МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение

высшего профессионального образования

«Ижевский государственный технический университет имени М.Т. Калашникова»

Факультет «Информатика и вычислительная техника»

Кафедра «Программное обеспечение»

Отчет по лабораторной работе №4

по дисциплине «Теория цифровой обработки сигналов»

на тему «Цифровая обработка речевых сигналов»

Выполнил:

студент группы Б18-191-2 Р.А. Гумметов

Принял: И.О.Архипов

Ижевск 2020

ЦЕЛЬ РАБОТЫ

Целью работы является изучение методов спектрального анализа речевых сигналов.

ЗАДАНИЕ

Проанализируйте полученные графики фрагментов сигнала и их спектров. Зная, что шипящие согласные содержат большое количество высокочастотного шума, а энергия гласных звуков преимущественно сосредоточена в области нижних частот, ответьте на следующие вопросы:

1. Найдите фрагменты сигнала, которые соответствуют шипящим звукам. Напишите код, который вырежет из сигнала указанный Вами фрагмент, вычислит его спектр и отобразит только данную пару графиков.
2. Аналогичным образом поступите с фрагментом гласного звука.
3. В отчете покажите Ваш код и выводимые на экран графики.
4. Проанализировав формы спектров гласного звука и шипящего звуков. Найдите отличия. Предложите алгоритм (можно словесный)классификации звуков на гласные и шипящие согласные.

КОД ПРОГРАММЫ

import wave

importnumpy as np

fromscipy.fftpack import fft

%matplotlib inline

importmatplotlib.pyplot as plt

wav = wave.open("Favorite\_station\_11.wav", mode="r")

(nchannels, sampwidth, framerate, nframes, comptype, compname) = wav.getparams()

content = wav.readframes(nframes)

types = {

1: np.int8,

2: np.int16,

4: np.int32

}

samples = np.fromstring(content, dtype=types[sampwidth])

N=1024 #Размер БПФ

b1=24000

b2=12800

win=np.hanning(N) # Сгенерируем временное окно

K=int(nframes/N) #Количество интервалов анализа

ninterval=np.arange(0,K) #номера интервалов анализа

k = np.arange(N)

T = float(N)/framerate

frq = k/T # Полный диапазон (период) спектра

frq = frq[0:int(N/2)] #Основная полоса частот (по Котельникову)

x1=np.arange(b1,b1+N) #Номера отсчетов сигнала в текущем интервале анализа

y1=win\*samples[b1:b1+N] #Взвешанная копия интервала анализа

x2=np.arange(b2,b2+N) #Номера отсчетов сигнала в текущем интервале анализа

y2=win\*samples[b2:b2+N] #Взвешанная копия интервала анализа

Y1=abs(fft(y1)/N) #Вычисление БПФ и нормализация

Y1log=20\*np.log10(Y1+0.000001)

Y2=abs(fft(y2)/N) #Вычисление БПФ и нормализация

Y2log=20\*np.log10(Y2+0.000001)

f, ax = plt.subplots(2, 2,figsize=(15, 10))

ax[0,0].plot(x, samples[b1:b1+N])

ax[0,1].plot(frq, Y1log[0:int(N/2)],'r')

ax[1,0].plot(x, samples[b2:b2+N])

ax[1,1].plot(frq, Y2log[0:int(N/2)],'r')

РЕЗУЛЬТАТ РАБОТЫ ПРОГРАММЫ

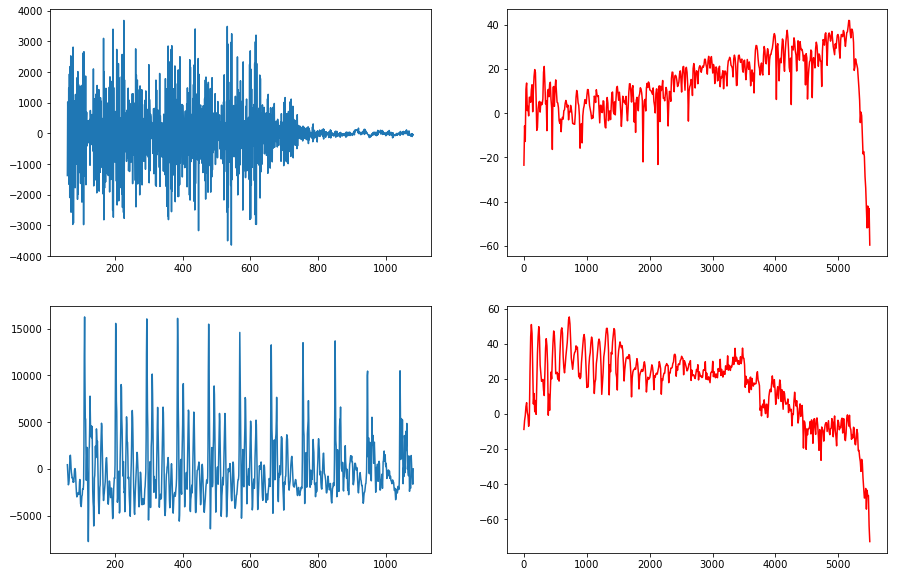


Рис.1

АЛГОРИТМ КЛАССИФИКАЦИИ ЗВУКОВ НА ГЛАСНЫЕ И ШИПЯЩИЕ СОГЛАСНЫЕ

1. Разделить спектр пополам
2. Посчитать энергию N1 и N2 в полученных частях спектра
3. Найти отношение N2/N1
4. Если N2/N1> 1, то звук шипящий, если N2/N1< 1, то звук гласный