**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра ПАО ГИС**

ОТЧЁТ

**по практической работе**

**по дисциплине «Архитектура ПО ГАС»**

**Тема: Реализация алгоритма БПФ с помощью библиотеки fftw**

|  |  |  |
| --- | --- | --- |
| Студентка гр. 5381 |  | Петрова Т.А. |
| Преподаватель |  | Пуеров Г.Ю. |

Санкт-Петербург

2020

**Цель работы.**

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

**Постановка задачи.**

Требуется выполнить обработку wav файла, куда записан тональный сигнал, для определения частоты. Преобразование строиться с помощью библиотеки fftw на языке Си в операционной системе Linux. Также необходимо измерить производительность функции с разными планами на личных рабочих машинах.

Сигнал записан в файл ton\_signal\_4.wav.

**Выполнение работы.**

В ходе работы программы выполняется чтение данных из звукового файла. Реализованы две функции: для считывания структуры заголовки файла и функция для чтения из файла порции данных указанного размера. На основе полученных из заголовка данных определяется общее количество отсчетов, количество целых порций по N и остаточное количество дополняется до N нулями. Для каждой порции значений, в цикле, выполняется БПФ, с использованием функций библиотеки fftw. Затем в файл записывается накопление спектра, которое считается в функции по формуле

,  
где некоторый коэффициент накопления, массив квадратов модулей на текущей итерации цикла. Далее полученные данные программой на языке Python выводятся в виде графика.

Реализация БПФ с помощью библиотеки fftw.

|  |
| --- |
| fftw\_plan plan; |
|  |
| plan = fftw\_plan\_dft\_r2c\_1d(N, data\_portion, out, FFTW\_ESTIMATE); |
|  |
| fftw\_execute(plan); |
|  |
| fftw\_destroy\_plan(plan); |

Для выполнения преобразования Фурье с помощью библиотеки необходимо сначала определить план, в который нужно записать параметры преобразования: размер БПФ, порцию обрабатываемых данных и определить флаг: FFTW\_ESTIMATE или FFTW\_MEASURE. Затем выполнить преобразование с заданным планом и очистить план как указано выше.

Результаты выполнения программы при разных размерах БПФ:

Ниже (см. рис.1) представлен график спектра сигнала при N = 4096. По графику видно, что частота сигнала около 1600-1700 Гц.

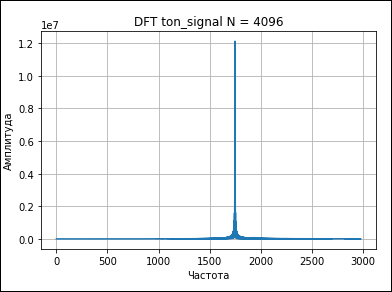


Рисунок 1 – График спектра сигнала при N = 4096

Далее представлен график спектра сигнала при N = 2048. На данном графике частота определяется также – 1600-1700 Гц.

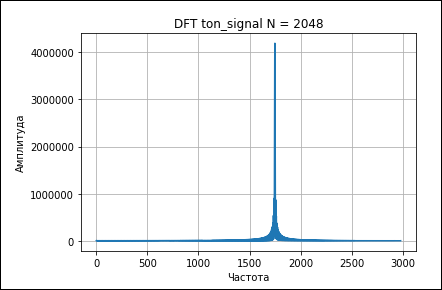


Рисунок 2 – График спектра сигнала при N=2048

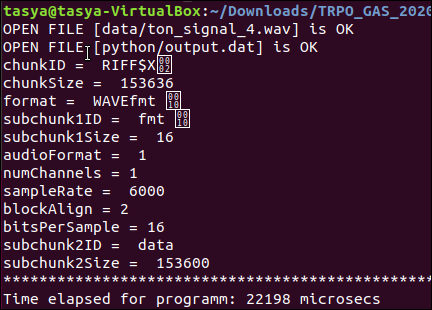


Рисунок 3 – вывод программы.

Для замера производительности функции fft был использован таймер. Для разных планов были получены следующие результаты (см.табл.1).

Таблица 1.

|  |  |  |
| --- | --- | --- |
| K/Флаг | FFTW\_ESTIMATE | FFTW\_MEASURE |
| 1 | 22198 мкс. | 227982 мкс. |
| 2 | 29734 мкс. | 301457 мкс. |
| 3 | 21457 мск. | 214083 мкс. |
| 4 | 26505 мск | 230731 мкс. |

Характеристики виртуальной машины: Ubuntu 18.04 64-bit, 1024 Мб основная память, 1 ЦП.

Полученные результаты показывают, что на данных характеристиках, время создания плана с использованием флага FFTW\_MEASURE намного больше, чем FFTW\_ESTIMATE.

**Вывод**

При выполнении данной работы, был получен навык построения БПФ на языке С. Были оценены временные характеристики различных планов.

Поскольку, FFTW\_MEASURE дает команду FFTW запустить и измерить время выполнения нескольких FFT, чтобы найти лучший способ вычислить преобразование размера N. Этот процесс занимает некоторое время (обычно несколько секунд), в зависимости от вашей машины и размера преобразования. FFTW\_ESTIMATE, напротив, не выполняет никаких вычислений, а просто строит разумный план, который, вероятно, не является оптимальным. То есть если программа выполняет много преобразований одного размера и время инициализации не важно, выгоднее использовать FFTW\_MEASURE.

**Задачи**

Задача 1. Найдите ДПФ сигнала ,

Воспользуемся формулой :

*При :*

Следовательно,

*При :*

Следовательно,

*При остальных*

Задача 2. Найдите взаимосвязь между ДПФ исходного сигнала и ДПФ преобразованного сигнала

Задача 3. Найдите взаимосвязь между ДПФ исходного сигнала и ДПФ преобразованного сигнала

Следовательно, результат:

Задача 4. Используя один из алгоритмов Кули-Тьюки, найдите ДПФ исходного сигнала

Используем алгоритм Кули-Тьюки с прореживанием по времени по основанию 2. Что разделает множество компонент исходного вектора на два подмножества – четные и нечетные индексы.

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

Для n-точечного БПФ алгоритма количество умножений вычисляется по формуле:

Поскольку БПФ приводится к p-точечному БПФ, можно записать формулу в виде:

Докажем по индукции.

База, n=2.

Верно, поскольку p – простое число.

Индуктивный переход. Предположим, что:

Тогда, для n=n+1:

Из индуктивного перехода верно, что:

Значит:

Следовательно, утверждение верно.