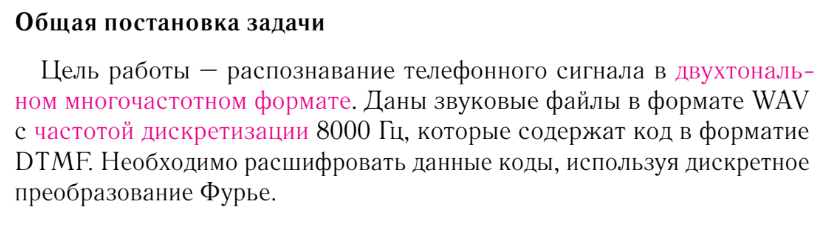
**Лабораторная работа №2**

**«Дискретное преобразование Фурье в приложениях»**

Чернякова Алексея, студента 3 курса 6б группы

Вариант 2. Распознавание звукового сигнала



2. Ход выполнения

Графики звуковых волн для каждого файла

Это и последующие задания выполнены средствами пакета MATLAB:

filename='s3.wav';

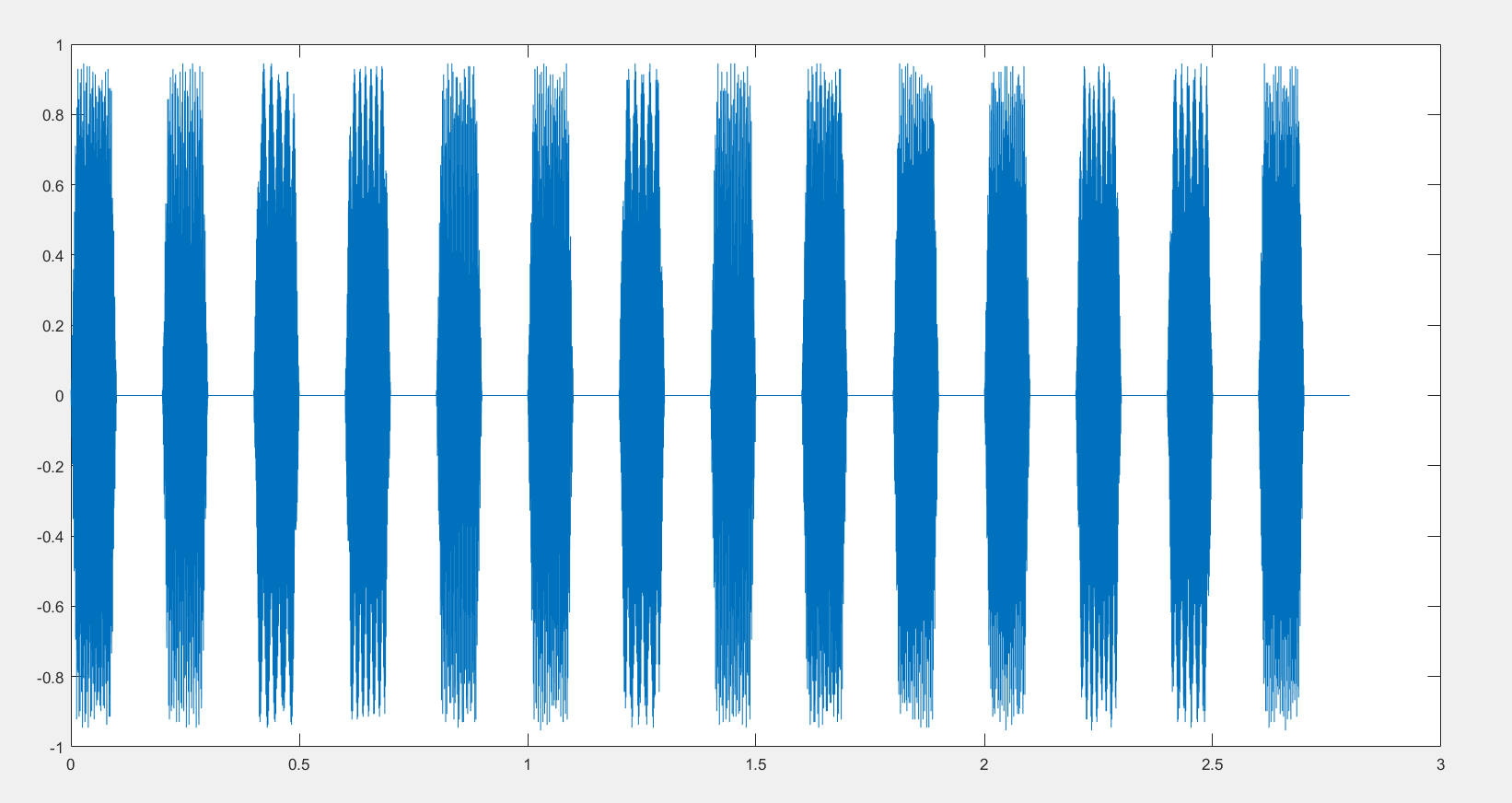
[data, freq] = audioread(filename);

time=(1:length(data))/freq;

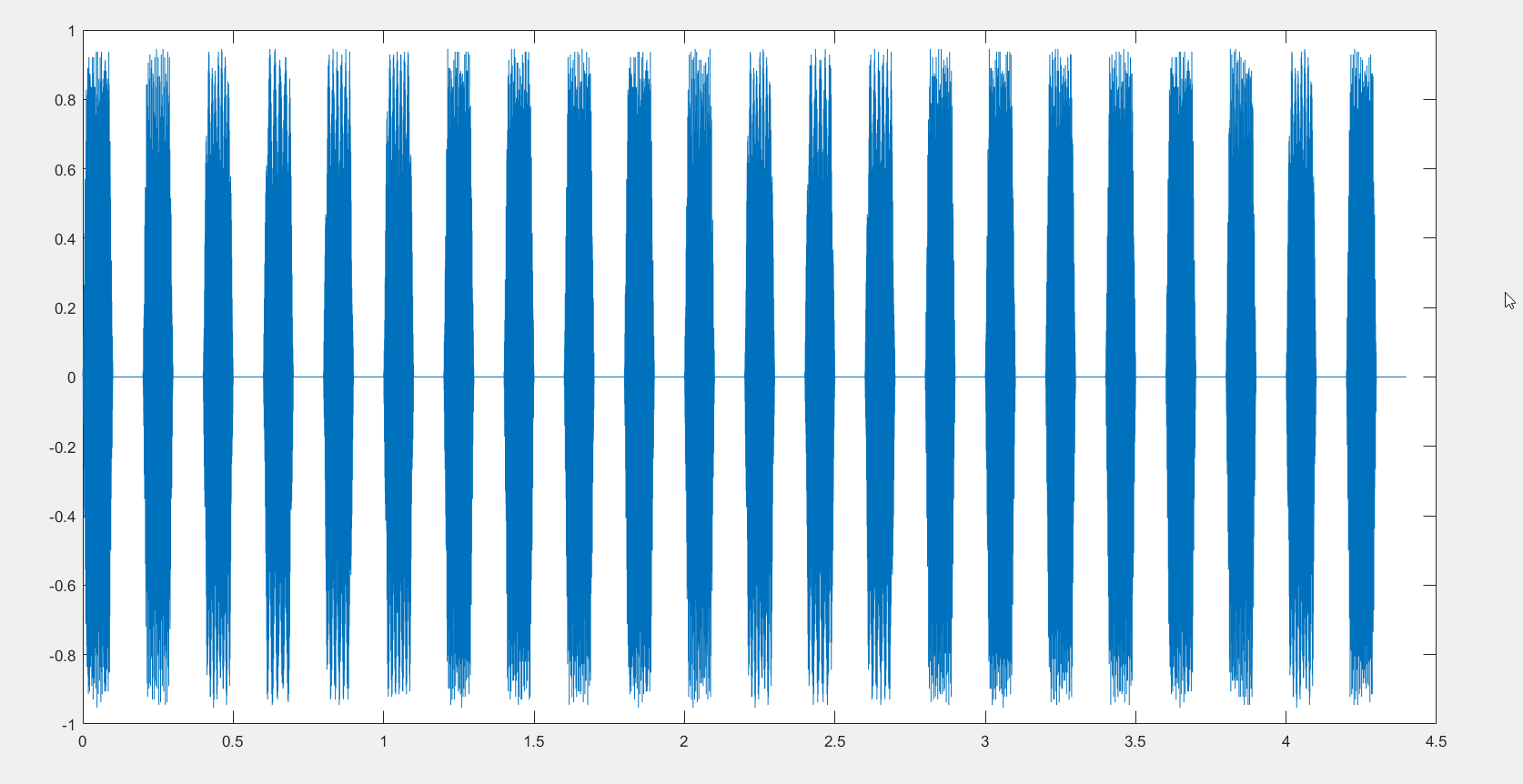
plot(time, data);

В результате получил графики следующего вида:

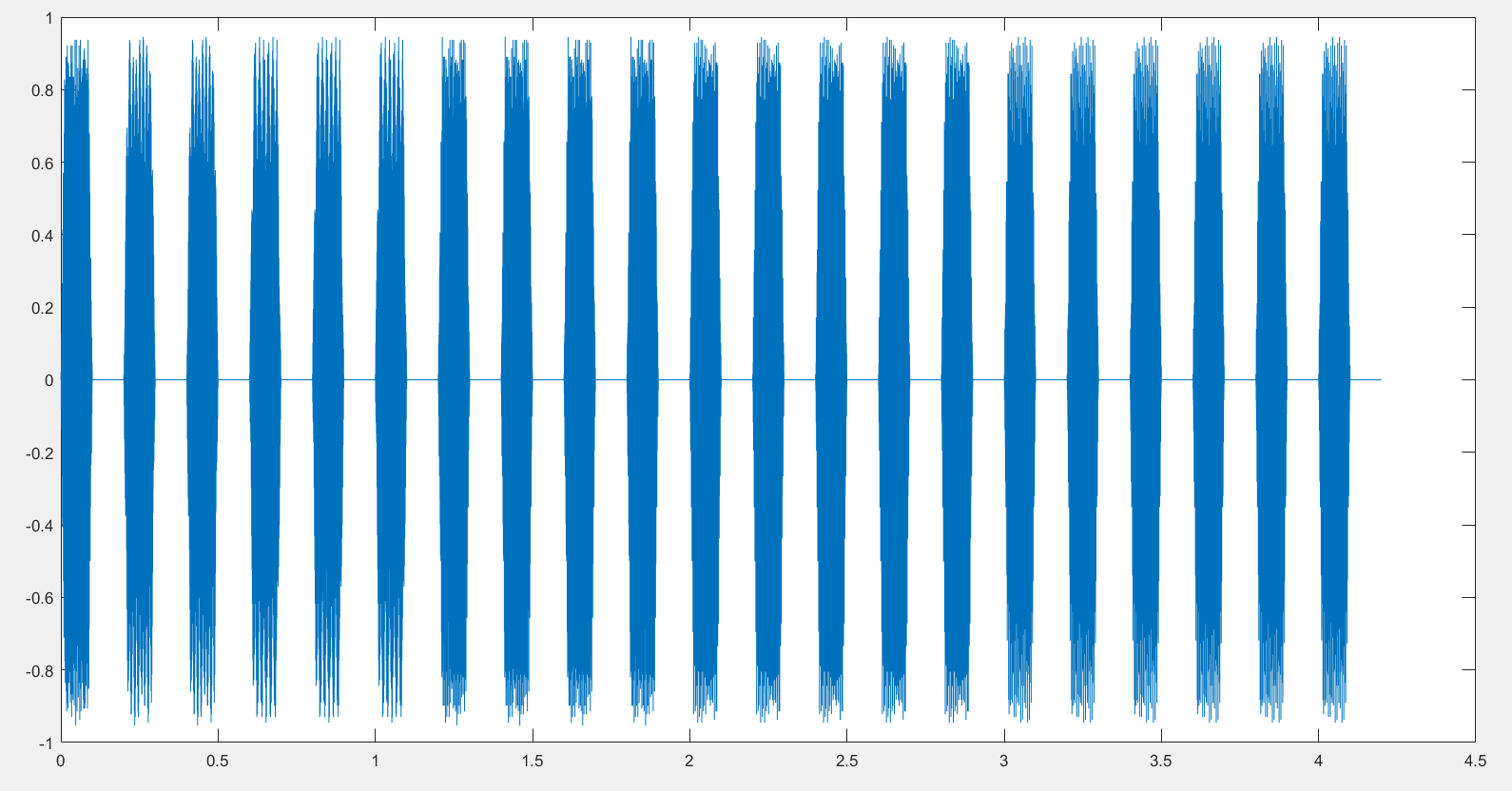
* s1



* s2



* s3



Из графиков видно, сколько раз был воспроизведен какой-либо звук.

В первом – 14, во втором – 22, в третьем – 21.

Алгоритм распознавания

**Шаг 1**

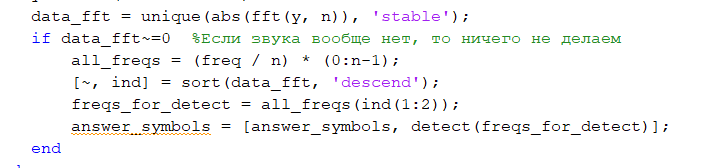
Сначала считываем файл функцией **audioread.**

filename='s1.wav';

[data, freq]=audioread(filename);

**Шаг 2**

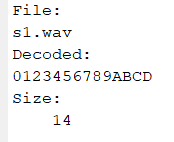
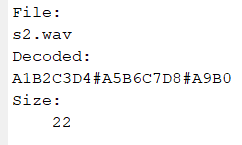
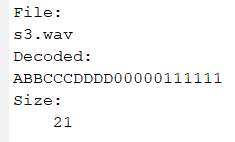
Найдя связь между длиной массива data и количеством цифр с предыдущего пункта, обрабатываем наш массив по 1600 значений, используя встроенную функцию **fft()**. Ответы храним в **answer\_symbols.** Обрабатываем данные, сохраняя только уникальные. Также строим все возможные частоты и определяем какие *две частоты* выделяются на общем уровне.



**Шаг 3**

Далее, используя if-else определяем по двум частотам какая была цифра и добавляем ее в **answer\_symbols**. В итоге печатаем название файла, расшифровку и длину расшифровки (т.е. количество нажатых цифр).

Результаты распознавания:

3. Код программы

function DTMF\_Decoder\_FFT ()

FFT('s1.wav')

FFT('s2.wav')

FFT('s3.wav')

end

function FFT(filename)

[data, freq] = audioread(filename);

time=(1:length(data)) / freq;

plot(time, data);

answer\_symbols = [];

n = 1600;

for i=1:length(data)/n

start = (i - 1) \* n + 1;

finish = i \* n;

y = data(start:finish);

data\_fft = unique(abs(fft(y, n)), 'stable');

if data\_fft~=0 %Если звука вообще нет, то ничего не делаем

all\_freqs = (freq / n) \* (0:n-1);

[~, ind] = sort(data\_fft, 'descend');

freqs\_for\_detect = all\_freqs(ind(1:2));

answer\_symbols = [answer\_symbols, detect(freqs\_for\_detect)];

end

end

disp('File:');

disp(filename);

disp('Decoded:');

disp(answer\_symbols);

disp('Size:');

disp(length(answer\_symbols));

end

function c = detect(freqs)

required\_freqs = [697 770 852 941 1209 1336 1477 1633];

[~, ind] = min(abs(required\_freqs - freqs(1)));

hifreq = required\_freqs(ind);

[~, ind] = min(abs(required\_freqs - freqs(2)));

lofreq = required\_freqs(ind);

if hifreq < lofreq

tmp = hifreq;

hifreq = lofreq;

lofreq = tmp;

end

c='('; %bad symbol

if lofreq==697

if hifreq==1209

c='1';

end

if hifreq==1336

c='2';

end

if hifreq==1477

c='3';

end

if hifreq==1633

c='A';

end

end

if lofreq==770

if hifreq==1209

c='4';

end

if hifreq==1336

c='5';

end

if hifreq==1477

c='6';

end

if hifreq==1633

c='B';

end

end

if lofreq==852

if hifreq==1209

c='7';

end

if hifreq==1336

c='8';

end

if hifreq==1477

c='9';

end

if hifreq==1633

c='C';

end

end

if lofreq==941

if hifreq==1209

c='\*';

end

if hifreq==1336

c='0';

end

if hifreq==1477

c='#';

end

if hifreq==1633

c='D';

end

end

end