МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Ижевский государственный технический университет

имени М.Т. Калашникова»

Кафедра «Программное обеспечение»

**ОТЧЕТ**

по лабораторной работе по дисциплине

«Теория цифровой обработки сигналов»

на тему «Кратковременный анализ сигналов»

Выполнил:

студент гр. Б05-191-1 И.Н. Васильев

Принял:

д.т.н., декан кафедры ПО И.О. Архипов

Ижевск, 2017 г.

**Цель работы:** Овладеть навыками обработки и анализа цифрового сигнала на кратковременных скользящих интервалах.

# Принципы кратковременного анализа сигналов

1.Свойства реального сигнала изменяются во времени.

2. Основа кратковременного анализа сигналов заключается в предположении о медленном изменении параметров сигнала во времени.

3. Сигналы обрабатываются на короткими участками, которые называются интервалы анализа.

4. Процедура повторяется от интервала к интервалу.

5. Интервалы анализа могут следовать примыкая друг к другу, а могут следовать с перекрытием.

6. Результатом обработки на каждом интервале является число (значение энергии или частоты переходов через ноль) или вектор чисел (набор коэффициентов Фурье, т.е. кратковременный спектр).

# Простейшие способы кратковременной обработки сигналов

1. Кратковременная энергия сигнала.

,

где

*N* – длительность интервала анализа;

*n* – начало интервала анализа.

short s;

double E;

for (int i=0; i<K/N; i++) // *K – длина сигнала*

{

sum=0;

for (int k=0; k<N; k++)

{

fread(&s,2,1,inf);

E+=s\*s;

}

s=0.001\*E/N; // *0.001 – масштабирующий множитель*

for (int k=0; k<N; k++)

fwrite(&s,2,1,outf);

}

2. Кратковременная энергия сигнала в логарифмическом масштабе.

.

3. Кратковременная сумма модулей сигнала.

.

4. Кратковременная функция среднего переходов через ноль

,

где

*t* – длительность интервала анализа в секундах;

*mn* – количество переходов через ноль (положительных и отрицательных) на интервале анализа.

5. Нормированный коэффициент корреляции с единичной задержкой



Листинг программы

#include "stdio.h"

#include "math.h"

#include <iostream>

#define LENGTH\_INTERVAL 200 // размер кадра анализа в отсчетах

struct rifftype { //заголовок RIFF файла

char id[4]; //идентификатор файла = "RIFF" = 0x46464952

long len; //длина файла без этого заголовка

} IDRiff;

struct chucktype { //Заголовок куска WAV

char id[4]; //идентификатор = "WAVE" = 0x45564157

char fmt[4]; //идентификатор = "fmt " = 0x20746D66

long len; //длина этого куска WAV - файла,

} IDChuckWave;

struct wavetype { //кусок WAV

short type; //тип звуковых данных

short channels; //число каналов

long SamplesPerSec; //частота выборки

long AvgBytesPerSec; //частота выдачи байтов

short align; //выравнивание

short bits; //число бит на выборку

} IDWave;

struct sampletype { //идентификатор выборки

char id[4]; //идентификатор ="data" =0x61746164

long len; //длина выборки ( кратно 2 )

} IDSampleWave;

FILE \*inf, \*outf;

void task1() { // кратковременная энергия сигнала

int N = (IDSampleWave.len / 2) / LENGTH\_INTERVAL;

short s;

double sum;

for (int i = 0; i<N; i++)

{

sum = 0;

for (int k = 0; k<LENGTH\_INTERVAL; k++)

{

fread(&s, 2, 1, inf);

sum += s\*s;

}

s = 3000 \* log10(0.01 + sum / LENGTH\_INTERVAL);

for (int k = 0; k<LENGTH\_INTERVAL; k++)

fwrite(&s, 2, 1, outf);

}

}

void task2() { // кратковременная энергия сигнала в логарифмической форме

int N = (IDSampleWave.len / 2) / LENGTH\_INTERVAL;

short s;

double sum;

for (int i = 0; i<N; i++)

{

sum = 0;

for (int k = 0; k<LENGTH\_INTERVAL; k++)

{

fread(&s, 2, 1, inf);

sum += s\*s;

}

s = 3000 \* log10(0.01 + sum / LENGTH\_INTERVAL);

for (int k = 0; k<LENGTH\_INTERVAL; k++)

fwrite(&s, 2, 1, outf);

}

}

//Кратковременная сумма модулей сигнала//

void task3() {

int N = (IDSampleWave.len / 2) / LENGTH\_INTERVAL;

short s;

double E = 0;

int N = 100;

for (int i = 0; i < N; i++) // K – длина сигнала

{

for (int k = 0; k < LENGTH\_INTERVAL; k++)

{

fread(&s, 2, 1, inf);

E += abs(s);

}

s = E / N; // 0.001 – масштабирующий множитель

for (int k = 0; k < LENGTH\_INTERVAL; k++)

fwrite(&s, 2, 1, outf);

E = 0;

}

}

//Кратковременная функция среднего переходов через ноль//

void task4() {

int N = (IDSampleWave.len / 2) / LENGTH\_INTERVAL;

short s\_1, s\_2;

int m = 0;

double temp;

for (int i = 0; i < LENGTH\_INTERVAL; i++) // K – длина сигнала

{

fread(&s\_1, 2, 1, inf);

for (int k = 1; k < LENGTH\_INTERVAL; k++)

{

fread(&s\_2, 2, 1, inf);

if (s\_1 > 0 && s\_2 < 0 || s\_1 < 0 && s\_2>0) ++m;

s\_1 = s\_2;

}

temp = LENGTH\_INTERVAL / 11.025;

s\_1 = m \* 5000 / (2 \* (N / 11.025)); // 0.001 – масштабирующий множитель

m = 0;

for (int k = 0; k < LENGTH\_INTERVAL; k++)

fwrite(&s\_1, 2, 1, outf);

}

}

//Нормированный коэффициент корреляции с единичной задержкой//

void task5() {

short s\_1, s\_2;

int N = (IDSampleWave.len / 2) / LENGTH\_INTERVAL;

double sum = 0;

double E = 0;

for (int i = 0; i < N; i++) // K – длина сигнала

{

fread(&s\_1, 2, 1, inf);

for (int k = 1; k < LENGTH\_INTERVAL; k++)

{

fread(&s\_2, 2, 1, inf);

E += s\_1\*s\_1;

sum += s\_1\*s\_2;

s\_1 = s\_2;

}

s\_2 = sum / E \* 1000;

sum = 0;

E = 0;

for (int k = 0; k < LENGTH\_INTERVAL; k++)

fwrite(&s\_2, 2, 1, outf);

}

}

int main()

{

inf = fopen("C:\\Users\\Илья\\Desktop\\Лаб2\\newSK.wav", "rb");

outf = fopen("C:\\Users\\Илья\\Desktop\\Лаб2\\newSK2.wav", "wb");

if (inf == 0)

{

return 0;

}

fread(&IDRiff, sizeof(IDRiff), 1, inf);

fread(&IDChuckWave, sizeof(IDChuckWave), 1, inf);

fread(&IDWave, sizeof(IDWave), 1, inf);

fread(&IDSampleWave, sizeof(IDSampleWave), 1, inf);

IDRiff.len = 236;

IDSampleWave.len = 200;

fwrite(&IDRiff, sizeof(IDRiff), 1, outf);

fwrite(&IDChuckWave, sizeof(IDChuckWave), 1, outf);

fwrite(&IDWave, sizeof(IDWave), 1, outf);

fwrite(&IDSampleWave, sizeof(IDSampleWave), 1, outf);

task1();

task2();

return 0;

}

РЕЗУЛЬТАТЫ РАБОТЫ

На рис.1 показана кратковременная энергия сигнала.

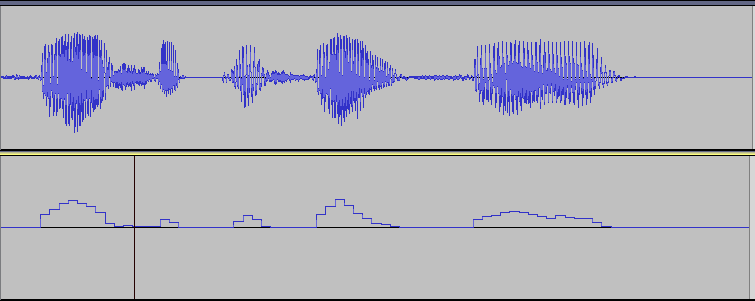


Рис.1

На рис.2 показана кратковременная энергия сигнала в логарифмическом масштабе.

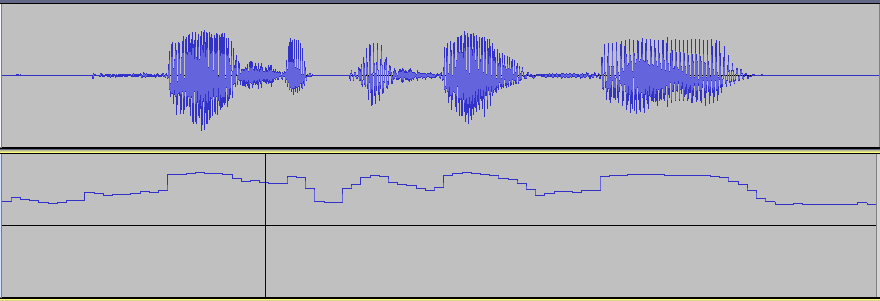


Рис.2

На рис.3 показана кратковременная сумма модулей сигнала.

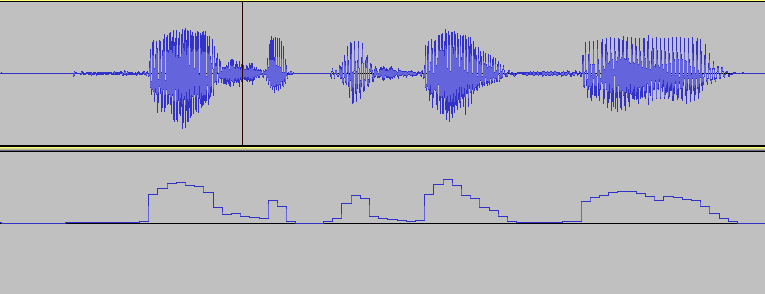


Рис.3

На рис.4 показана кратковременная функция среднего переходов через ноль.

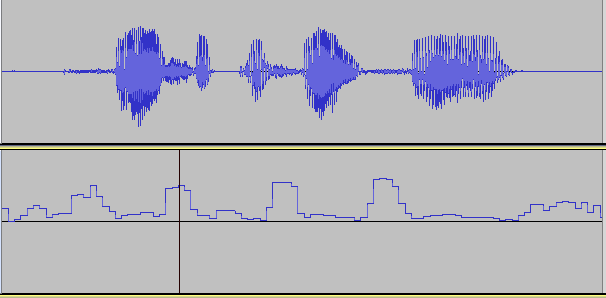


Рис.4

На рис.5 показан нормированный коэффициент корреляции с единичной задержкой.

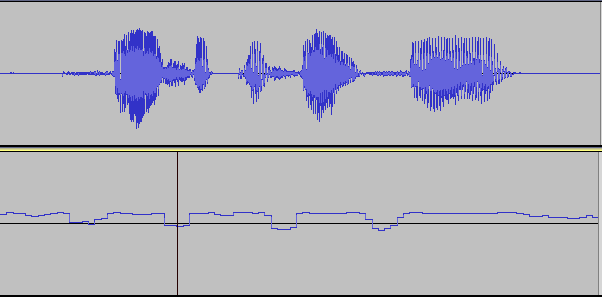


Рис.5

ВЫВОД

В ходе выполнения данной лабораторной работы были получены базовые знания по теории цифровой обработки сигналов и навыки их предварительной обработки. Полученные результаты соответствуют ожидаемым результатам.