

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра ПАО ГИС

ОТЧЁТ
по практической работе
по дисциплине «Архитектура программного обеспечения
гидроакустических систем»

Студент гр. 6381

Спас А.А.

Преподаватель

Пуеров Г.Ю.

Санкт-Петербург

2021

Цель работы.

Целью работы является изучение механизма обработки сигнала с помощью быстрого преобразования Фурье (БПФ) и его программной реализации.

Постановка задачи.

Требуется выполнить обработку сигнала, прочитанного из wav файла, с помощью библиотеки fftw3.3.10.

Сигнал записан в файл `ton_signal_1.wav`.

Выполнение работы.

В выполнения данной практической работы была разработана программа на языке C++.

На первом этапе считываем заголовок wav-файла, в котором хранится такая информация как размер данных в байтах, частота дискретизации и др. Чтение заголовка выполняется функцией *readHeader*.

Далее с помощью функций *calcPortion* и *calcTail* вычисляем количество полных отсчетов (порций) wav-файла и количество оставшихся отсчетов.

Для каждой порции отсчетов происходит чтения данных из wav-файла. Для чтения порции файла определенного размера определена функция *readData*. На каждой итерации чтения данных из файла выполняется БПФ, вычисляются квадраты модулей комплексных чисел и выполняется накопление квадратов модулей.

Для выполнения преобразования Фурье с помощью библиотеки fftw-3.3.10 необходимо сначала определить план, в который нужно записать параметры преобразования: размер БПФ, порцию обрабатываемых данных и флаги `FFTW_ESTIMATE` или `FFTW_MEASURE`. Затем выполнить преобразование с заданным планом и очистить план как указано выше.

Экспоненциальное накопление квадратов модулей выполняется по формуле:

$$acc = \left(1 - \frac{1}{c}\right) acc + \left(\frac{1}{c}\right) Sq,$$

где c - некоторый коэффициент накопления, например $c=4$
 Sq - массив квадратов модулей на текущей итерации цикла.

Ниже представлены результаты выполнения программы при разных размерах БПФ.

На рис.1 представлен график спектра сигнала при $N = 256$. По массиву данных определяется частота сигнала - 890 Гц.

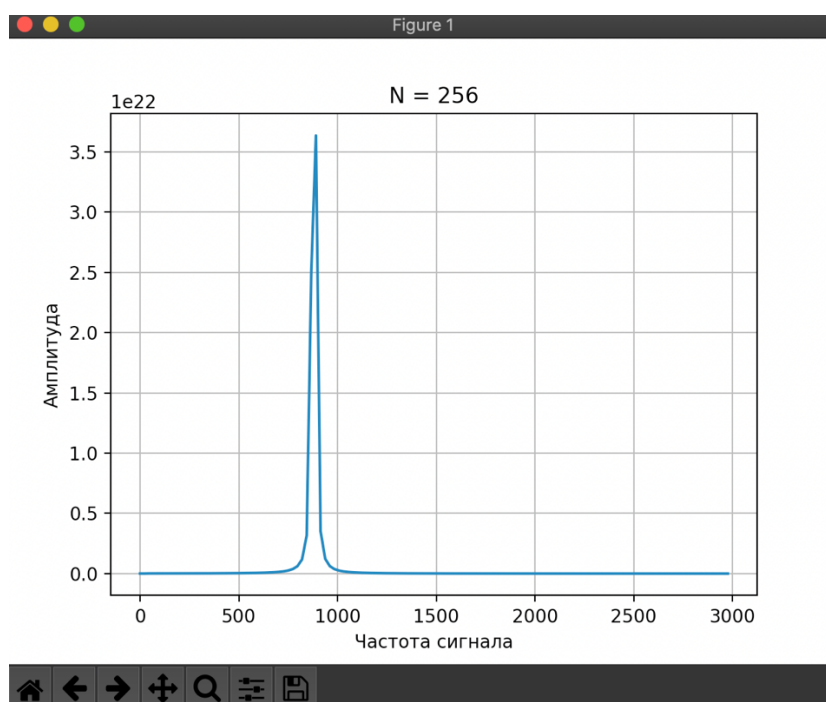


Рисунок 1 – График спектра сигнала при $N = 256$

На рис.2 представлен график спектра сигнала при $N = 512$. По массиву данных определяется частота сигнала - 879 Гц.

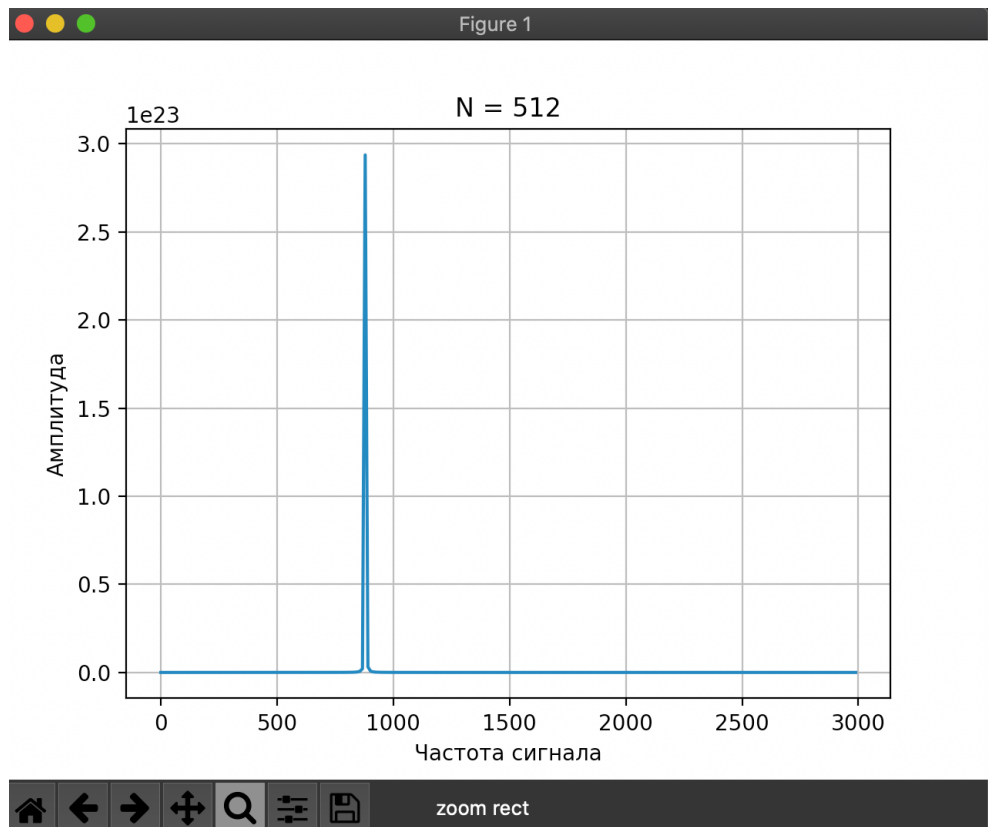


Рисунок 2 – График спектра сигнала при $N = 512$

На рис.3 представлен график спектра сигнала при $N = 1024$. По массиву данных определяется частота сигнала - 879 Гц.

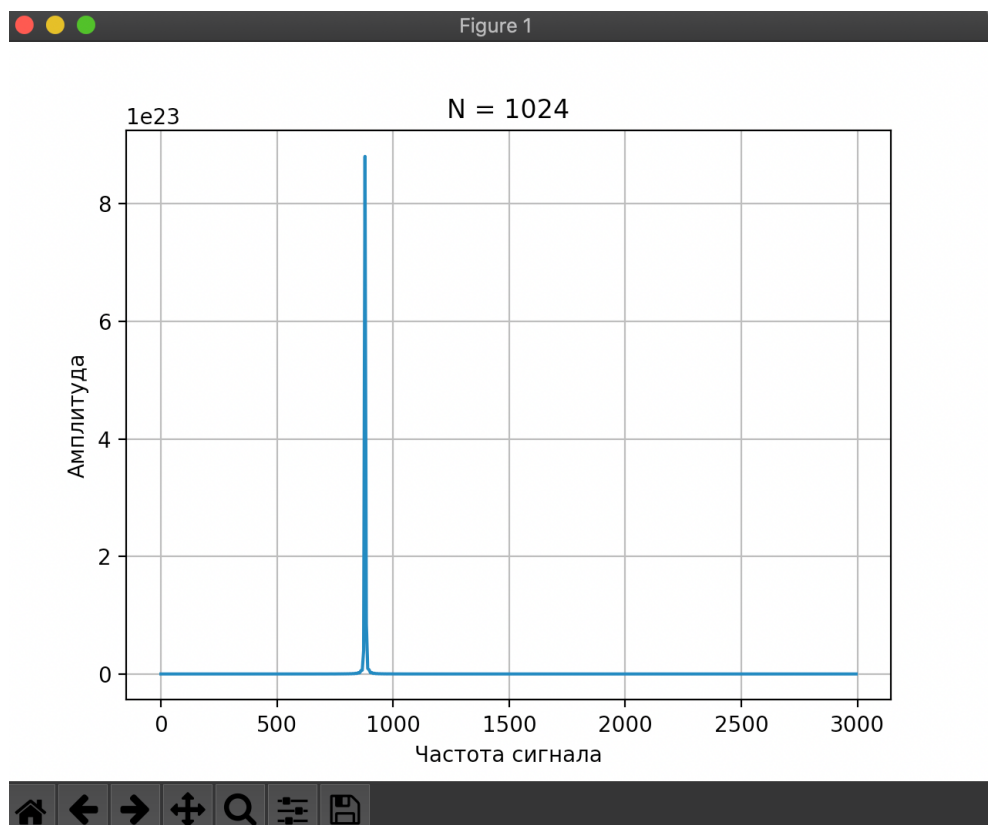


Рисунок 3 – График спектра сигнала при $N = 1024$

При замере производительности функции `fft` с разными планами были получены следующие результаты (см. табл. 1).

Таблица 1 – Результаты измерение производительности функции `fft`

Флаг	FFTW_ESTIMATE	FFTW_MEASURE
Время создание плана, мкс	1485	67501
Выполнение БПФ, мкс	32	14

Выполнение работы.

В результате выполнения данной работы был изучен механизм обработки сигнала с помощью быстрого преобразования Фурье (БПФ) и была выполнена его программная реализация.

Полученные результаты показывают, что время создания плана с использованием флага `FFTW_MEASURE` намного больше, чем `FFTW_ESTIMATE`.

`FFTW_MEASURE` дает команду `FFTW` запустить и измерить время выполнения нескольких FFT, чтобы найти лучший способ вычислить преобразование размера `N`. `FFTW_ESTIMATE`, напротив, не выполняет никаких вычислений, а просто строит разумный план.

Теоретические задачи.

Задача 1. Найдите ДПФ сигнала $x \in \mathbb{C}_N$,

$$x_j = \sin \frac{2\pi j}{N} \cos \frac{2\pi j}{N}, j \in 0:N-1$$

Решение.

$$X(k) = \sum_{j=0}^{N-1} x_j w_N^{kj} = \sum_{j=0}^{N-1} x_j e^{-kj \frac{2\pi i}{N}}:$$

$$X(k) = \sum_{j=0}^{N-1} x_j e^{-ik \frac{2\pi j}{N}} = \sum_{j=0}^{N-1} \sin \frac{2\pi j}{N} \cos \frac{2\pi j}{N} e^{-kj \frac{2\pi i}{N}} =$$

$$= \sum_{j=0}^{N-1} \left(\frac{e^{j \frac{2\pi i}{N}} - e^{-j \frac{2\pi i}{N}}}{2i} \right) \left(\frac{e^{j \frac{2\pi i}{N}} + e^{-j \frac{2\pi i}{N}}}{2} \right) e^{-kj \frac{2\pi i}{N}} =$$

$$= \sum_{j=0}^{N-1} \frac{1}{4i} \left(e^{2i \frac{2\pi j}{N}} - e^{-2i \frac{2\pi j}{N}} \right) e^{-ik \frac{2\pi j}{N}} = \frac{1}{4i} \sum_{j=0}^{N-1} e^{(2-k)j \frac{2\pi i}{N}} - e^{-(2+k)j \frac{2\pi i}{N}} =$$

$$= \frac{1}{4i} \sum_{j=0}^{N-1} w_N^{-(k-2)j} - \frac{1}{4i} \sum_{j=0}^{N-1} w_N^{-(k+2)j}$$

При $k = 2$:

$$w_N^{-(k-2)} = 1$$

Следовательно,

$$\frac{1}{4i} \sum_{j=0}^{N-1} w_N^{-(k-2)j} - \frac{1}{4i} \sum_{j=0}^{N-1} w_N^{-(k+2)j} = \frac{1}{4i} \left(N - \frac{1 - w_N^{-(2+2)N}}{1 - w_N^{-(2+2)}} \right) = \frac{N}{4i}$$

При $k = -2$:

$$w_N^{-(k+2)} = 1$$

Следовательно,

$$\frac{1}{4i} \sum_{j=0}^{N-1} w_N^{-(k-2)j} - \frac{1}{4i} \sum_{j=0}^{N-1} w_N^{-(k+2)j} = \frac{1}{4i} \left(\frac{1 - w_N^{-(-2-2)N}}{1 - w_N^{-(-2-2)}} - N \right) = -\frac{N}{4i}$$

При остальных k :

$$\frac{1}{4i} \sum_{j=0}^{N-1} w_N^{-(k-2)j} - \frac{1}{4i} \sum_{j=0}^{N-1} w_N^{-(k+2)j} = \frac{1}{4i} \left(\frac{1 - w_N^{-(k-2)N}}{1 - w_N^{-(k-2)}} - \frac{1 - w_N^{-(k+2)N}}{1 - w_N^{-(k+2)}} \right) = 0$$

$$X(k) = \begin{cases} k = 2, & \frac{N}{4i} \\ k = -2, & -\frac{N}{4i} \\ k \neq \pm 2, & 0 \end{cases}$$

Задача 2. Найдите взаимосвязь между ДПФ исходного сигнала $x \in \mathbb{C}_N$ и ДПФ преобразованного сигнала $y_l \in \mathbb{C}_N$:

$$y_l(j) = -2i \sin\left(\frac{2\pi lj}{N}\right) x(j), \quad j, l \in \mathbb{C}_N$$

Решение.

$$\begin{aligned} Y_l(k) &= \sum_{j=0}^{N-1} \left(e^{\frac{2\pi ilj}{N}} - e^{\frac{-2\pi ilj}{N}} \right) x(j) w_N^{kj} = \sum_{j=0}^{N-1} (w_N^{-lj} - w_N^{lj}) x(j) w_N^{kj} \\ &= \left(\sum_{j=0}^{N-1} x(j) w_N^{(k-l)j} - \sum_{j=0}^{N-1} x(j) w_N^{(k+l)j} \right) = X(k-l) - X(k+l) \end{aligned}$$

Задача 3. Найдите взаимосвязь между ДПФ исходного сигнала $x \in \mathbb{C}$ и ДПФ преобразованного сигнала $y \in \mathbb{C}$

$$y(j) = \overline{x(N-j)}, \quad j \in 0:N-1$$

Решение.

$$\begin{aligned} Y(k) &= \sum_{j=0}^{N-1} \overline{x(N-j)} w_N^{kj} \\ \overline{Y(k)} &= \overline{\sum_{j=0}^{N-1} \overline{x(N-j)} w_N^{kj}} = [e^{ix} = e^{-ix}] = \sum_{j=0}^{N-1} x(N-j) w_N^{-kj} = \left[\begin{matrix} N-j=p \\ j=N-p \\ p=1 \dots N \end{matrix} \right] = \\ &= \sum_{p=1}^N x(p) w_N^{-k(N-p)} = w_N^{-kN} \sum_{p=1}^N x(p) w_N^{pk} = X(p) \\ Y(k) &= \overline{X(p)} \end{aligned}$$

Задача 5. Пусть p – простое число, $N = p^n$. Докажите, что количество умножений при N -точечном БПФ алгоритме не превосходит $C(p)N \log_p N$, где константа $C(p)$ зависит от p , но не зависит от n . Например, можно взять $C(p) = p^2$.

Решение.

Доказательство по индукции.

Пусть $N = n_1 n_2$. Где n_1, n_2 – любые целые числа.

Для -точечного БПФ алгоритма количество умножений вычисляется по формуле:

$$M(N) = n_1 M(n_2) + n_2 M(n_1) + N$$

Нужно доказать, что $n_1 M(n_2) + n_2 M(n_1) + N \leq p^2 p^n n$.

База индукции.

Пусть $n = 1$

$N = n_1 n_2 = p^n = p \Rightarrow$ (по свойству простых чисел) $n_1 = 1, n_2 = p$,

т.к. p - простое число.

Тогда из неравенства $n_1 M(n_2) + n_2 M(n_1) + N \leq p^2 p^n n$ получаем:

$p + 2 \leq p^2$ – верно при всех возможных простых p .

Индуктивный переход.

Пусть $N = n_1 n_2 = p^n$

Примем неравенство $n_1 M(n_2) + n_2 M(n_1) + p^n \leq p^2 p^n n$ как верное.

Докажем, что при $R = r_1 r_2 = p^{n+1}$ верно и

$$r_1 M(r_2) + r_2 M(r_1) + p^{n+1} \leq p^2 p^{n+1} (n + 1)$$

$$R = r_1 r_2 = p^{n+1} = p^n p \Rightarrow r_1 = n_1, r_2 = n_2 p$$

Можем записать $r_1 M(r_2) + r_2 M(r_1) + p^{n+1} \leq p^2 p^{n+1} (n + 1)$ в виде:

$$n_1 M(n_2 p) + n_2 p M(n_1) + p^n p \leq p^2 p^n p (n + 1)$$

$$n_1 (n_2 M(p) + p M(n_2) + n_2 p) + n_2 p M(n_1) + p^n p \leq p^2 p^n p (n + 1)$$

$$n_1 (n_2 p^2 + p M(n_2) + n_2 p) + n_2 p M(n_1) + p^n p \leq p^2 p^n p (n + 1)$$

$$n_1 (n_2 p + M(n_2) + n_2) + n_2 M(n_1) + p^n \leq p^2 p^n (n + 1)$$

$$n_1 n_2 p + n_1 M(n_2) + n_1 n_2 + n_2 M(n_1) + p^n \leq p^2 p^n n + p^2 p^n$$

$$p^n p + n_1 M(n_2) + p^n + n_2 M(n_1) + p^n \leq p^2 p^n n + p^2 p^n$$

Выразили данное неравенство для $n + 1$ через неравенство для n :

$$(n_1 M(n_2) + n_2 M(n_1) + p^n) + p^n p + p^n \leq p^2 p^n n + p^2 p^n$$

Осталось доказать, что $p^n p + p^n \leq p^2 p^n$

$$p^{n+1} + p^n \leq pp^{n+1}$$

$$p^{n+1} + p^n < p^{n+1} + p^{n+1} = 2p^{n+1}$$

$2p^{n+1} \leq pp^{n+1}$, т.к. p – простое.

Следовательно,

$(n_1M(n_2) + n_2M(n_1) + p^n) + p^np + p^n \leq p^2p^nn + p^2p^n$ – верно.

Что и требовалось доказать.