# Digital Signal Processing Lab

## Lab Manual 8

### Objective:

In this Lab, our aim is to apply speech processing to identify the speaker of a specific speech. Mel frequency Cepstral coefficients algorithm (MFCC) is a technique which takes voice sample as inputs and generates spectral energy coefficients. After processing, the calculated are unique to a particular sample set (persons' voice) which are used for distinguishing different speakers.

### Description:

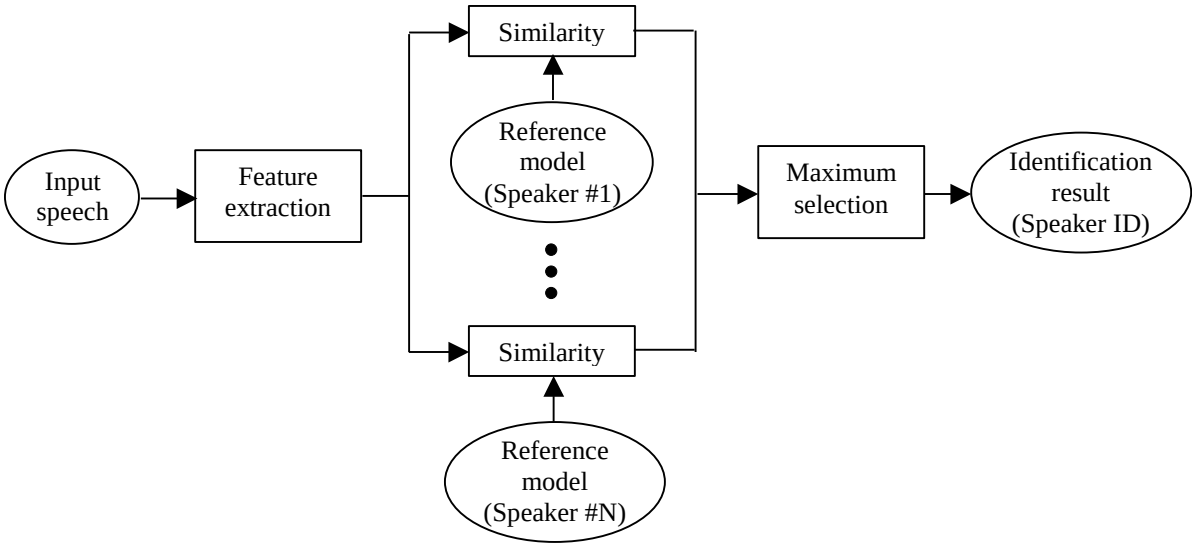
The goal of this lab/assignment is to build a simple, yet complete and representative automatic speaker recognition system. Due to the limited space, we will test our system on a small (but already non-trivial) speech database. There are 8 speakers, labeled from S1 to S8. All speakers uttered the same single digit "zero", once in a training session and once in a testing session. These sessions are at least 6 months apart to simulate the voice variation over the time.

A zip file containing data for training and testing data is available. After unzipping the file correctly, you will find two folders, TRAIN and TEST, each contains 8 files, named: S1.WAV, S2.WAV, ..., S8.WAV; each is labeled after the ID of the speaker. These files were recorded in Microsoft WAV format. In Windows systems, you can listen to the recorded sounds by double clicking into the files.

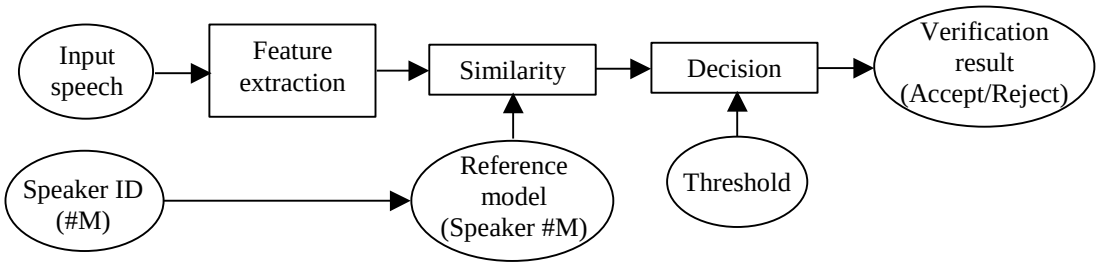
Your task is to train a voice model for each speaker S1 - S8 using the corresponding sound file in the TRAIN folder. After this training step, the system would have knowledge of the voice characteristic of each (known) speaker. Next, in the testing phase, the system will be able to identify the (assumed unknown) speakers of each sound file in the TEST folder.

### Algorithm for calculating MFCC:

In order to distinguish between the speakers we take into considerations the spectral changes which can be observed in speech of different people. At the highest level, all speaker recognition systems contain two main modules (refer to Figure 1): *feature extraction* and *feature matching*. Feature extraction is the process that extracts a small amount of data from the voice signal that can later be used to represent each speaker. Feature matching involves the actual procedure to identify the unknown speaker by comparing extracted features from his/her voice input with the ones from a set of known speakers.




(a) Speaker identification




(b) Speaker verification

**Figure 1**. Basic structures of speaker recognition systems

The complete system is divided into 2 parts, namely training and testing. The training stage creates a database of features for different people, these features are computed using MFCC. MFCC takes human perception sensitivity with respect to frequencies into consideration, and therefore are best for speech/speaker recognition. The step-by-step computation of MFCC is explained later on. While testing stage follows the same steps are training and generates features which are compared against the database (training features) using VQ codebook technique. The closest match is used to label the unknown testing data.

**The main purpose of the MFCC processor is to mimic the behavior of the human ears. In addition, rather than the speech waveforms themselves, MFFC's are shown to be less susceptible to variations.**

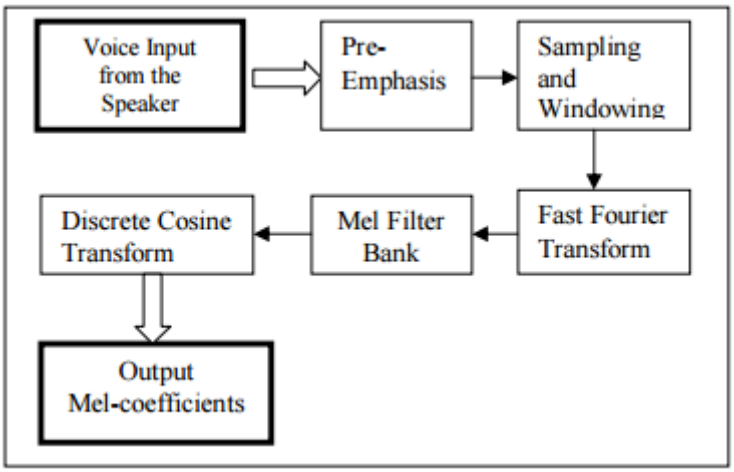



**Figure 2**. Basic diagram showing the integral blocks required in MFCC algorithm

Steps at a glance:

1. Apply high pass filter
2. Frame the signal into short frames.
3. For each frame calculate the periodogram estimate of the power spectrum.
4. Apply the mel filterbank to the power spectra, sum the energy in each filter.
5. Take the logarithm of all filterbank energies.
6. Take the DCT of the log filterbank energies.
7. Keep DCT coefficients 2-end, discard the rest

Detailed discussion is as follows

1. **PRE EMPHASIS:**
   a. The speech signal x(n) is sent to a high-pass filter :

$$y(n) = x(n) - a * x(n-1)$$

   where y(n) is the output signal and the value of a is usually between 0.9 and 1.0.

2. **FRAME BLOCKING**
   a. The input speech signal is segmented into frames of 15~20 ms with overlap of 50% of the frame size. E.g If the sample rate is 16 kHz and the frame size is 256 sample points, then the frame duration is 256/16000 = 0.016 sec = 16 ms.

b. Additional, for 50% overlap meaning 128 points, then the frame rate is 16000/(256-128) = 125 frames per second. Overlapping is used to produce continuity within frames.

c. In this step the continuous speech signal is blocked into frames of N samples, with adjacent frames being separated by M (M < N). The first frame consists of the first N samples. The second frame begins M samples after the first frame, and overlaps it by N - M samples and so on. This process continues until all the speech is accounted for within one or more frames. Typical values for N and M are N = 256 and M = 100.

## 3. HAMMING WINDOW

**a.** The next step in the processing is to window each individual frame so as to minimize the signal discontinuities at the beginning and end of each frame. The concept here is to minimize the spectral distortion by using the window to taper the signal to zero at the beginning and end of each frame.

**b.** Apply Hamming windowing as, $z_i(n) = y_i(n) .* w(n)$ where i =1 : no_Of_Frames

where w(n) is the Hamming window defined by:

$$w(n) = 0.54 - 0.46 * cos\left(\frac{2\pi n}{N-1}\right) (4) \qquad \text{where } 0 \le n \le N\text{-}1$$

## 4. FAST FOURIER TRANSFORM:

**a.** The next processing step is the Fast Fourier Transform, which converts each frame of *N* samples from the time domain into the frequency domain. The FFT is a fast algorithm to implement the Discrete Fourier Transform (DFT)

$$\boldsymbol{Q_i = fft(z_i)}$$ %where i=1: Total_number_of_Frames

b. In general $Q_i$'s are complex numbers and we only consider their absolute values (frequency magnitudes). The resulting sequence is interpreted as follow: positive frequencies $0 \le f < F_s/2$ correspond to values $0 \le n \le N/2-1$, while negative frequencies $-F_s/2 < f < 0$ correspond to $N/2+1 \le n \le N-1$. Here, Fs denotes the sampling frequency.

c. The result after this step is often referred to as spectrum or periodogram.

## 5. TRIANGULAR BANDPASS FILTERS:

a. Psychophysical studies have shown that human perception of the frequency contents of sounds for speech signals does not follow a linear scale. Thus for each tone with an actual frequency, f, measured in Hz, a subjective pitch is measured on

a scale called the 'mel' scale. The mel-frequency scale is a linear frequency spacing below 1000 Hz and a logarithmic spacing above 1000 Hz.

b. One approach to simulating the subjective spectrum is to use a filter bank, spaced uniformly on the mel-scale (see Figure 4). That filter bank has a triangular bandpass frequency response, and the spacing as well as the bandwidth is determined by a constant mel frequency interval. The number of mel spectrum coefficients, K, is typically chosen as 20. Note that this filter bank is applied in the frequency domain, thus it simply amounts to applying the triangle-shape windows as in the Figure 4 to the spectrum. A useful way of thinking about this mel-wrapping filter bank is to view each filter as a histogram bin (where bins have overlap) in the frequency domain.

c. The formula for converting from frequency to Mel scale is (Equation 1):

$$M(f) = 2595 \log_{10}\left(1 + \frac{f}{700}\right) \cong 1125 \ln\left(1 + \frac{f}{700}\right)$$

d. To go from Mels back to frequency (Equation 2):

$$M^{-1}(m) = 700(\exp(m/1125) - 1)$$

e. To get the filterbanks shown in figure 3, we first have to choose a lower and upper frequency. E.g. let us consider 300Hz for the lower and 8000Hz for the upper frequency. Of course if the speech is sampled at 8000Hz our upper frequency is limited to 4000Hz. Then follow these steps:

   i. Using equation 1, convert the upper and lower frequencies to Mels. In our case 300Hz is 401.25 Mels and 8000Hz is 2834.99 Mels.

   ii. For this example we will do 10 filterbanks, for which we need 12 points. This means we need 10 additional points spaced linearly between 401.25 and 2834.99. This comes out to:

   m(i) = 401.25, 622.50, 843.75, 1065.00, 1286.25, 1507.50, 1728.74, 1949.99, 2171.24, 2392.49, 2613.74, 2834.99

   iii. Now use equation 2 to convert these back to Hertz:

   h(i) = 300, 517.33, 781.90, 1103.97, 1496.04, 1973.32, 2554.33, 3261.62, 4122.63, 5170.76, 6446.70, 8000

   iv. Notice that our start- and end-points are at the frequencies we wanted.

   v. We don't have the frequency resolution required to put filters at the exact points calculated above, so we need to round those frequencies to the nearest FFT bin. This process does not affect the accuracy of the features. To

convert the frequncies to fft bin numbers we need to know the FFT size and the sample rate,

vi. f(i) = floor( (nfft+1)*h(i)/samplerate)

vii. This results in the following sequence:

f(i) = 9, 16, 25, 35, 47, 63, 81, 104, 132, 165, 206, 256

viii. We can see that the final filterbank finishes at bin 256, which corresponds to 8kHz with a 512 point FFT size.

ix. Now we create our filterbanks. The first filterbank will start at the first point, reach its peak at the second point, then return to zero at the 3rd point. The second filterbank will start at the 2nd point, reach its max at the 3rd, then be zero at the 4th etc. A formula for calculating these is as follows:

$$H_m(k) = \begin{cases} 0 & k < f(m-1) \\ \dfrac{k - f(m-1)}{f(m) - f(m-1)} & f(m-1) \le k \le f(m) \\ \dfrac{f(m+1) - k}{f(m+1) - f(m)} & f(m) \le k \le f(m+1) \\ 0 & k > f(m+1) \end{cases}$$

where $M$ is the number of filters we want, and $f()$ is the list of M+2 Mel-spaced frequencies. The final plot of all 10 filters overlayed on each other is:

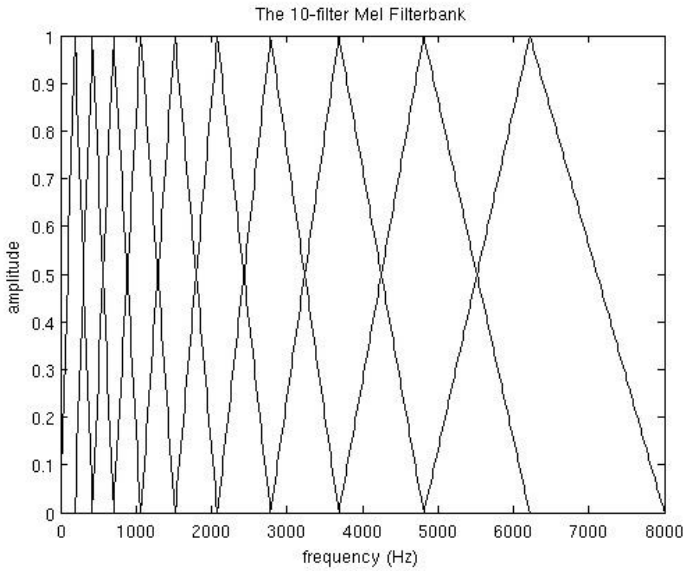



**Figure 3**. An example of mel-spaced filterbank

## 6. DISCRETE COSINETRANSFORM:

In this final step, we convert the log mel spectrum back to time. The result is called the mel frequency cepstrum coefficients (MFCC). The cepstral representation of the speech spectrum provides a good representation of the local spectral properties of the signal for the given frame analysis. Because the mel spectrum coefficients (and so their logarithm) are real numbers, we can convert them to the time domain using the Discrete Cosine Transform (DCT). Therefore if we denote those mel power spectrum coefficients that are the result of the last step are $\tilde{S}_0, k = 1,2,...,K-1$, we can calculate the MFCC's, $\tilde{c}_n$, as

$$\tilde{c}_n = \sum_{k=1}^{K} (\log \tilde{S}_k) \cos\left[ n\left( k - \frac{1}{2} \right) \frac{\pi}{K} \right], \qquad n = 0,1,..., K\text{-}1$$

Note that we exclude the first component, $\tilde{c}_0$, from the DCT since it represents the mean value of the input signal, which carried little speaker specific information.

$c = dct(spectr_i);$ `% use built in function`

`%i = 1:Total number of samples`

## 7. Summary

By applying the procedure described above, for each speech frame of around 30msec with overlap, a set of mel-frequency cepstrum coefficients is computed. These are result of a cosine transform of the logarithm of the short-term power spectrum expressed on a mel-frequency scale. This set of coefficients is called an *acoustic vector*. Therefore each input utterance is transformed into a sequence of acoustic vectors.