектры с ограничениями на разрядность

тором нет ограничений на разрядность АЦП, т.е становится ближе к идеальному.