Audio Request Identification

For Patient Aid

Communication Lab II

Mini- Project Report

By

Asmita Goyal –Roll no. 36

Shreya K.R –Roll no. 60

Department of Electronics and Communication Engineering

MANIPAL INSTITUTE OF TECHNOLOGY

(A Constituent Institute of Manipal University)

MANIPAL-576 104

Month 2017

**Introduction:**

This project aims to develop a program using MATLAB that recognizes basic patient requests, like ‘I am very hungry’, or ‘I’m in pain’. This can be used in a hospital setup to communicate messages to the hospital staff or family members, when they aren’t close by.

To do this, different types of audio samples from the patient are pre-recorded. Each time the patient makes a request, an audio file is generated and compared with the existing audio files to find the closest match. Both the pre-recorded signals are sampled quantized and then further processed to perform this matching operation.

Further processing involves counting the number of negative samples, no. of samples with a zero value and an addition of the absolute value of all the signals. Each of these factors is used as a basis for comparison, and when a majority of these features match, the closest match is found. The comparison is performed by subtracting the corresponding values of each of these features in the pre-recorded samples and the new audio input, and the minimum answer post subtraction gives the nearest sample.

**Block Diagram:**

Pre-Recorded Audio

Count the number of negative samples

Subtract corresponding values and find closest match

Count no. of zero values

Quantize

Sample

Add all quantized samples

New audio input

**Methodology:**

Three functions were developed in MATLAB, to perform this task:

1. **First2.m:** This function in used to collect audio data from the user directly from the recorder present in the laptop. A recording object will be created, audio-data will be received from the user and finally be written onto a .wav file that will be stored for later use. This was done for 3 requests- “I’m very hungry”, “I’m in pain”, and “Please help me.” Resulting in three .wav files, hungry.wav, pain.wav and help1.wav.
2. **Squanti.m:** This function is used to sample and quantize the audio data present in a wav file. The data was already sampled at 8 KHz. From these audio files of 2 seconds each, with 8000 samples per second, 16000 samples were obtained. Each 64th sample amplitude was taken into a new matrix, thereby taking 250 samples from each audio file.

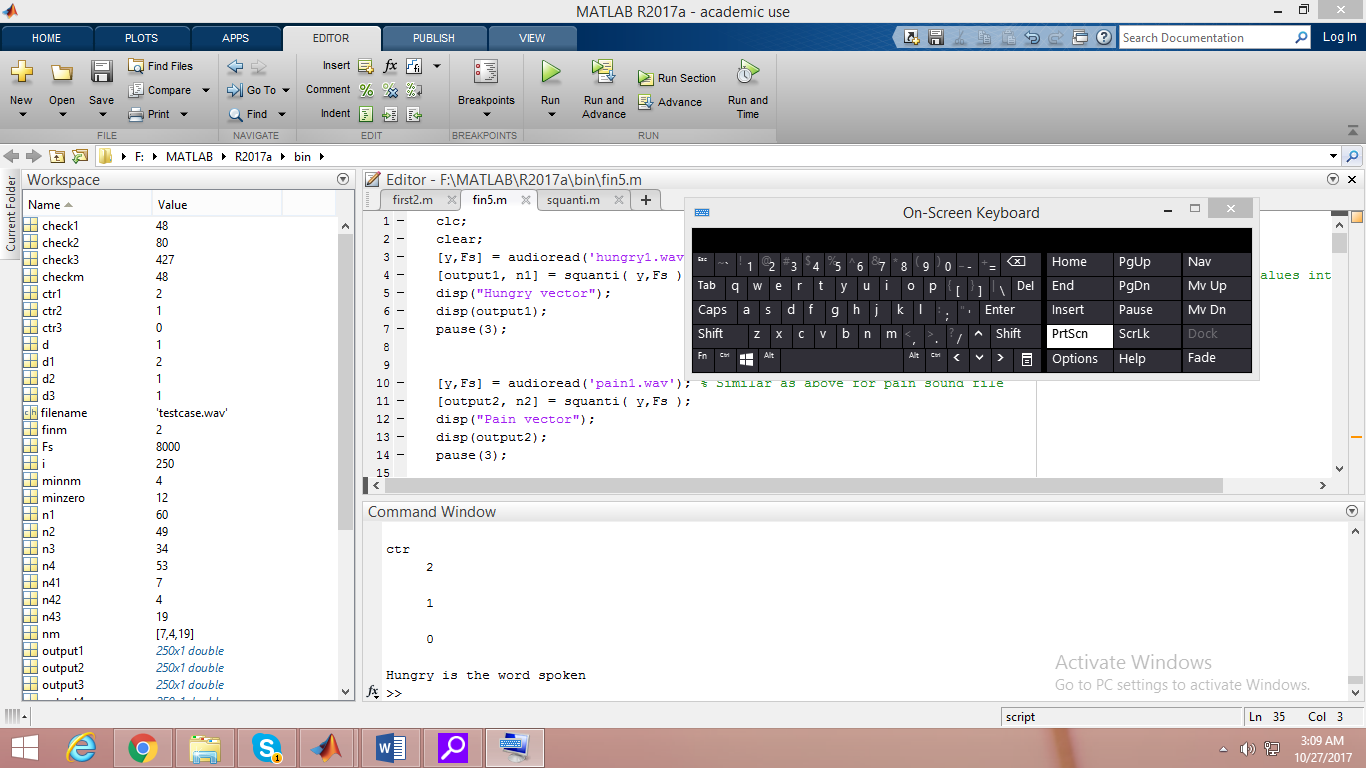
From theses 250 samples, the number of negative samples was calculated for later use. The samples were then amplified by a factor of hundred and their absolute value was taken. They were quantized into 16 levels, on the basis of their amplitude.

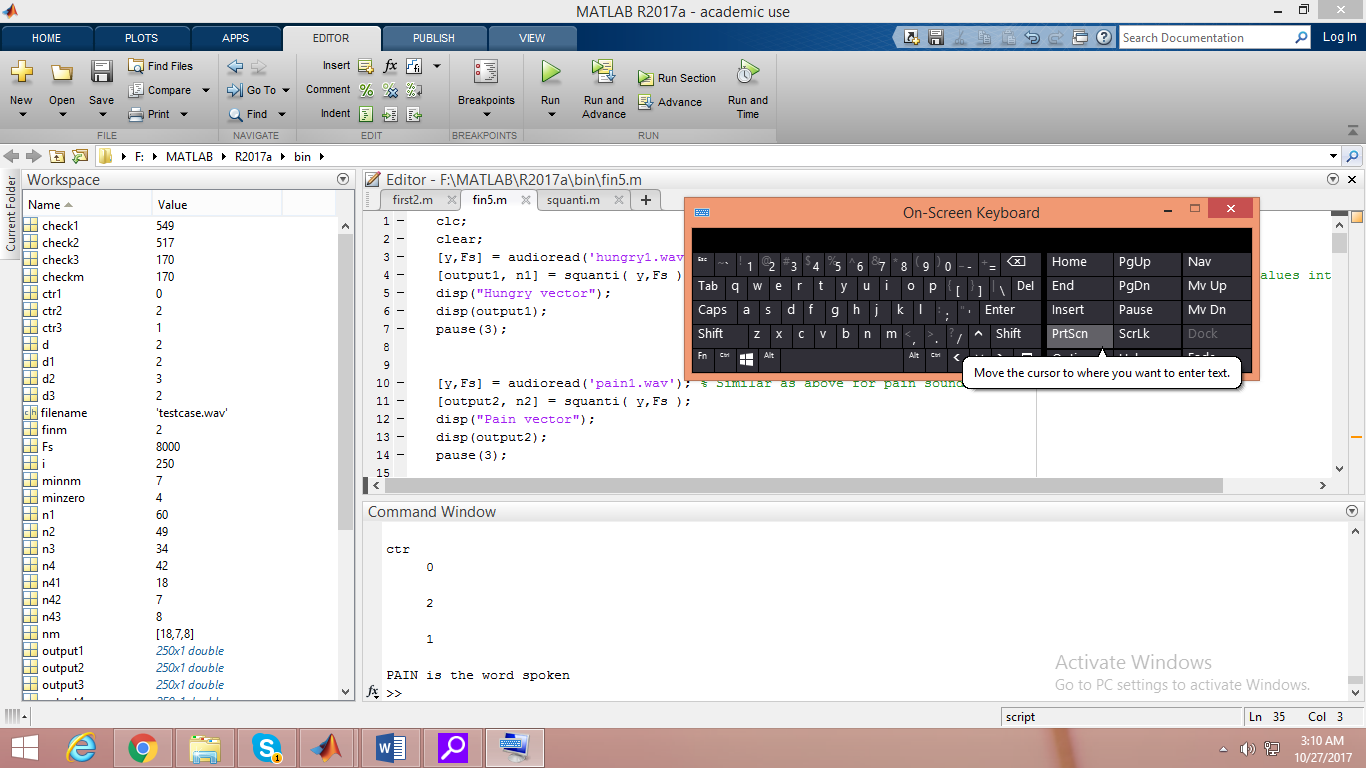
1. **Fin5.m:** This function is used to find the final match. Initially, it samples and quantizes each of the pre-recorded signals using the squanti function. Then it accepts audio input from the user which is again sampled and quantized in the same way. After this, the sum of all the 250 samples for each case is calculated. The zero occurrences in each audio sample is also calculated.

Now, we subtract the zero occurrences of each pre-recorded audio from the input audio and take the absolute value of it. This process is repeated for the number of negative samples and the sums. From each of these three features, a minimum will emerge corresponding to a particular pre-recorded audio. The pre-recorded audio with the maximum number of minimum correspondences is the best match. This match will be displayed as output.

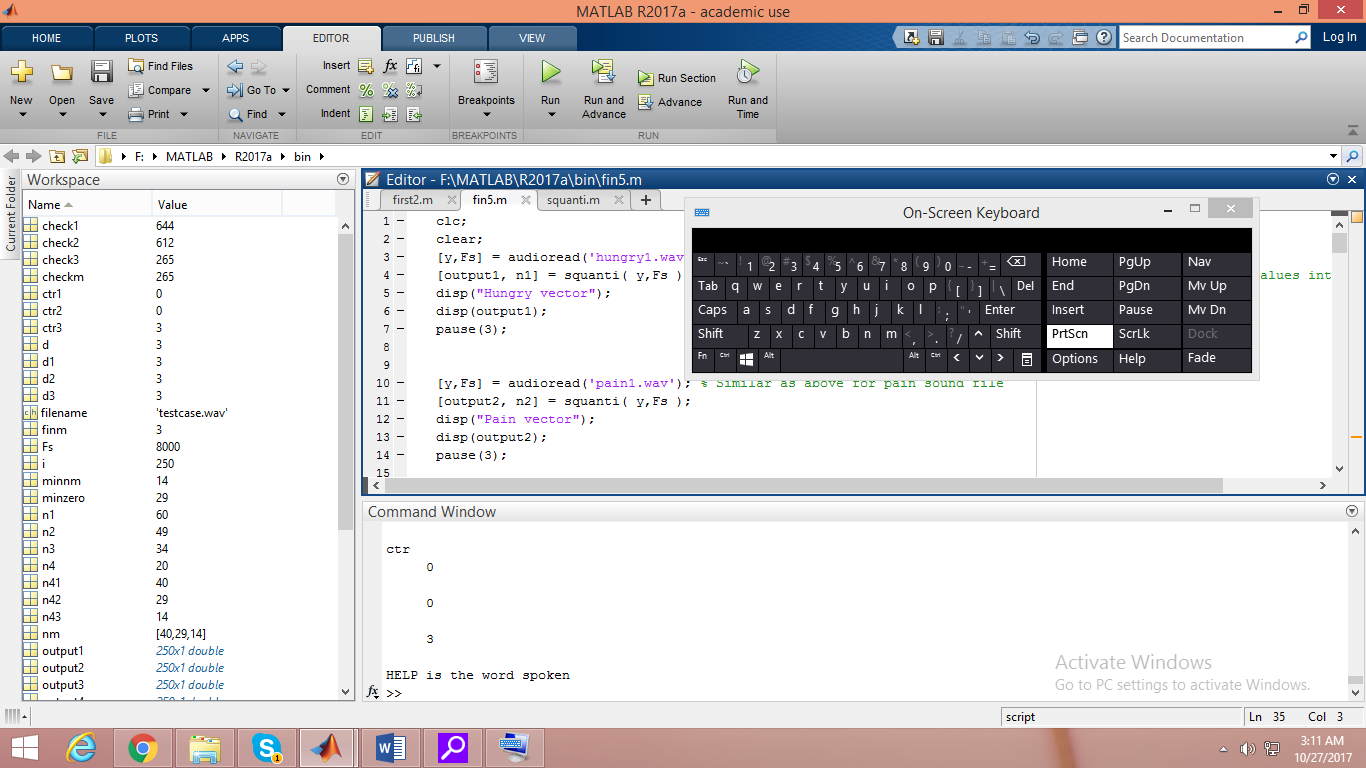
**Results:**

In the absence of external noise, the program successfully identified the request spoken. The ctr value indicates how many features out of three (zeroes, negatives and sum) gave minimum values, for each of the three pre-recorded signals. The audio with maximum minimum correspondences is chosen as the perfect match.



*Fig1 :When “I’m very hungry” is said.* 

*Fig2 :When “I’m in pain” is said*.



*Fig3: When “Please help” is said.*

The following graphs show all 250 samples plotted with their quantization levels from 0-15 (16 levels). As can be seen from the graphs, the number of negative samples, the number of zero samples and the sum of the amplitudes of the samples are different in each case and thus can be used as unique identifying features.

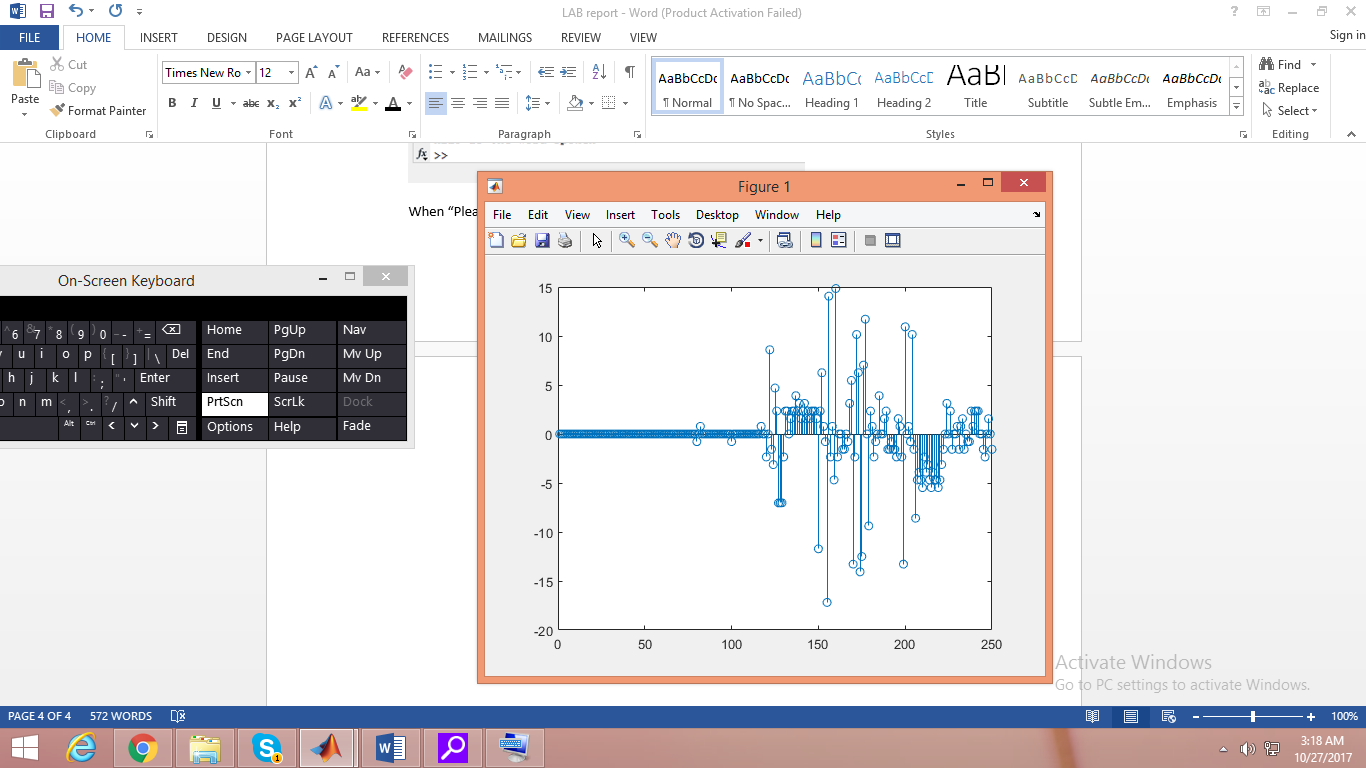
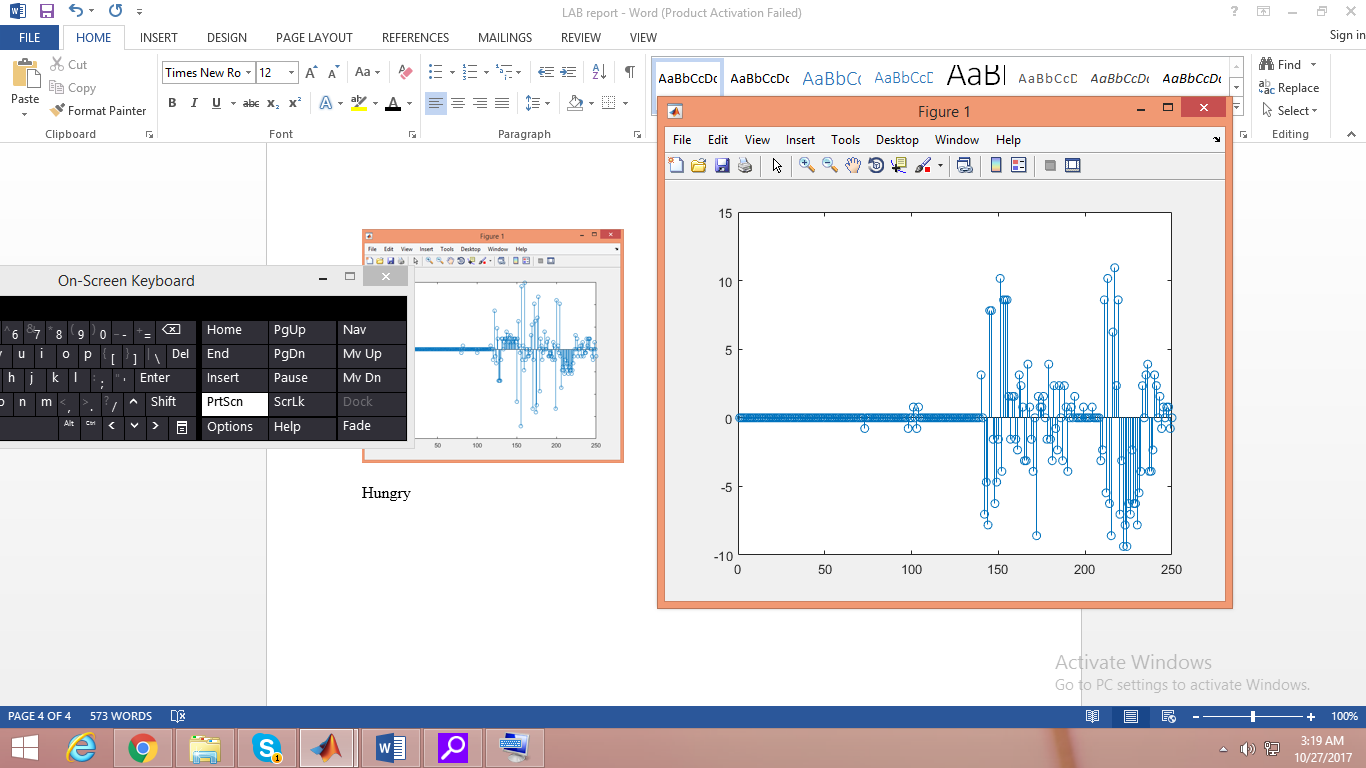
 

Fig 4: ‘Hungry’ samples Fig 5: ‘Pain’ samples

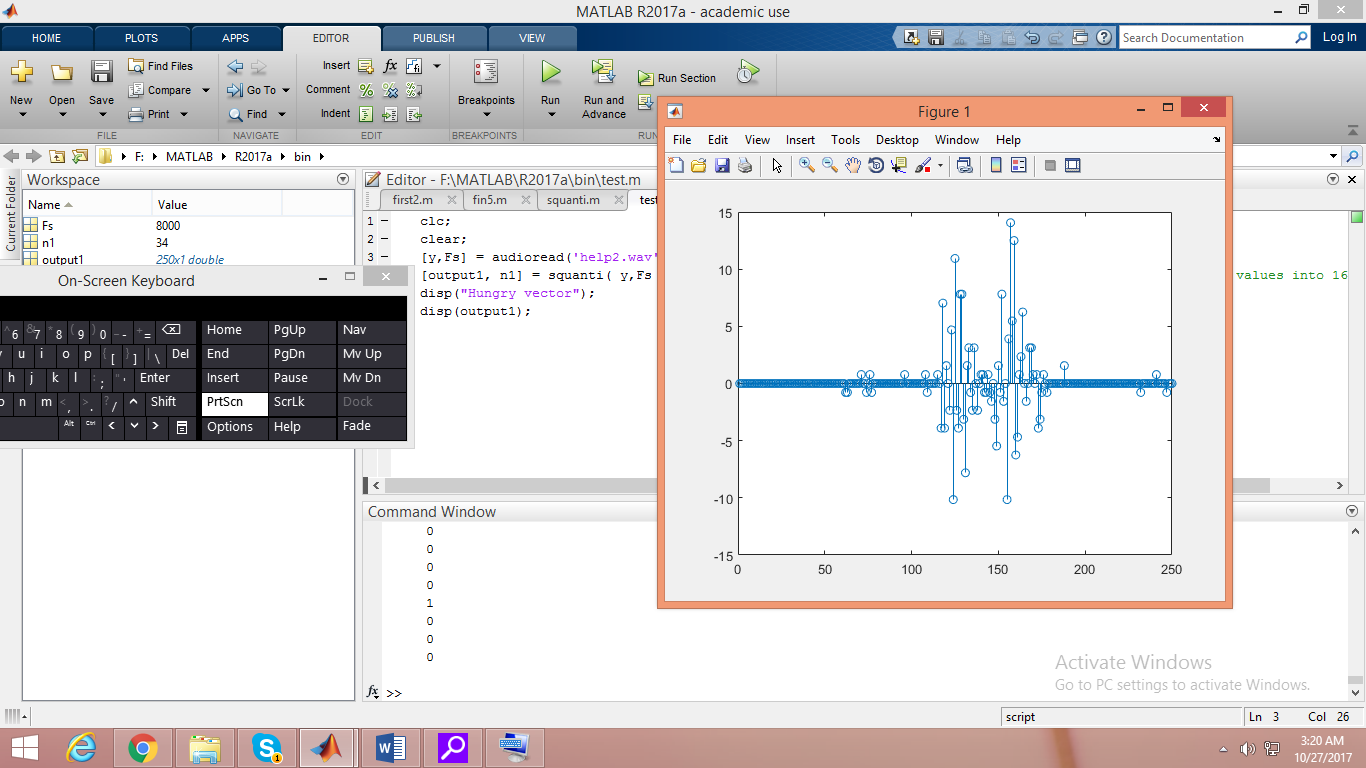


Fig 5: ‘help’ samples

The inputs used were all .wav files of 2 seconds duration. Due to the short length of the audio, and the use of amplitude based measures to uniquely identify the sample, this model is susceptible to external noise.

**Conclusion:**

This program uniquely identifies three patient request, in a considerably noise free environment. It does so, by counting the number of negative samples, zero samples, and calculating the sum of all samples. These three features are compared for each of the three pre-recorded audio and the audio input. The pre-recorded audio with maximum correspondences is matched to the input.

This is a very simple demonstration of the concept. Instead of taking pre-recorded audio from each user to customize the system to eat user, a neural network can be trained using a database from several users, such that a new input can easily be identified.

It can also be improved to work in the presence of considerable noise, by using frequency based measurements in place of amplitude.