

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	4
1 ВИЗУАЛИЗАЦИЯ СИГНАЛА.....	6
2 ПОИСК МАКСИМАЛЬНОЙ ЧАСТОТЫ В СПЕКТРЕ СИГНАЛА ..	7
3 ВЫБОР ЧАСТОТЫ ДИСКРЕТИЗАЦИИ	8
4 ОЦИФРОВКА (СЕМПЛИРОВАНИЕ) СИГНАЛА	9
5 ПРЯМОЕ ДИСКРЕТНОЕ ПРЕОБРАЗОВАНИЕ ФУРЬЕ. ВЫЧИСЛЕНИЕ ШИРИНЫ СПЕКТРА СИГНАЛА	10
6 ЦИФРО-АНАЛОГОВОЕ ПРЕОБРАЗОВАНИЕ (ЦАП)	12
7 ВЛИЯНИЕ ЧАСТОТЫ ДИСКРЕТИЗАЦИИ НА ЦАП	13
8 СПЕКТР ГОЛОСА	14
9 ЧТЕНИЕ АУДИОФАЙЛА В МАТЛАВ	15
10 РАССЧЕТ ЧАСТОТЫ ДИСКРЕТИЗАЦИИ, С КОТОРОЙ БЫЛ ЗАПИСАН АУДИОФАЙЛ НА НОСИТЕЛЕ	16
11 ИСКАЖЕНИЕ СИГНАЛА	17
12 СРАВНЕНИЕ СПЕКТРА ОРИГИНАЛЬНОГО И ИСКАЖЕННОГО СИГНАЛА. РАССЧЕТ ШИРИНЫ АМПЛИТУДНОГО СПЕКТРА....	19
13 ВЛИЯНИЕ РАЗРЯДНОСТИ АЦП НА СПЕКТР СИГНАЛА.....	21
14 КОНТРОЛЬНЫЕ ВОПРОСЫ	23
15 ВЫВОД	26
16 ДОПОЛНИТЕЛЬНЫЕ ЗАДАНИЯ	27

17 GITHUB	31
-----------------	----

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.

ВИЗУАЛИЗАЦИЯ СИГНАЛА

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

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

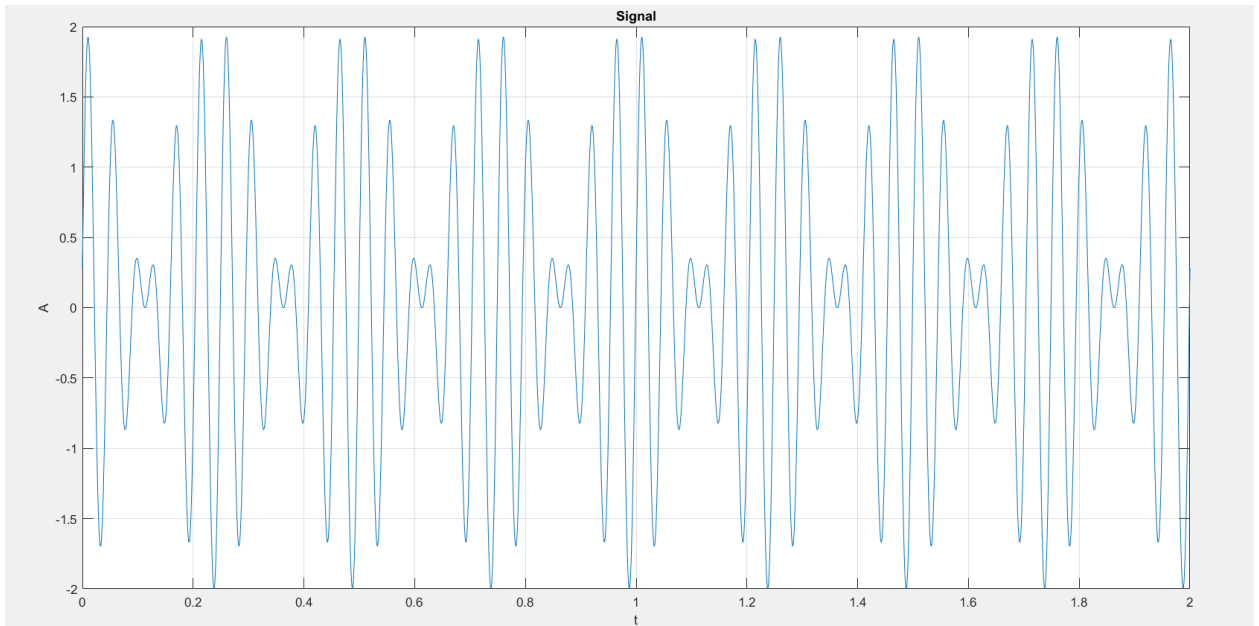


Рисунок 1 — Пример сигнала

ПОИСК МАКСИМАЛЬНОЙ ЧАСТОТЫ В СПЕКТРЕ СИГНАЛА

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:

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

// Найдём 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.

ВЫБОР ЧАСТОТЫ ДИСКРЕТИЗАЦИИ

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:

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

ОЦИФРОВКА (СЕМПЛИРОВАНИЕ) СИГНАЛА

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:

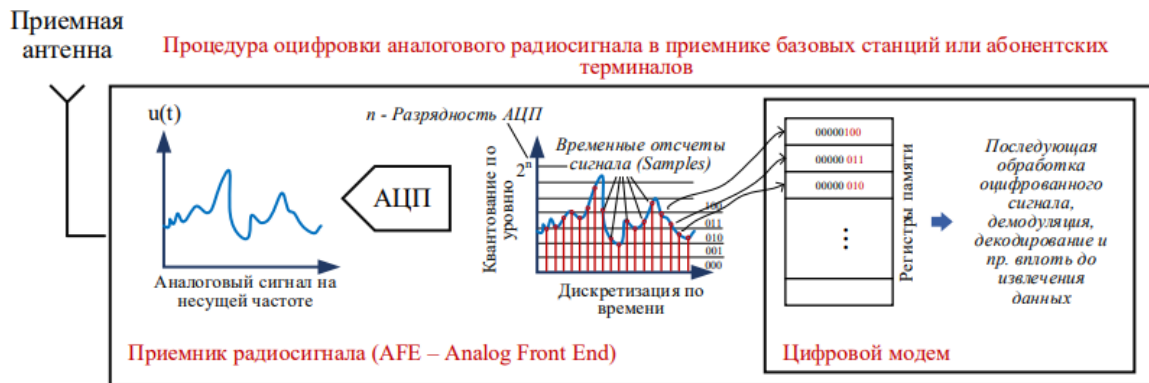


Рис. 2. Аналого-цифровое преобразование радиосигнала.

Рисунок 2 — Пример оцифровки сигнала

Реализация:

```

18 %sampling
19 Fs = 48 * 4;
20 t = 0 : 1/Fs : 1-(1/Fs);
21
22 samples = my_signal(t, f);
23

```

Рисунок 3 — Реализация оцифровки сигнала

Посмотрим, что хранит массив семплов после расчетов:

Variables - samples

samples

1x49 double

	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

ПРЯМОЕ ДИСКРЕТНОЕ ПРЕОБРАЗОВАНИЕ ФУРЬЕ. ВЫЧИСЛЕНИЕ ШИРИНЫ СПЕКТРА СИГНАЛА

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:

	1	2	3	4	5	6	7	8	9	10	
1	1.1990e-14...	-5.8842e-1...	1.3545e-14...	-1.2990e-1...	5.44722e-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.

Визуализируем спектральное представление сигнала

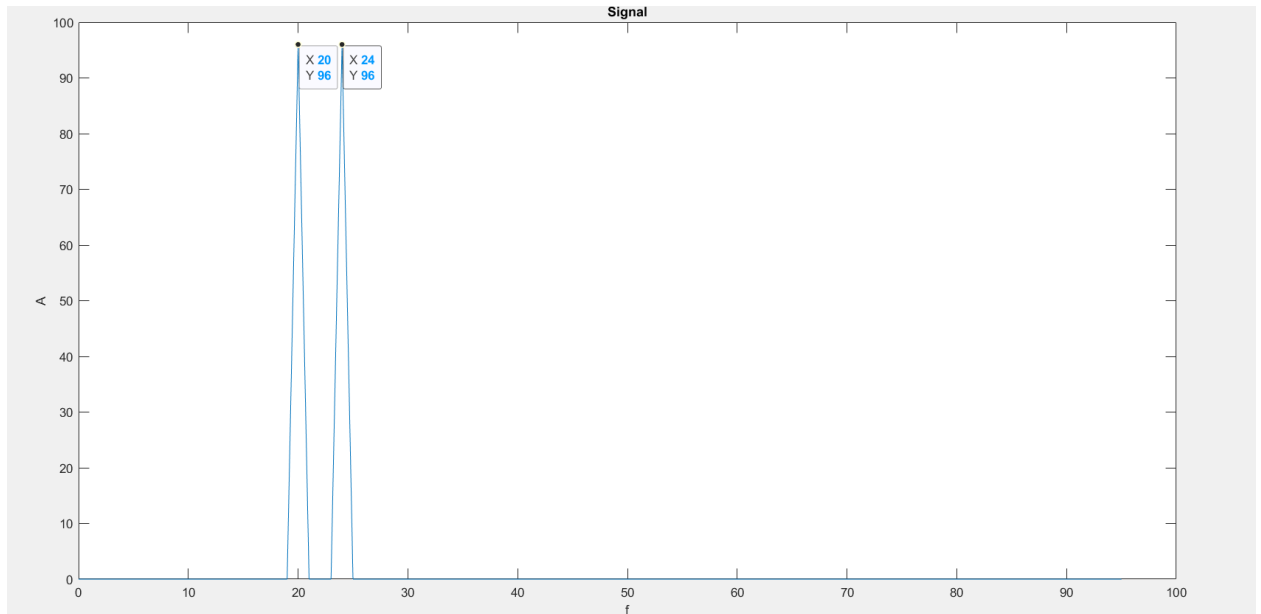


Рисунок 7 — Спектральное представление сигнала

Вычислим ширину спектра сигнала

Ширина спектра сигнала определяется как разность между максимальной и минимальной частотой, присутствующей в спектре сигнала:

$$\text{Ширина спектра} = f_{\max} - f_{\min}$$

В нашем случае сигнал состоит из двух компонент с частотами

$$f_1 = 24 \text{ Гц}, \quad f_2 = 20 \text{ Гц}.$$

Следовательно, ширина спектра сигнала равна

$$\text{Ширина спектра} = 24 - 20 = 4 \text{ Гц}.$$

ЦИФРО-АНАЛОГОВОЕ ПРЕОБРАЗОВАНИЕ (ЦАП)

Восстановим сигнал после оцифровки:

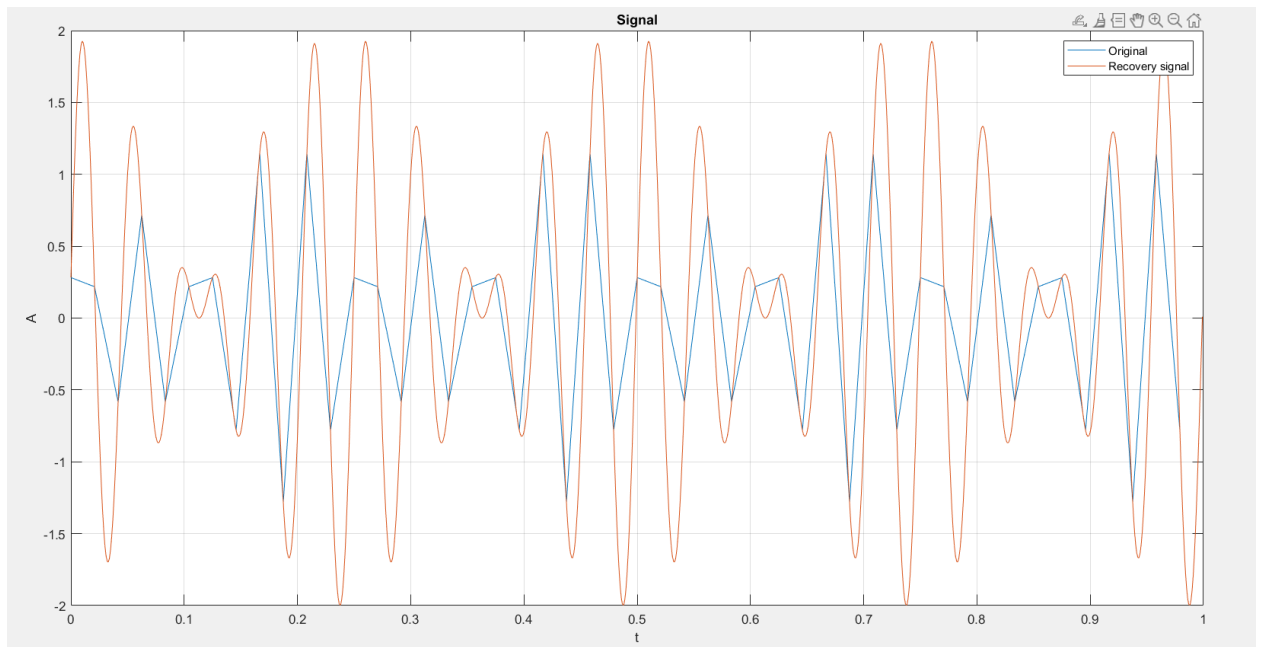


Рисунок 8 — Сигнал после оцифровки ($F_s = 48$)

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

ВЛИЯНИЕ ЧАСТОТЫ ДИСКРЕТИЗАЦИИ НА ЦАП

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:

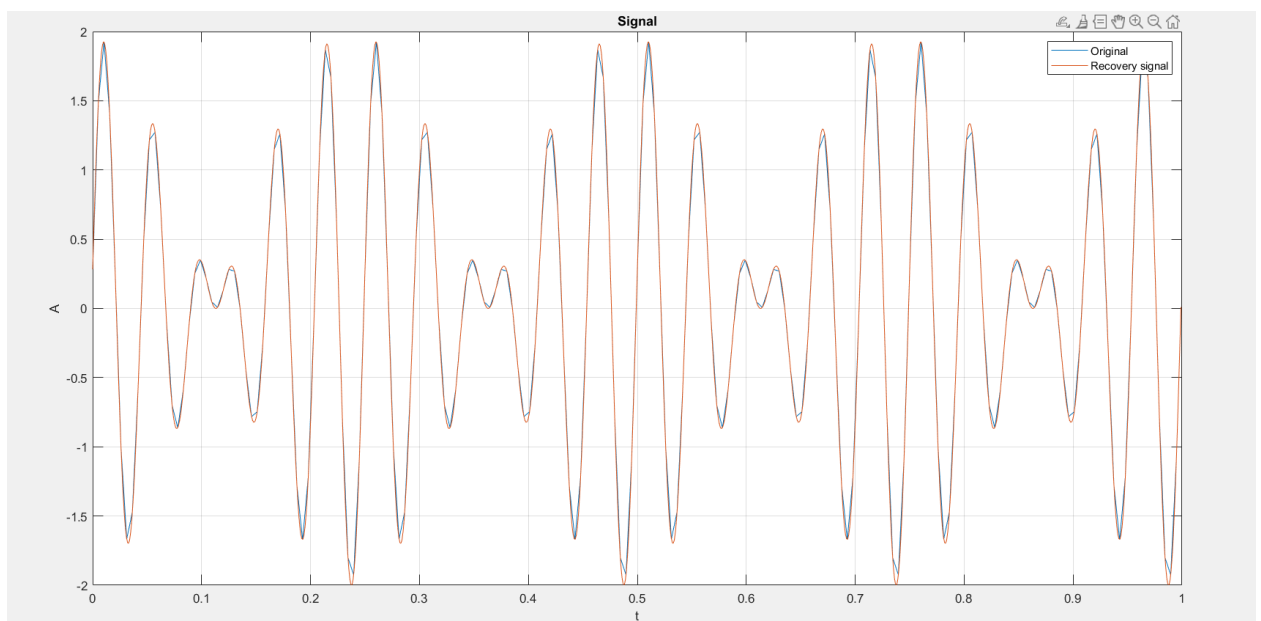


Рисунок 9 — Сигнал после оцифровки ($F_s = 192$)

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

СПЕКТР ГОЛОСА

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

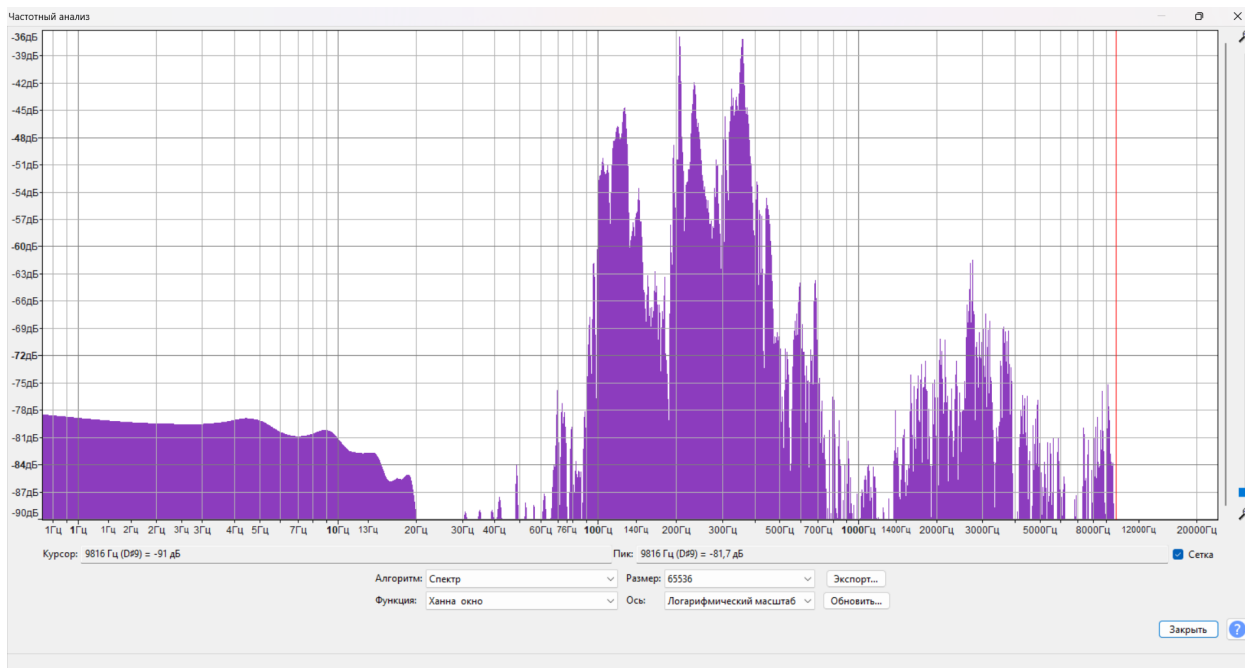


Рисунок 10 — Спектр голоса

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

ЧТЕНИЕ АУДИОФАЙЛА В MATLAB

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

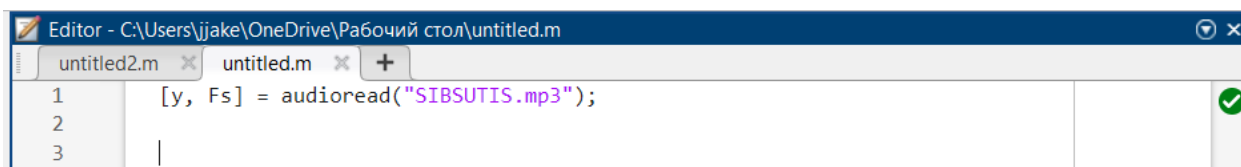
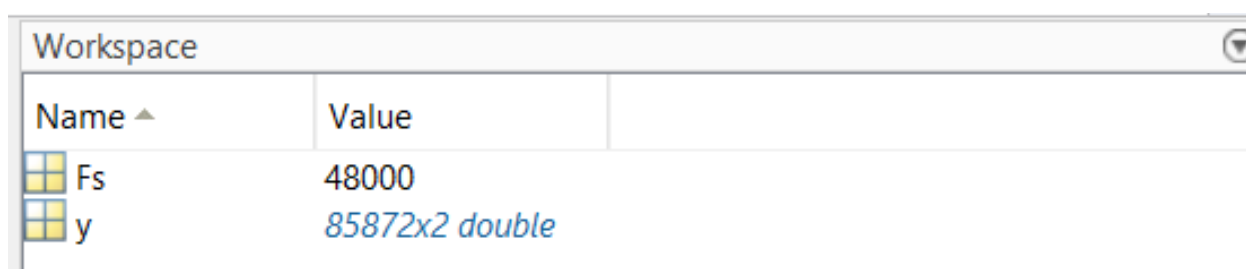


Рисунок 11 — Пример работы с аудиофайлами в MATLAB

Результаты:

The image shows the MATLAB Workspace window. It contains a table with two columns: 'Name' and 'Value'. The variable 'Fs' has a value of '48000'. The variable 'y' has a value of '85872x2 double'.

Name ▲	Value
Fs	48000
y	85872x2 double

Рисунок 12 — Результаты чтения аудиофайла

Матлаб сделал 85872 отсчета с частотой дискретизации 48000Гц

РАССЧЕТ ЧАСТОТЫ ДИСКРЕТИЗАЦИИ, С КОТОРОЙ БЫЛ ЗАПИСАН АУДИОФАЙЛ НА НОСИТЕЛЕ

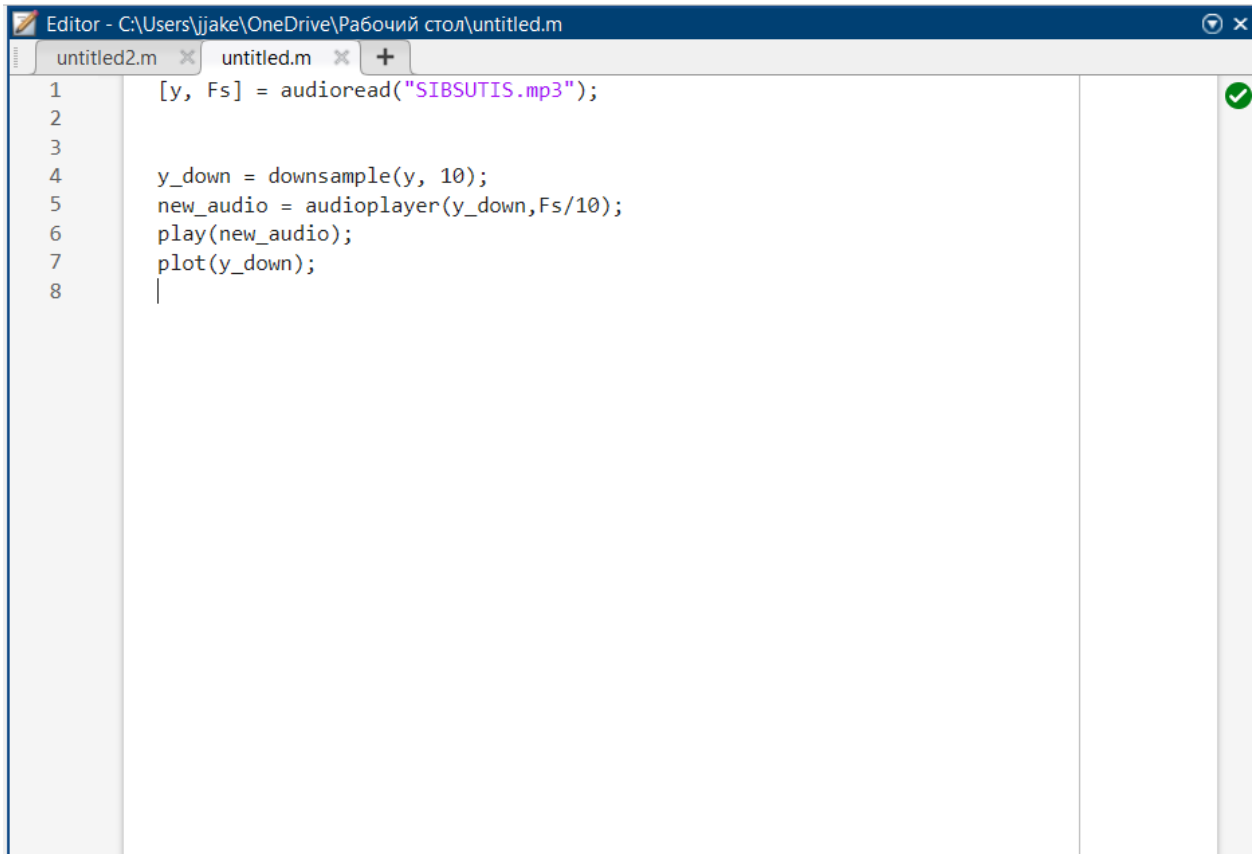
Найдем частоту дискретизации, с которой был записан аудиофайл на носителе. Для этого разделим кол-во элементов после оцифровки (у из прошлого шага) на длительность записи (1.8сек)

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

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

ИСКАЖЕНИЕ СИГНАЛА

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:



```
Editor - C:\Users\jjake\OneDrive\Рабочий стол\untitled.m
untitled2.m x untitled.m x +
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 |
```

Рисунок 13 — Пример искажения записи

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

График амплитуды сигнала от времени выглядит так:

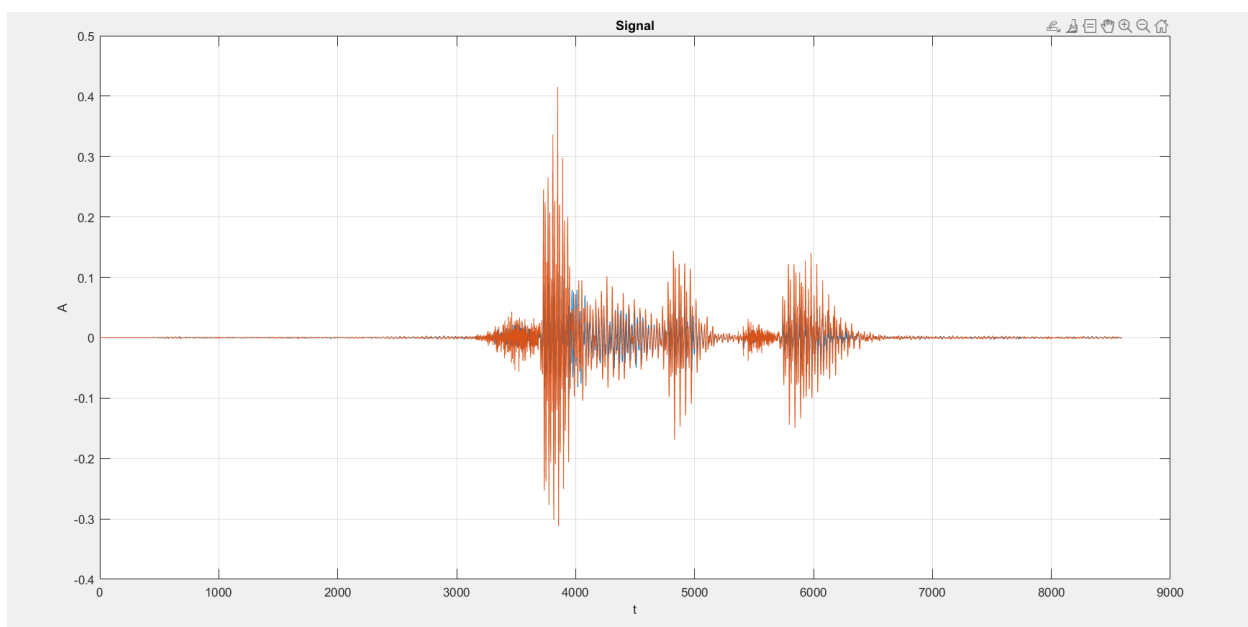


Рисунок 14 — График голоса

СРАВНЕНИЕ СПЕКТРА ОРИГИНАЛЬНОГО И ИСКАЖЕННОГО СИГНАЛА. РАССЧЕТ ШИРИНЫ АМПЛИТУДНОГО СПЕКТРА

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

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

Рисунок 15 — График голоса

Результат

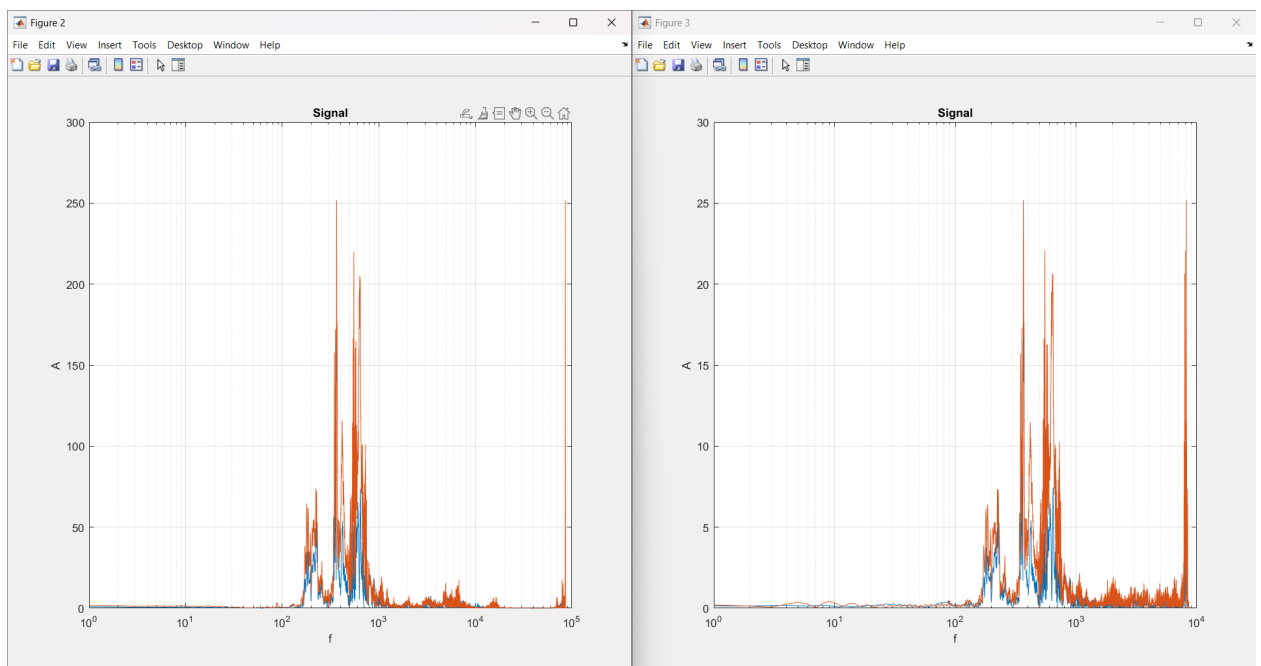


Рисунок 16 — График голоса

Слева амплитудный спектр оригинального сигнала, справа - искаженного.

Ширина амплитудного спектра — это диапазон частот, на которых сигнал имеет значительную амплитуду. Обычно задается какое-то значение, которое считается значимым, но я такого значения вводить не буду и грубо оценю ширину амплитудного спектра.

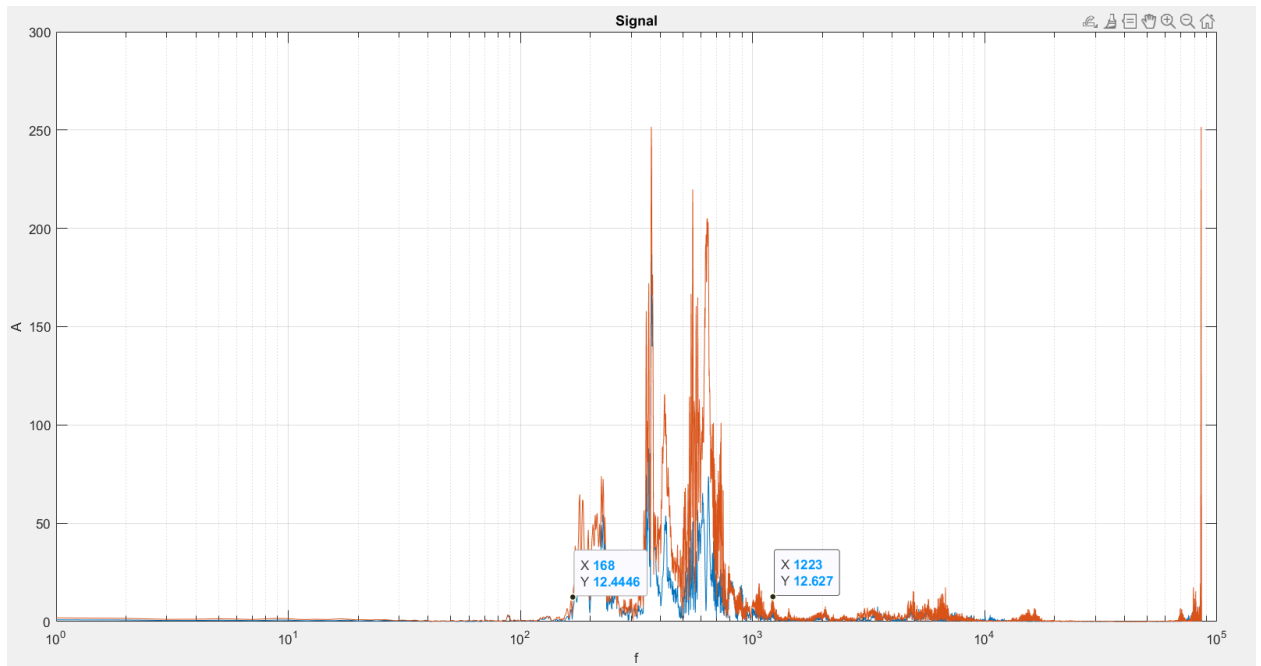


Рисунок 17 — Амплитудный спектр

Из графика выше видим, что ширина амплитудного спектра будет составлять $1223 - 168 = 1055 \text{ Гц}$

ВЛИЯНИЕ РАЗРЯДНОСТИ АЦП НА СПЕКТР СИГНАЛА

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, чтобы можно было увидеть разницу.

Реализация:

```
67 %% Test ADC capacity
68 ADC_start = 3;
69 ADC_end = 6;
70 for ADC_capacity = ADC_start:ADC_end
71
72     %quantum error rate
73     QER = 0;
74
75     sum_signal_value = 0;
76
77     max_value = 2^ADC_capacity - 1;
78
79     samples = zeros(1, length(t));
80
81     for k = 1:length(t)
82
83         signal_value = 40 * my_signal(t(k), f);
84         sum_signal_value = sum_signal_value + abs(signal_value);
85
86         if signal_value > max_value
87             samples(k) = max_value;
88             QER = QER + abs(signal_value - max_value);
89
90         else
91             samples(k) = round(signal_value);
92             QER = QER + abs(signal_value - round(signal_value));
93
94         end
95
96     end
97
98     F = my_fft(samples);
99
100     amps = abs(F);
101
102     N = length(F);
103     f_axis = (0:N/2-1)*(Fs/N);
104     subplot(1, ADC_end - ADC_start + 1, ADC_capacity - ADC_start + 1);
105     stem(f_axis, amps(1:N/2));
106     xlabel('f,Hz')
107     ylabel('A,B')
108     label = sprintf("ADC test, capacity = %d bit, QER = %.3f", ADC_capacity, round(QER/sum_signal_value, 3));
109     title(label)
110
111
112 end
113
```

Рисунок 18 — График голоса

Результат:

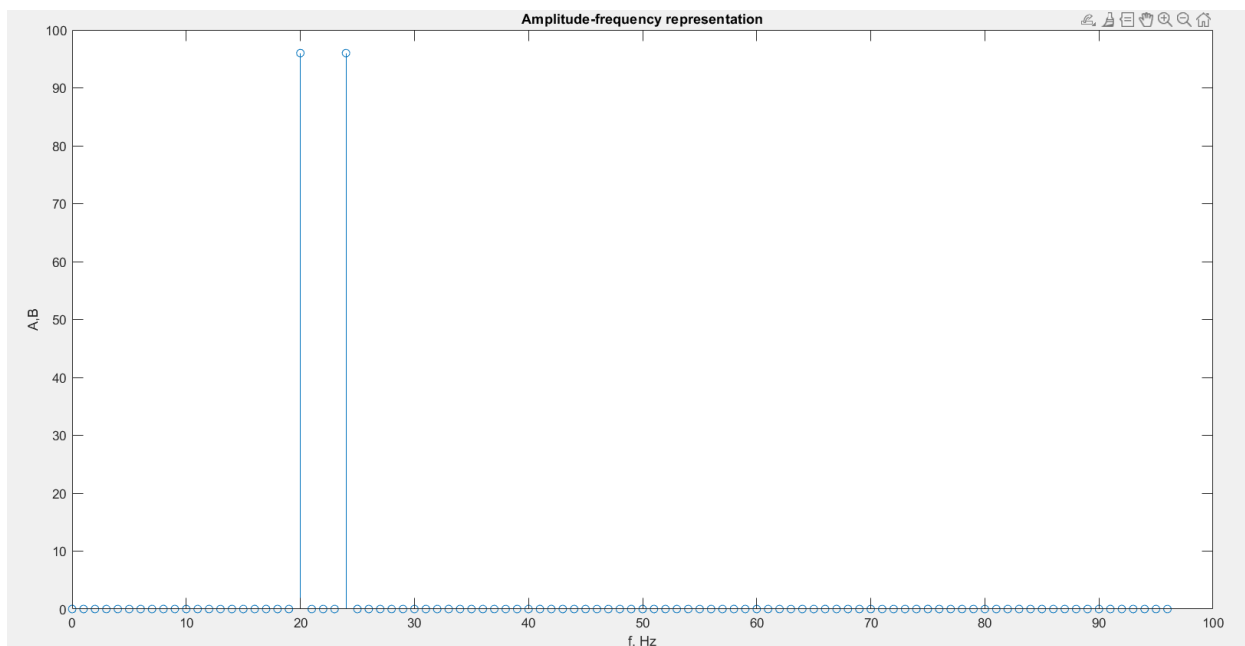


Рисунок 19 — Спектр без ограничений на разрядность

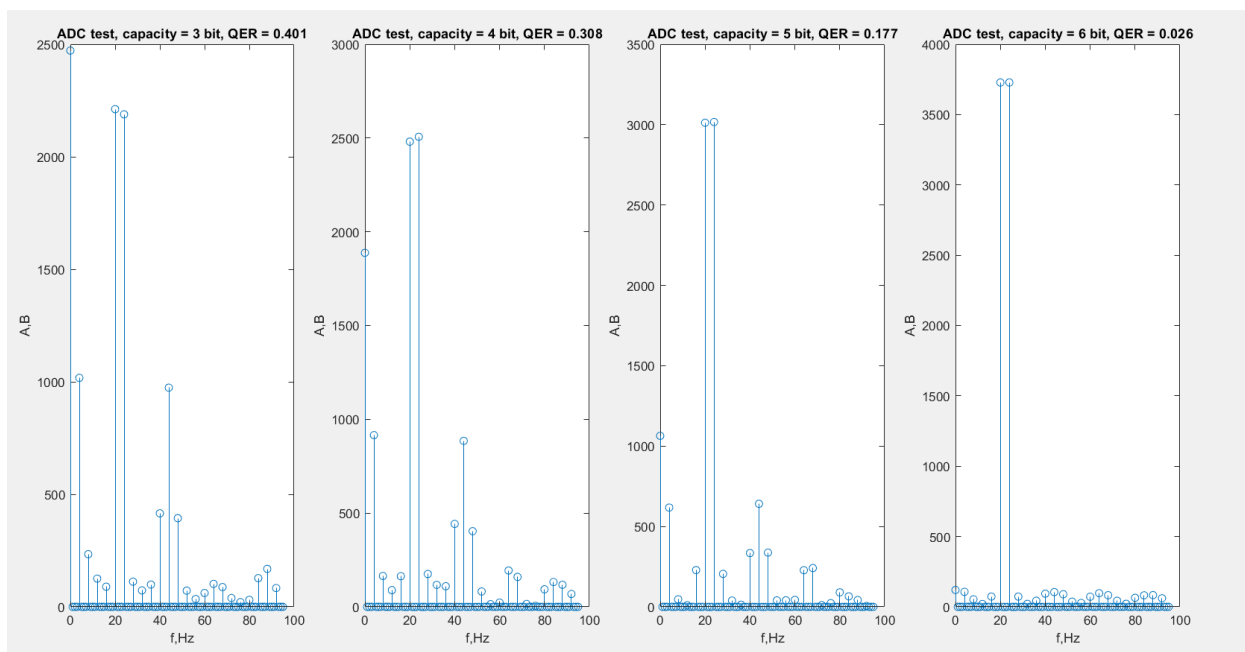


Рисунок 20 — Спектры с ограничениями на разрядность

Обратим внимание, что с увеличением разрядности АЦП на графике становится меньше пиков, и он становится больше похож на график, на котором нет ограничений на разрядность АЦП, т.е. становится ближе к идеальному. На это же указывает параметр QER (Quantization Error Rate), который вычисляется как сумма округлений, деленная на общую сумму значений сигнала.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Для чего используются прямое и обратное преобразование Фурье?
ППФ используется для того, чтобы получить частотное представление сигнала (амплитуды от частоты) по временному представлению сигнала (амплитуды от времени). Частотное представление сигнала позволяет нам узнать, какие частоты входят в спектр сигнала.
ОПФ работает в обратную сторону и даёт нам временное представление сигнала по частотному.
2. Что такое ошибка квантования и дискретизации?
Значения сигнала в определенный момент времени чаще всего являются вещественным числом, а АЦП разделяет значения сигнала на уровни в соответствии со своей разрядностью. Допустим, значение 0.1 В будет соответствовать значению 01, а значение 0.2 В — значению 10. Значения сигнала не всегда попадают в значения уровней АЦП, и отсюда возникает ошибка квантования и дискретизации — разница между значением сигнала и уровнями, установленными в АЦП.
3. Какое количество разрядов АЦП требуется, чтобы оцифровать голос?
4. Как математически получить дискретные отсчёты непрерывного сигнала?
Необходимо воспользоваться дельта-функцией (она же функция Дирака, она же оператор выборки). Это такая функция, которая принимает ненулевое значение только в точке 0, но если мы решим её сдвинуть, то она будет принимать ненулевое значение в другой точке. Интеграл на бесконечности от такой функции равен 1. Если перемножить сигнал на дельта-функцию с определённым смещением и поместить произведение под знак интеграла, то мы получим значение сигнала в конкретной точке.

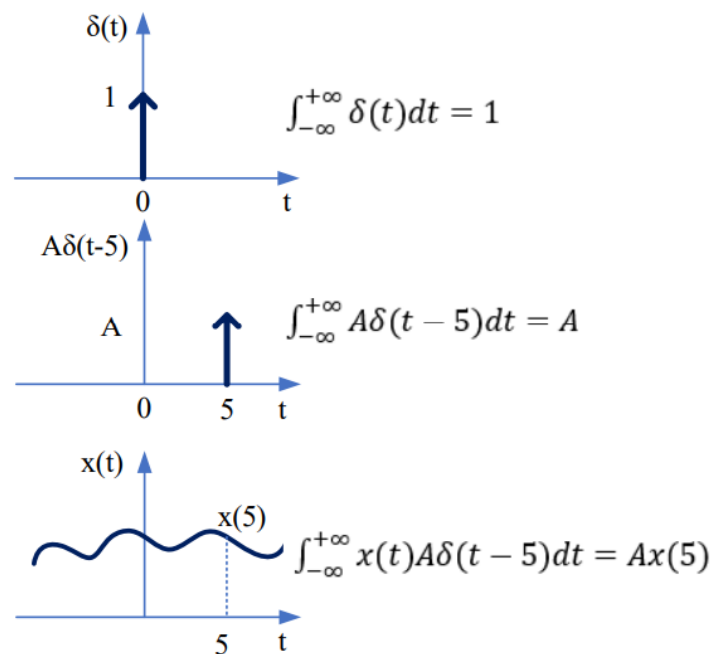


Рис. 3. Дельта-функция.

$$\int_{-\infty}^{+\infty} \delta(t) dt = 1 \quad (1.1)$$

$$\int_{-\infty}^{+\infty} A\delta(t-5) dt = A \quad (1.2)$$

$$\int_{-\infty}^{+\infty} A\delta(t)\delta(t-5) dt = 0 \quad (1.3)$$

Если взять интеграл произведения непрерывной функции $x(t)$ и дельта-функции, например в точке $t=5$, то он будет равен произведению значения функции $x(t)$ в $t=5$ на множитель, стоящий перед дельта-функцией (1.4).

$$\int_{-\infty}^{+\infty} x(t)A\delta(t-5) dt = A \cdot x(5) \quad (1.4)$$

Рисунок 21 — Дельта функция

5. Что такое быстрое преобразование Фурье?

Это более совершенная версия алгоритма, производящего преобразование Фурье. Этот алгоритм работает в разы быстрее обычного преобразования за счёт того, что не делает лишних вычислений, которые часто возникают из-за периодичности функций. Вместо вычислений одного и того же можно обращаться за значениями в кэш.

6. Как определяется минимальная требуемая для оцифровки частота дискретизации сигнала?

Минимальная частота дискретизации сигнала определяется по теореме Котельникова, которая говорит о том, что минимальная частота дискретизации сигнала должна быть минимум вдвое больше, чем

максимальная частота из спектра этого сигнала.

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

7. Какой спектр у периодического сигнала $\sin(10\pi t + \pi/2)$?

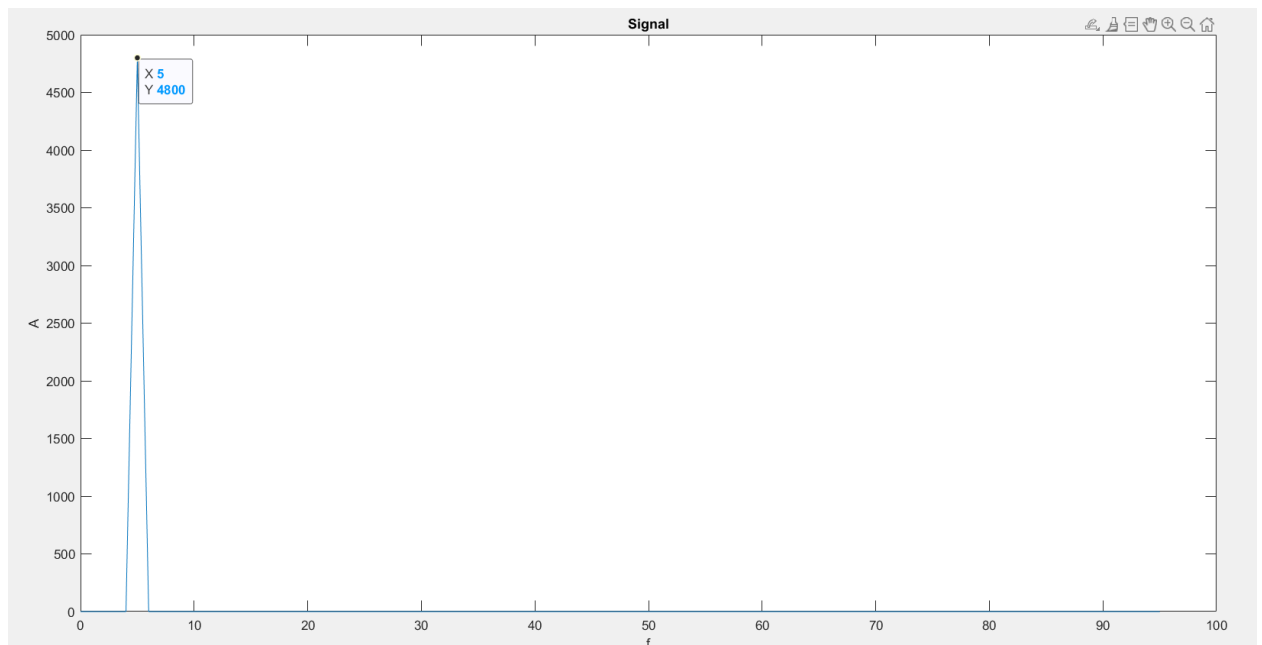


Рисунок 22 — Спектр

ВЫВОД

В ходе проделанной работы я получил представление о формах радиосигналов, их частотном и временном представлении, а также о преобразованиях Фурье и аналоговоцифровых преобразованиях сигналов, частоте дискретизации сигналов. Вспомнил теорему Котельникова и применил ее на практике. Усовершенствовал свои навыки работы с MATLAB

ДОПОЛНИТЕЛЬНЫЕ ЗАДАНИЯ

1. Сравнить собственный алгоритм преобразования Фурье со встроенной функцией MATLAB:

Значения абсолютно идентичны.

2. Сдвинуть свой голос в область высоких частот

Для этого нужно записать свой голос, считать в MATLAB и выбрать величину, на которую будем сдвигать частоту. Сдвиг происходит путем умножения значения сигнала в текущий момент времени на $e^{i(2\pi f_{shift}t)}$. Это свойство ПФ: $Fx(t) * e^{i2\pi f_{shift}t} = X(f - f_{shift})$. Т.е. при умножении на такую экспоненту будет меняться частотное представление сигнала.

Реализация:

3. Сыграть "Имперский марш"

Представим, что у нас есть набор частот и для каждой из них есть время, которое она длится. Можем ли мы воссоздать мелодию? Да, можем. Для этого нужно сформировать массив семплов и потом воспроизвести. Массив семплов создается следующим образом:

1. Выбираем частоту дискретизации (44100 Гц)
2. Берем отрезок $[0; \text{duration}(k) - 1/F_s]$ с шагом $1/F_s$, где $\text{duration}(k)$ - массив длительности частоты.
3. Для каждой точки t из отрезка вычисляем функцию $\sin(2\pi * \text{frequencies}(k) * t)$, где $\text{frequencies}(k)$ - массив частот.

Реализация

Variables - matlab_fft_samples				
matlab_fft_samples ✕				
192x1 complex double				
	1	2	3	
1	-1.2000e+02 + 0.0000e+00i			
2	0.0000 + 0.0000i			
3	0.0000 + 0.0000i			
4	0.0000 + 0.0000i			
5	-96.1568 + 45.9300i			
6	0.0000 + 0.0000i			
7	0.0000 + 0.0000i			
8	0.0000 + 0.0000i			
9	-42.9025 + 31.5360i			

my_ft_samples ✕				
192x1 complex double				
	1	2	3	4
1	-1.2000e+02 + 0.0000e+00i			
2	9.9476e-14 + 8.7930e-14i			
3	-9.9476e-14 + 1.9895e-13i			
4	1.7764e-13 + 3.5705e-13i			
5	-96.1568 - 45.9300i			
6	1.0303e-12 - 1.8030e-13i			
7	6.2528e-13 - 6.6613e-13i			
8	-2.9843e-13 + 6.9988e-13i			
9	-42.9025 - 31.5360i			

Рисунок 23 — Сравнение собственной и встроенной функции преобразования Фурье

```

159 %      N
160 %      X(k) =      sum x(n)*exp(-j*2*pi*(k-1)*(n-1)/N), 1 <= k <= N.
161 %      n=1
162
163 function F = my_fft(samples)
164     N = length(samples);
165     F = zeros(N, 1);
166
167     for k = 1 : N
168         sum = 0;
169         for n = 1 : N
170             sum = sum + samples(n) * exp(-1i * 2 * pi * (k-1) * (n-1)/N);
171         end
172
173         F(k) = sum;
174     end
175
176 end

```

Рисунок 24 — Реализация собственной функции fft

```

42
43 %%change voice
44
45 % get first column from y (mono)
46 y = y(:, 1);
47
48 %timeline
49 t = (0:length(y)-1)' / Fs;
50
51 %shift
52 f_shift = 759;
53 y_shifted = real(y .* exp(1i * 2 * pi * f_shift * t));
54
55 % check voice
56 new_audio = audioplayer(y_shifted, Fs);
57 play(new_audio);
58

```

Рисунок 25 — Сдвиг спектра голоса в область высоких/низких частот

```

90
91     frequencies = [392, 392, 392, 311, 466, 392, 311, 466, 392,...
92     587, 587, 587, 622, 466, 369, 311, 466, 392,...
93     784, 392, 392, 784, 739, 698, 659, 622, 659,...
94     415, 554, 523, 493, 466, 440, 466,...
95     311, 369, 311, 466, 392];
96
97     durations = [
98     350, 350, 350, 250, 100, 350, 250, 100, 700, ...
99     350, 350, 350, 250, 100, 350, 250, 100, 700, ...
100    350, 250, 100, 350, 250, 100, 100, 100, 450, ...
101    150, 350, 250, 100, 100, 100, 450, ...
102    150, 350, 250, 100, 750
103    ];
104
105     my_song = [];
106     fs = 44100;
107
108     for k = 1:length(frequencies)
109         t = 0:1/fs:durations(k)/1000*1.75-1/fs;
110         tone = 1.5 * sin(2*pi*frequencies(k)*t);
111         my_song = [my_song, tone];
112     end
113
114
115     audio = audioplayer(my_song,fs);
116
117     %play
118     play(audio);
119

```

Рисунок 26 — Имперский марш

GITHUB

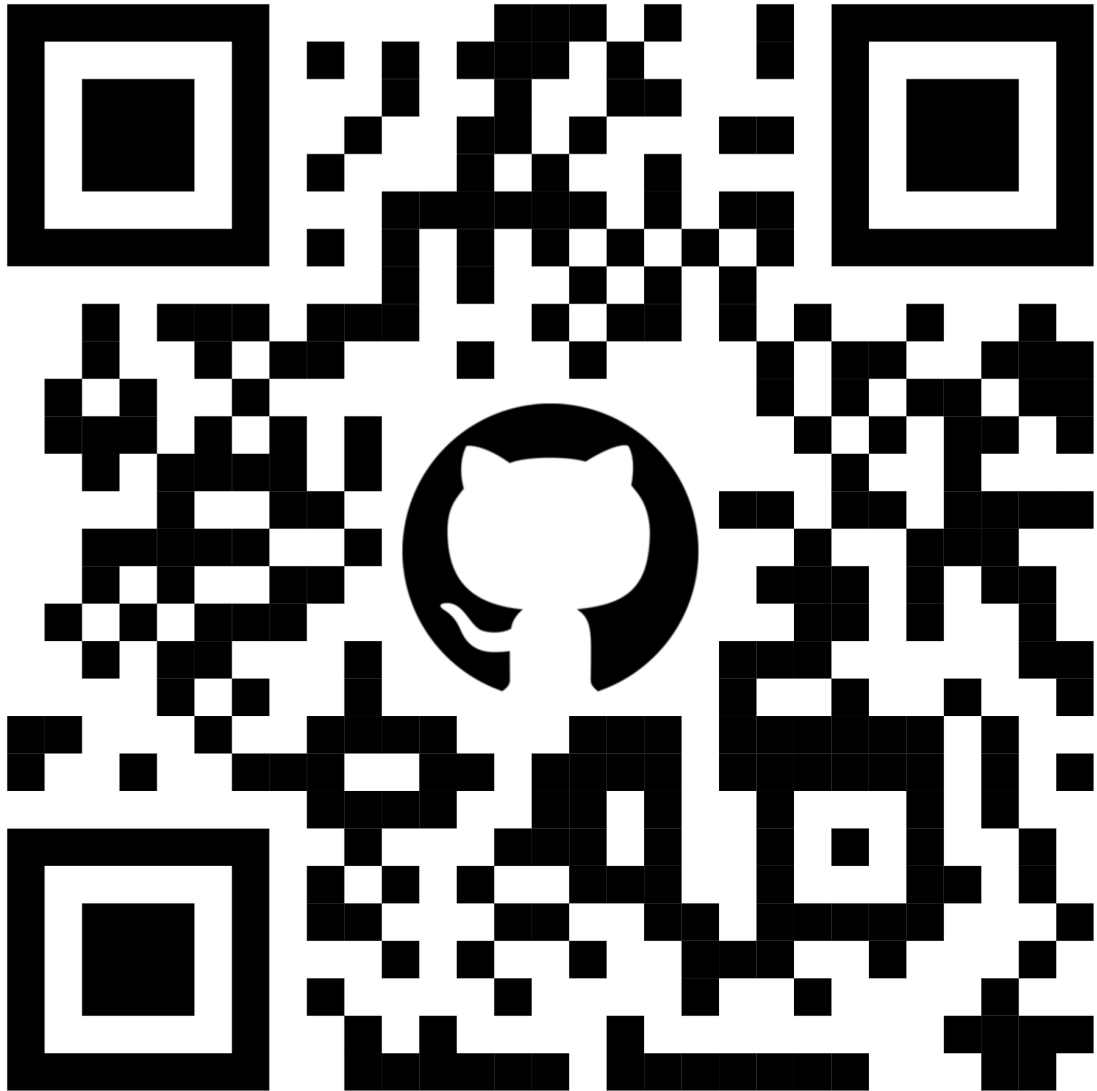


Рисунок 27 — Ссылка на Github репозиторий