# An isolated word speech to British Sign Language translator

*Pichon Antoine, Lostanlen Matéo*

Bordeaux INP
ENSEIRB
MATMECA

# Table of contents

# Introduction

- Nowadays, with the help of many connected objects, we can improve the daily life of deaf or hard-of-hearing people who actually have considerable difficulties in order to communicate with other people. This is a crucial issue and engineers plays a key role in solving this issue by devising innovative devices.

- Even if they can communicate with other deaf or hard-of-hearing people by using the British Sign Language, the communication with non-disabled people is still challenging that is why we will study and set up in this project a part of a system which can translate spoken words into British Sign Language pictures or videos.

- This system works by using a classifier, and it works primarily in two phases : Learning Phase and Testing Phase.

  The Learning Phase is used for extracting features from data files which contains speech signals in a set of known data named train set in order to learn the model parameters.

  The Testing Phase is used for comparing features extracted from a signal of a set of data named test set with the features extracted in the train set during the Learning phase in order to estimate the confusion's rate during the classification of features.

  Finally, the classifier extracts features from the input speech signal and compares them with the features extracted during the Learning Phase to determine which words are spoken.

# Speech signal analysis

## *Introduction to speech models*

Speech is formed by two phenomena.
Firstly, an input signal is formed by breathing out air from lungs, and then, this input signal is modulated by the vocal tracts and the mouth.

Furthermore, speech is composed of a succession of small sounds named phonemes, these phonemes can be voiced (periodic) if they use vocal cords or unvoiced if they do not.

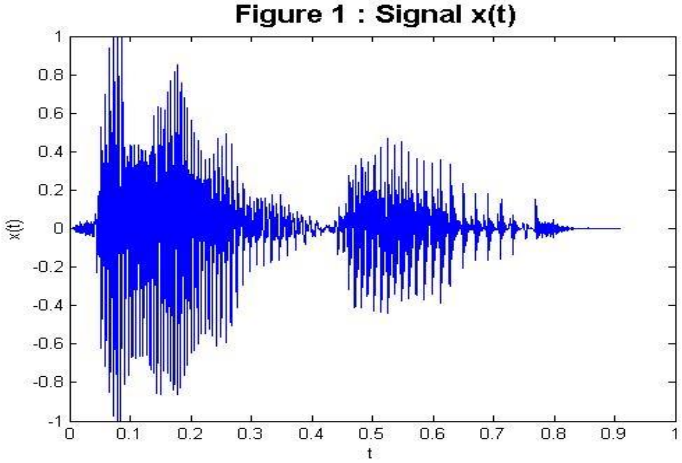
Based on this biological model, a signal processing model has been created, the goal is to establish a MATLAB model corresponding to this signal processing model.

In this part, the different features of a given speech signal "Hello world" will be analyzed, the state of the different phonemes and the frequency features.

## *Time analysis*

This graph represents the amplitude evolution in time of the speech signal $x(t)$ containing the sentence "Hello world."

**Figure 1 : Signal x(t)**




This speech signal is sampled at a sampling rate Fs.

Two different parts of this signal were studied :

-> $x_1(t) : x(t)$ restricted to $t \in [0.01s, 0.04s]$

-> $x_2(t) : x(t)$ restricted to $t \in [0.2s, 0.23s]$

**Figure 2 : Signal x1(t)**

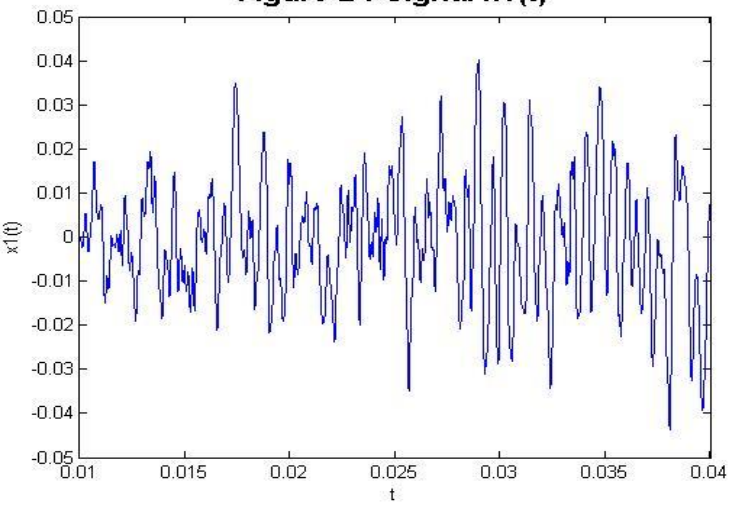
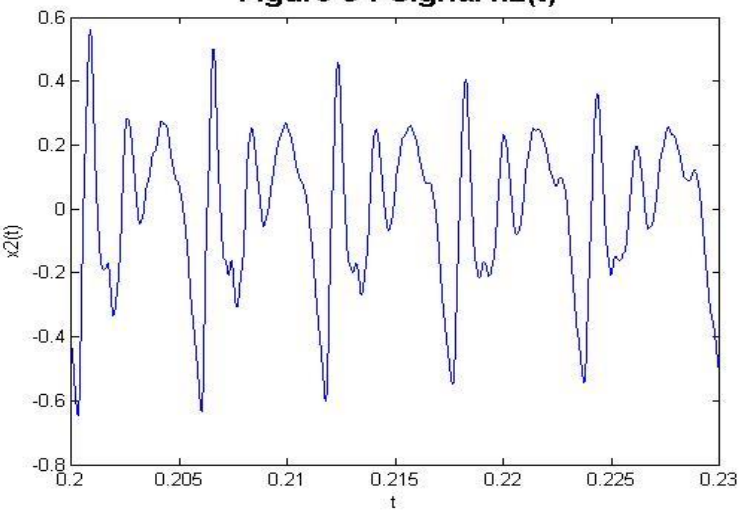



**Figure 3 : Signal x2(t)**

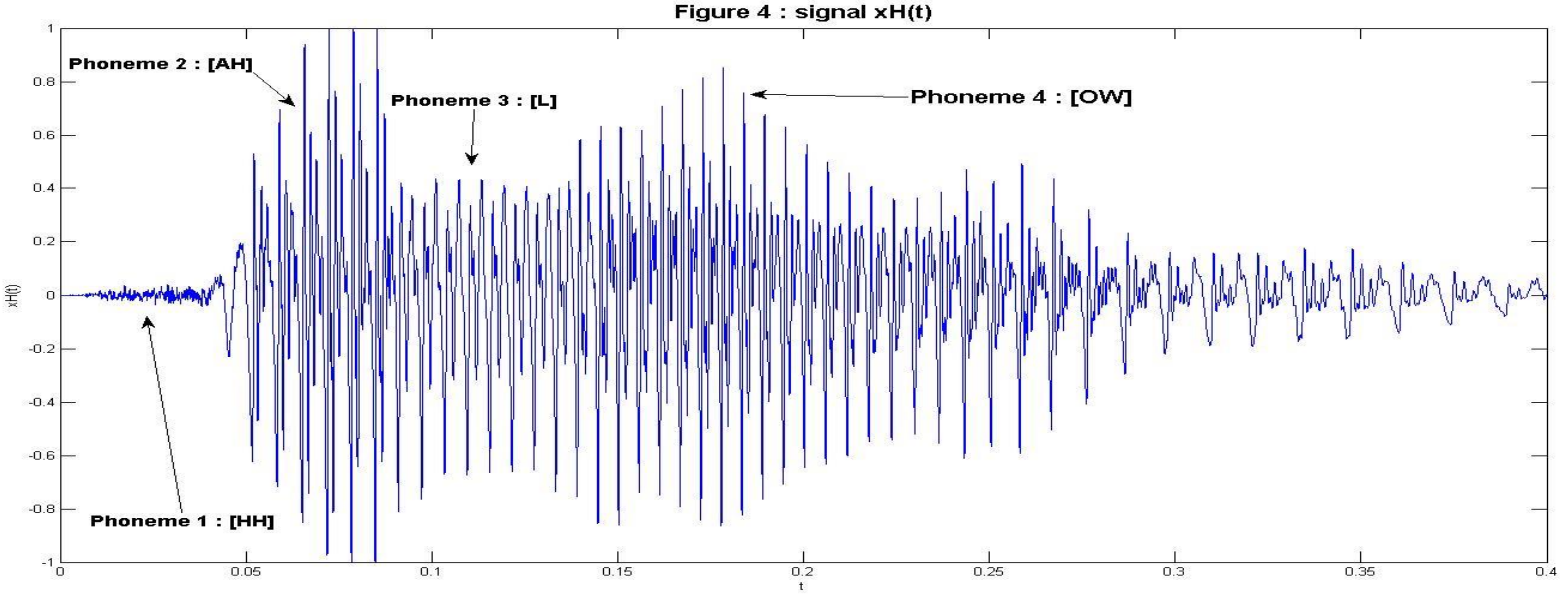



The purpose of this part is to identify if a part of $x(t)$ is voiced or unvoiced depending on whether this part is periodic or not, a voiced signal corresponds to a periodic signal and an unvoiced signal corresponds to a non-periodic signal.

Thus, one can notice that $x_1(t)$ is unvoiced and $x_2(t)$ is voiced, by observing the figure 2 and figure 3.

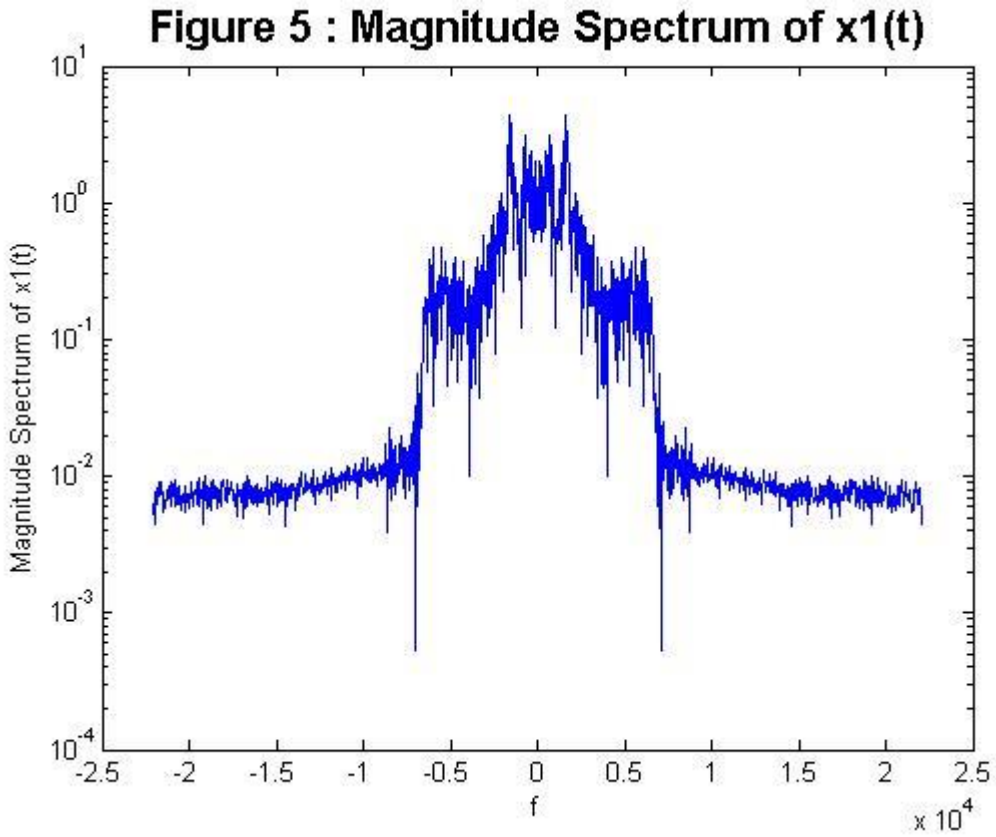
Then, the part of $x(t)$ which corresponds to "Hello" named $x_H(t)$ was studied, and it was seen where the phonemes are located :

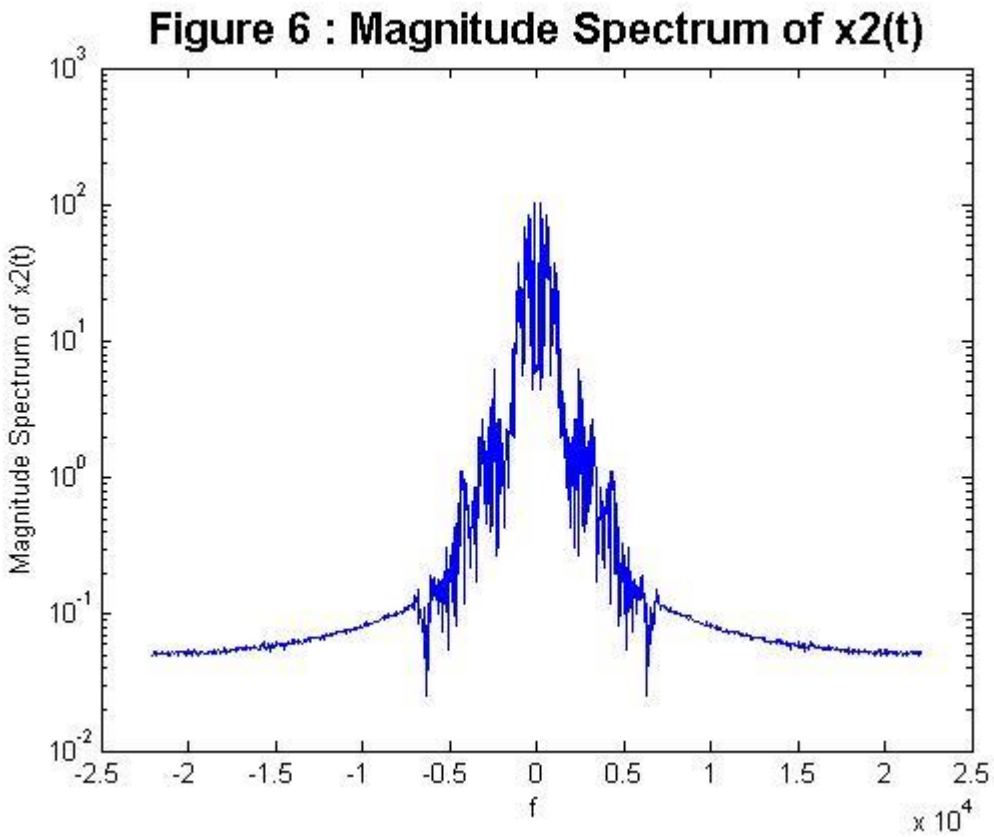
**Figure 4 : signal xH(t)**



*Frequency analysis*

In this part, the magnitude of $x_1(t)$ and $x_2(t)$ was computed and plotted for $f \in \left[-\frac{Fs}{2}, \frac{Fs}{2}\right]$ in Hz :




One can notice that the magnitude of $x_1(t)$ is significantly lower than the magnitude of $x_2(t)$, and also that $x_1(t)$ is a "noisy" part of $x(t)$.

The previous statement is logical, because the noise is negligible compared to the amplitude of $x_2(t)$.

Thus, the fact that $x_1(t)$ is unvoiced and $x_2(t)$ is voiced is obvious.

## *Short-Term Fourier Transform and Spectrogram*

In this part, two functions stft and spectro were created, these functions perform the Short-Term Fourier Transform of a speech signal $x(t)$ and computes its spectrogram.

The Short-Term Fourier Transform was used because the fft function doesn't take into account the time evolution of the frequency, which is a crucial issue for speech signal studying.
For instance, in a symphony, two same notes will be layered on the spectrum.

In addition to that, the signal $x(t)$ is stationary for periods of 20 to 30 milliseconds, hence an analysis of every $x(t)$ signal samples using a sliding window of duration 20-30 milliseconds and a computing of the Discrete Fourier Transform of each window has to be done.
Mathematically, the STFT is expressed as :

$$\boldsymbol{X(m,v) = \sum_{n=0}^{L-1} x_n w_{n-m} e^{-j2\pi vn}}$$

With the help of these functions, three different spectrograms of $x(t)$ were plotted for different input parameters, by changing first the window signal w and then the number of samples N.

**1st Spectrogram :** $\boxed{N = 441, d = 441, N_{fft} = 1024, w = hamming(N)}$

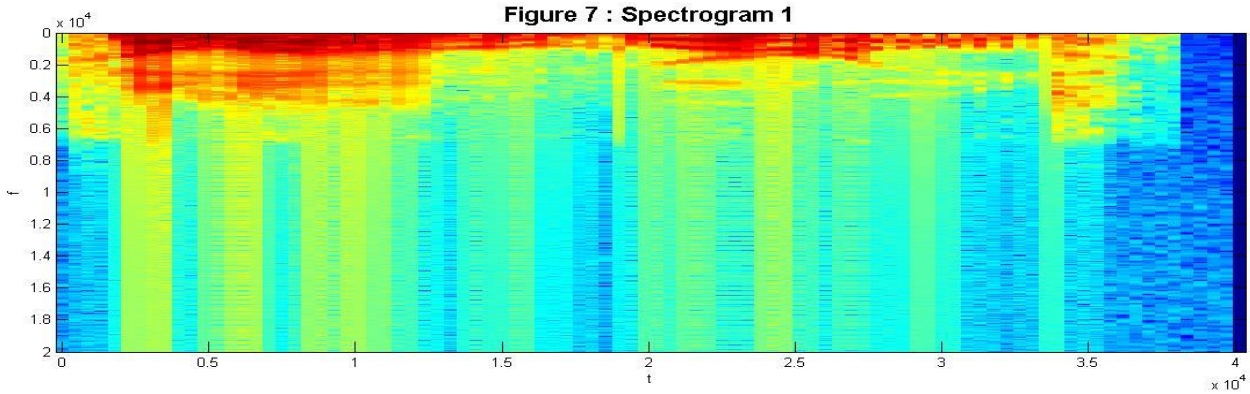


Figure 7 : Spectrogram 1

**2nd Spectrogram :** $\boxed{N = 441, d = 441, N_{fft} = 1024, w = ones(1, N)}$

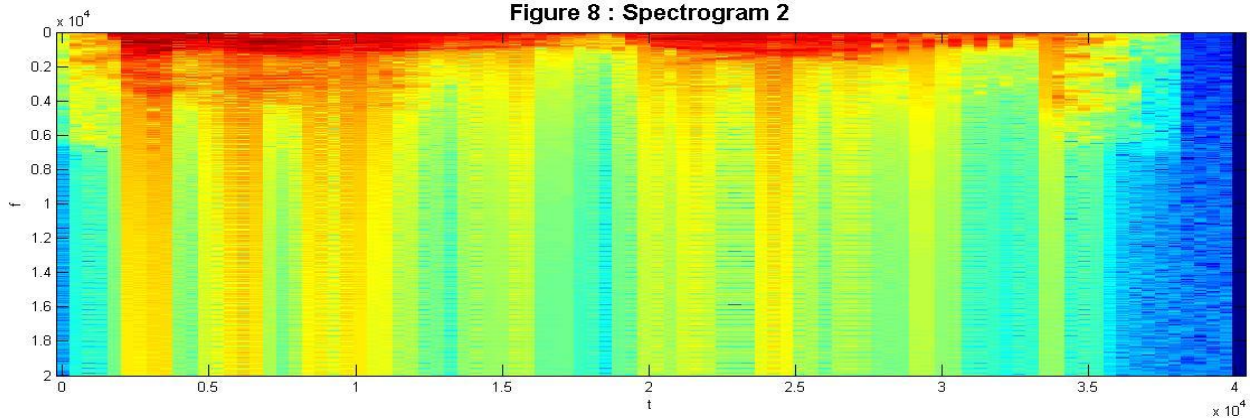


Figure 8 : Spectrogram 2

**3rd Spectrogram :** $\boxed{N = 882, d = 441, N_{fft} = 1024, w = hamming(N)}$

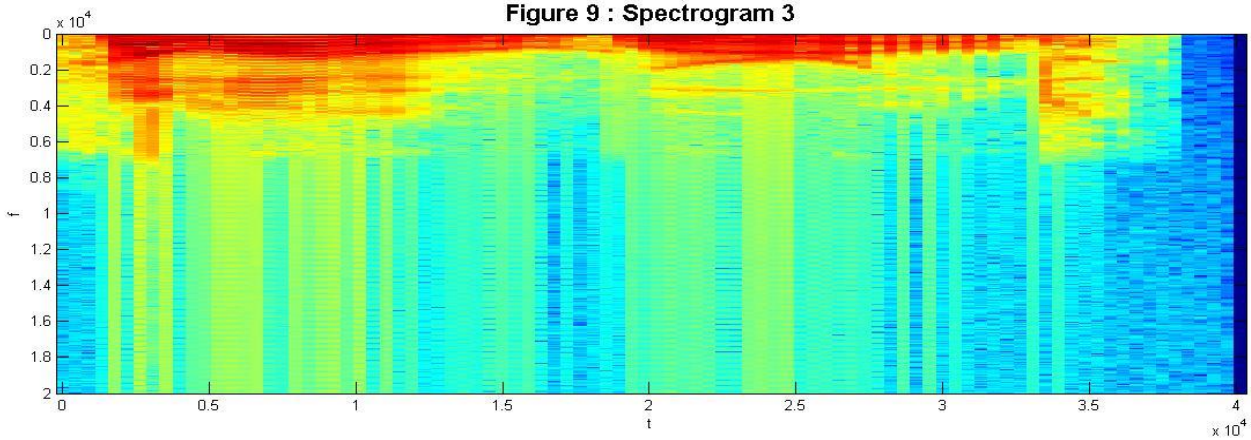


Figure 9 : Spectrogram 3

First and foremost, a spectrogram is the representation of the Fourier's amplitude of a given speech signal in the time-frequency plane.

The color's scale indicate the amplitude's value.

One can notice that the narrower the temporal window is, the poorer the frequency's resolution is.

Thus, the temporal windows can be classified by their size : $\boxed{hamming(441) < hamming(882) < ones(1{,}441)}$

That can explain why the 2nd Spectrogram is the best because it has a greater frequency's resolution, that is to say it has more intense colors, and also, the 3rd Spectrogram is better than the 1st Spectrogram.

---

# Feature extraction

---

## *Voiced/Voiceless flag and pitch*

In this part, the aim is to determine if a given speech signal is voiced or unvoiced.
With this aim in mind, the auto-correlation was computed for a given speech signal $x(t)$ by using an unbiased estimator $\gamma_u(p)$.

Let $P = \underset{p\in\left[\frac{Fs}{3000},\frac{Fs}{300}\right]}{\operatorname{argmax}} \gamma_u(p)$.

If $\gamma_u(P) > 0.6\gamma_u(0)$ , it will be considered that $x(t)$ is voiced, else $x(t)$ is unvoiced.

If $x(t)$ is voiced, this signal is periodic with a period T and the pitch is defined by $p = \frac{1}{T}$, although if $x(t)$ is unvoiced, this signal is non-periodic and the pitch $p = 0$.

So, two functions were created which are named autocorr and isvoiced, autocorr computes $\gamma_u(p)$ and isvoiced evaluates if the speech signal is voiced or unvoiced and computes the pitch.

These functions were used for four signals one1, one2, two1 and two2 and it was noticed that two1 is the only one signal to be voiced and its pitch is equal to $p = \frac{1}{T} \approx 182\ Hz$.
So, the coefficients which corresponds to a same number seem actually to be independent of each other, and thus these coefficients are not reliable enough for speech recognition.

## *MFCC*

In this part, it is asked to compute the Mel-Frequency Cepstrum Coefficients (MFCC) of the four previous speech signals one1, one2, two1, two2 in order to realize speech classification and recognition.

The study of MFCC of a given speech signal enables the extraction of features of this signal around the FFT and the DCT, converted on the mel scale.

The main advantage of this study is that the coefficients computed are uncorrelated.

The steps of the computation of the MFCC are described below :

1 . **Frame blocking :** Frame the signal into short frames of N samples
2 . **Hamming windowing** : Each frame has to be multiplied with a hamming window in order to keep the continuity of the first and the last points in the frame
3 . **Discrete Fourier Transform :** Spectral analysis shows that different timbres in speech signals corresponds to different energy distribution over frequencies. Therefore we usually perform FFT to obtain the magnitude frequency response of each frame.

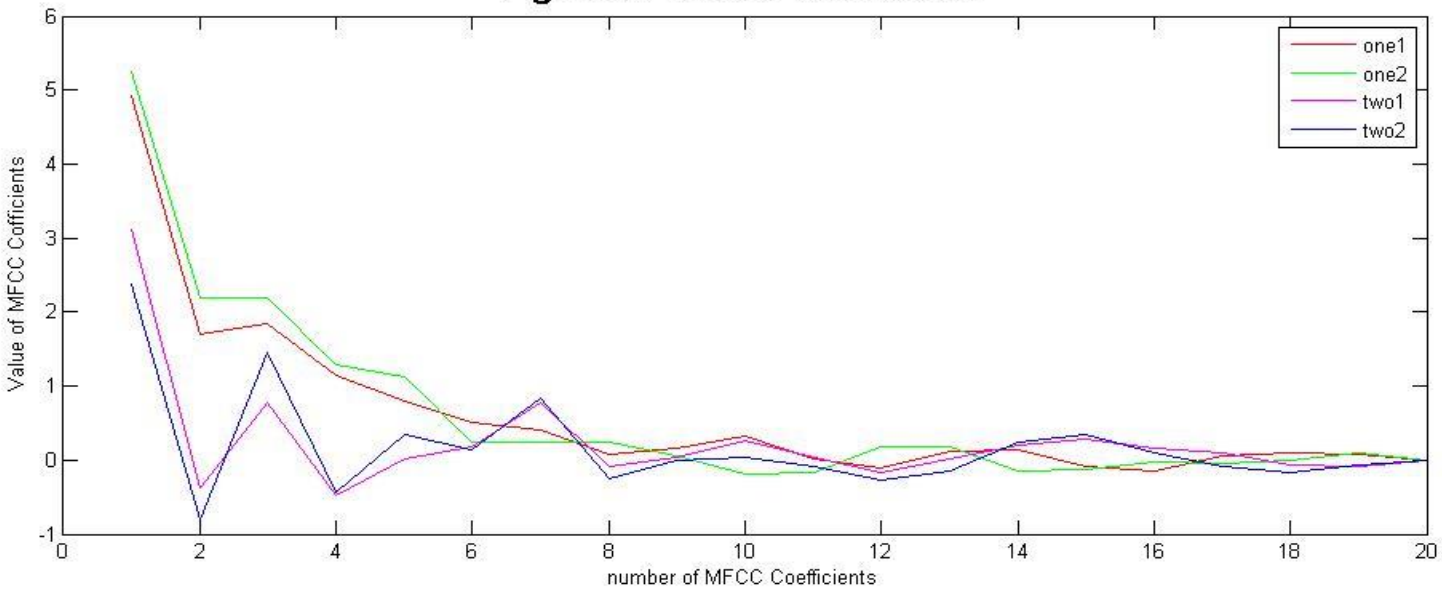
4 . **Triangular Bandpass Filters :** Multiplication of the magnitude frequency response by a set of 20 triangular bandpass filters in order to convert each frame on the mel scale
5 . **Log :** Computation of the log energy of each filtered signals.
6 . **Discrete cosine transform(DCT)** : In this step, application of the DCT on the 20 log energy $E_j$ obtained previously by using the following formula :

$$mfcc_i = \sqrt{\frac{2}{P}} \sum_{j=1}^{P} \log(E_j) \cos(\frac{\pi}{P} i(j - 0.5))$$

where I represents the ith MFCC coefficient with $i \in [1,20]$ and P represents the number of triangular bandpass filters.
Eventually, The MFCC coefficients are the amplitudes of the resulting spectrum, which are named MFCC features.

For that purpose, a computation of the MFCC features of the four speech signals one1, one2, two1, two2 was realized.

**Figure 10 : MFCC Coefficients**




As can be seen on the Figure 10, in spite of the fact that speech signals one1, one2, two1, two2 represents the numbers one and two spoken by two different people, the curves corresponding to the same number are clearly close.

That is why, MFCC Coefficients are much more reliable for speech recognition than the previous coefficients flag and pitch.

---

# Classification

---

In this part, pronounced numbers will be tried in two frame test, to achieve that mfcc coefficients of those numbers will be compared with a reference frame train.
To do that, the KNN algorithm is used, it involved comparing distance between number's mfcc coefficient and frame train's coefficient. The number will be the most recurrent among the closest K.

## *Train the classifier*

In a first time, a train_classifier function is built, which is a matrix with in each column the class and the mfcc coefficients for each sample.

It is applied to train_1 and train_2.

*Test the classifier*

Train_classifier is also applied to test_1 and test_2. Thanks to the KNN algorithm, a proba function is created, which takes as inputs data, train, test and K, which give us the probability of finding the correct test sample's class. For test 1 and 2, with K=10, the results are P1= 0.59 and P2= 0.66

Figure 11 : confusion matrix $test_1$

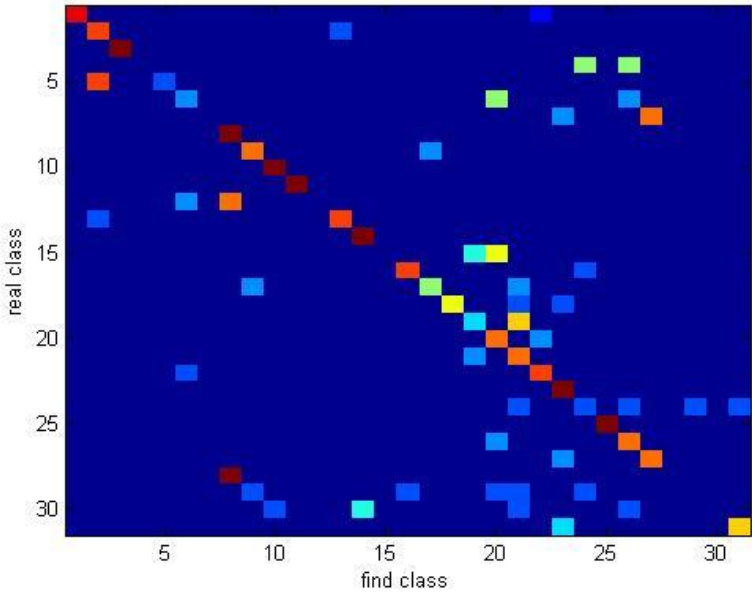
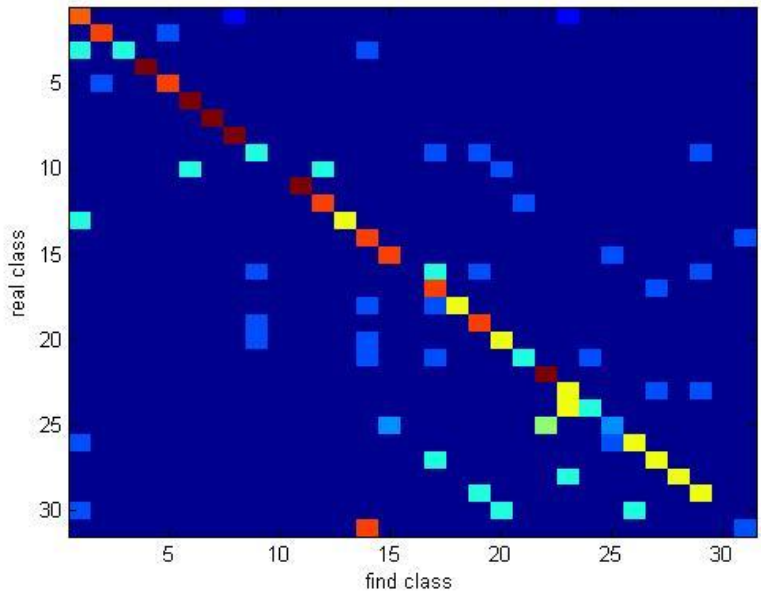



Figure 12 : confusion matrix $test_2$




More class are found in matrix 2, which is equivalent to more points on the straight line.
This is consistent with the past probabilities.
Probabilities can be also computed in this way :

$$P_1 = \frac{1}{30} * Tr(Cm_1) = 0,58$$

$$P_2 = \frac{1}{30} * Tr(Cm_2) = 0,68$$

Those probabilities are similar to the previous ones.

---

# Improved versions of the classifier

---

It is now time to loop the loop, that means build a true system with which our voice is recorded as inputs data.

This system will also translate it into British sign language (BSL) in outputs data. In this case, the system will takes two seconds recording of number between 1 and 20 in inputs data, and will post the corresponding sign in outputs data.

Therefore, a recording of 3 times of the number between 1 and 20 row is made.
Then, the room's noise is recorded.
Then, the first recording is subtracted by the second to delete the noise, a NB signal is then obtained.
The train_nb matrix is created, that allows to create the matFtr matrix with the train_classifier function.
Then, the real_time_classification function is applied in the same way than previously, which takes in inputs data the recording, computes the test matrix thanks to an intensity sensor and its matFte matrix with the train_classifier function.

The supposed class is consequently found, then the function posts the corresponding number's picture.

This system is quite reliable, but has its limits to find the correct number **:** only the frame train's recorder can use it. He must try to get the same intonation than the recording's intonation.
To improve the system, a data base can be imagined with a large recording choice and different voices.