Pitch and formants estimation

A project of Speech Processing and Recognition Course

University of Genoa

Zeinab Namakizadeh Esfahani
4786728

1. Introduction and method:

This project aims to identify the parameters of an audio file. These parameters and their changes over time, can be later used in Speech Analysis and recognition. Pitch is the fundamental frequency of a signal that has the lowest frequency but high magnitude. Pitch (F0) is produced by the excitation and is a characteristic of quasi periodic signals (Voiced), namely vowels, semi-vowels, and nasals.

Next significant frequencies are called Formants. Formants are the parameters by which the pitch is affected while it passes a lossless tube (a digital time varying filter) called vocal tract. These frequencies mainly vary due to the phoneme being articulated rather than age, gender, etc. Each formant corresponds to a resonance in the vocal tract. The Pitch and formants are very important to model the languages, and determining the age, gender, and especially emotion of the speaker.

To extract the pitch, the autocorrelation function or the Cepstrum analysis can be used. In this project, only Cepstrum analysis is used. To extract the formants, a common method is to employ DFT log-spectrum and then choose the peaks. Another method is called Linear Predictive Coding. Both methods are implemented in this project in order to have the possibility of comparing the results.

As we know in theory, since the speech parameters vary with time especially for consonants, it is needed to search the parameters in each frame assuming that these characteristics do not vary much during this frame. The duration of the frame is adjustable in this project, but the default duration is 30msec as an approximation of the duration of a phoneme. As we limit our search in a frame, the short time analysis is needed. Which means, the operations are all done on a slice of signal once at a time.

Keywords- Pitch detection, formant estimation, homomorphic analysis, log-spectrum.

2. Guide to use

This project is done by Matlab, the GUI is explained in this section.

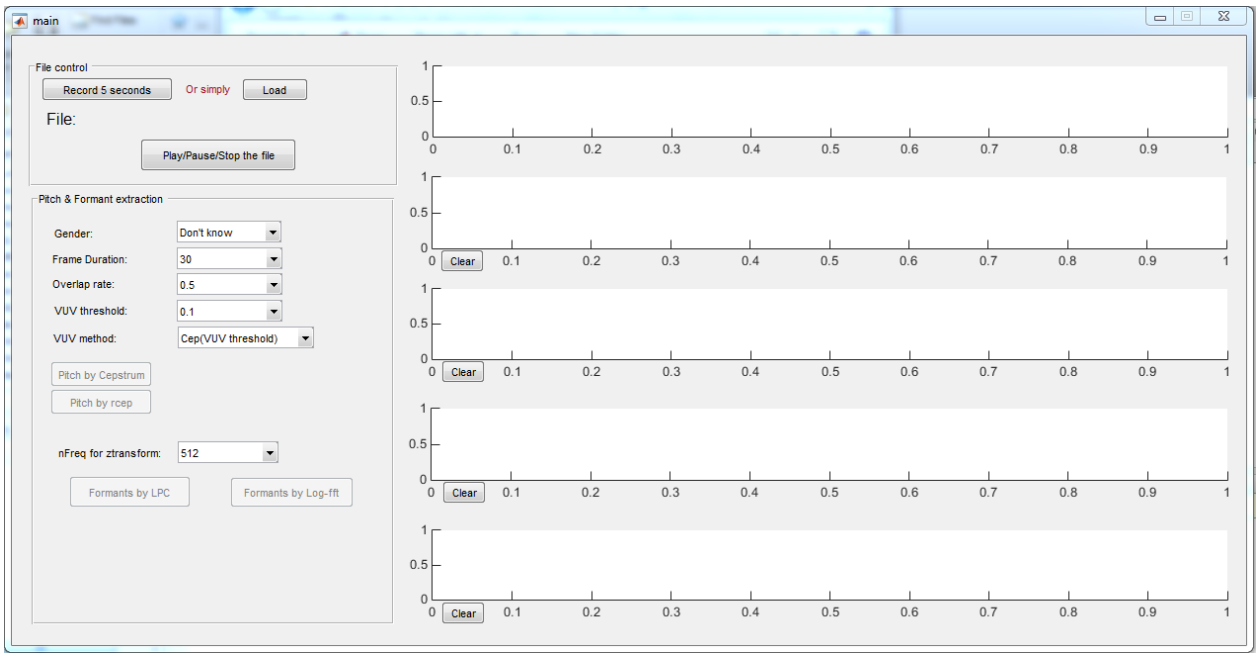

Figure 1

As it can be seen in fig.1, Running the main file, provides two control panels, first of which is to record a wave file and the second is to open a file. Both are loaded into the program area and can be seen as a signal in the first plot. The path is displayed and it is also possible to listen to this audio file.

The second panel is where we can choose the parameters and ask the program to use them in order to show the plot of F0, F1, F2, F3 and the average of these values for the whole signal. The parameters are as follows:

2.1. Gender: is used to confine the search among frequencies more relevant to each gender. The default value is unknown. We can choose male or female if we are sure and want a more precise result.

2.2. Frame duration: The time interval in which we will have a single value for each parameter. This duration should be as long as a short phoneme (10 to 30msec) and we assume that we maintain stationarity in this small interval in time domain. But the smaller this frame, the wider main beam in frequency we will have after applying Fourier transform on the frame. Therefore, less but more important side lobes lead to spectral leakage. Hence, choosing an appropriate frame size is crucial. This frame is later multiplied by a window, so that the frequencies are better obtained.

2.3. Overlap rate: The process is done for each frame, but these frames should overlap so that no information is missed in between. This rate is usually considered 50% of the frame size.

2.4. VUV Threshold: This parameter as an input to the function that detects Voiced frames, is used when the pitch is detected according to the desired frequency range, but this detected f0 may be a false positive, since it may have a magnitude which is not high enough to be a pitch. That is where we should compare this potential F0 with VUV threshold to determine if it is a voiced frame or not. This threshold is experimentally set to 0.1 in most references.

2.5. VUV method: If we don't want to use VUV threshold to tell the difference between Voiced and Unvoiced frames, we can use the second method (ZCR) that is based on a simple computation.

2.6. nFreq: The Z-transform is the main calculation in formant extraction done by LPC method, so the Number of evaluation points has to be adjusted by the user. nFreq is a positive integer scalar no less than 2. N should be set to a value greater than the filter order.

There are four Buttons in the panel that are responsible of performing Homomorphic analysis for pitch and formant extraction. After pressing each button, the call backs run and we can see the resulting plot with the desired frequencies over time. An average of this parameter is also shown.

There are two buttons to calculate pitch but both follow the same steps, the only difference is that the second button uses Matlab rceps command. The reason of its presence is to compare its result with the result of the Cepstrum that is taken step by step in the code.

3. Implementation and the theory behind

3.1. main.m

In the main file, there are call backs of each object on the GUI. To plot the signal, we need time as x axes which is a 0 to Ns vector where Ns is the length of the signal. Before any further process, the average of the signal should be subtracted from it in order to cancel out the DC component. Then the recorded or loaded signal is plotted, and ready to be played by the play function.

The call backs that are responsible to estimate the pitch and formants, call the functions related to these procedures with the appropriate parameters and get the outputs as an array of desired frequency with one element per frame and another array of same size for time to be plotted together. F0 average is also passed to the main program to be displayed.

3.2. pitchcep1.m

Clicking on the first pitch button calls this function. After initializing the variables and arrays, there is a for loop of iterations equal to frame numbers, that computes f0 for each frame, in this loop, another function(cepvuvandf0) is called to provide this frame with the potential F0 value, the decision of Voiced/Unvoiced, and two other values only for plotting the Cepstrum of this frame.

The final F0 is a multiplication of F0 potential and VUV decision. This decision is either the output of cepvuvandf0, or the output of "VUVZCR" function, if it is selected by the user. Calculating the median of 3 adjacent frames of the F0, is done to give a smoother curve with less oscillations. F0 values are summed up in that loop, to take the average. Figure 3 shows the process of finding pitches after calling "pitchcep1" function.

3.3. cepvuvandf0.m

"cepvuvandf0" returns F0 by Cepstrum and VUV as output. Before any operation on frames, a window should be convolved to the framed signal. The square window is the implicit default, but not a good choice due to spectral leakage. The FT of a rectangular window to cut off signal and limit it in a frame is the Sinc function which extends in the window over the entire frequency spectrum. This causes the ripple on either side of the peak. While the FT of a Hamming shaped window concentrates this "splatter" much nearer to the frequency peak after the convolution. So it is better, compared to a Sinc function. The hamming window gives a more accurate frequency spectrum. There are other window functions that can be employed, and there are techniques to choose one of them for each application regarding their properties. The following equation generates the coefficients of a Hamming window:

$$w(n) = \{0.54 - 0.46\ cos(\ 2\pi.\ N.n)\ otherwise\ if\ n \in [0,\ N]$$

Then the process begins with computing Cepstrum which is the inverse of spectrum. In fact it returns the reverse Spectrum (in Quefrency), but not the signal in time domain. The Cepstrum of a voiced frame shows some peaks (similar to the DFT spectrum peaks). These peaks correspond to the fundamental period. The Cepstrum of an unvoiced frame shows no clear peaks. So there's no fundamental period.

So we have to first Calculate the Cepstrum in this short windowed frame, then look for peaks around the origin and at the end compare the peak to a threshold. If this peak is greater than the threshold, this corresponds to the pitch period which should be converted to the frequency.

Since the Cepstrum is the inverse DFT of the log of the amplitude Spectrum of the DFT, to calculate the Complex Cepstrum in this function, the following pipeline is implemented:

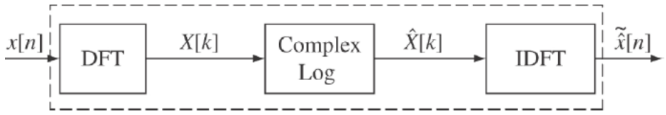

Figure 2

Because STFT can be defined as the DFT of a windowed signal, we use fft command of Matlab.

$x_n(e^{j\omega}) = \sum_{m=-\infty}^{\infty} w(n-m)x(m)e^{-j\omega m} = DFT[x(m)w(n-m)]$

Since the desired frequency is a coefficient of sampling frequency namely 1/Quefrency (The index of the desired frequency), after finding the Maximum frequency, its index that we find is added to the index of the beginning of the search (index of beginning point + indmax), and gives the frequency of the Pitch:

$$f_0 = f_s / (f_0 indmin + indmax)$$

Note that, the disadvantage of the Cepstrum is that it involves computationally-expensive frequency-domain processing.

3.4. VUVZCR.m

Short-time ZCR helps to identify voiced/unvoiced regions typical values. In voiced regions (vowels, voiced fricatives (/b/, /d/, /g/): ZC~14. In unvoiced regions (the rest of consonants): ZC~49. So we can take a threshold: 40 to tell the difference between Voiced and Unvoiced frames. After this simple operation, the VUV value is set to 1 for voiced and to 0 for unvoiced frames. Then it can be used by "pitchcep1.m", if the user has chosen ZCR method in the main figure.

$$Z(n) = [\text{sign}(x(n)) - \text{sign}(x(n-1))]/2N \text{ with } \text{sign}(n) = \begin{cases} 1 \text{ if } x(n) \geq 0 \\ -1 \text{ otherwise} \end{cases}$$

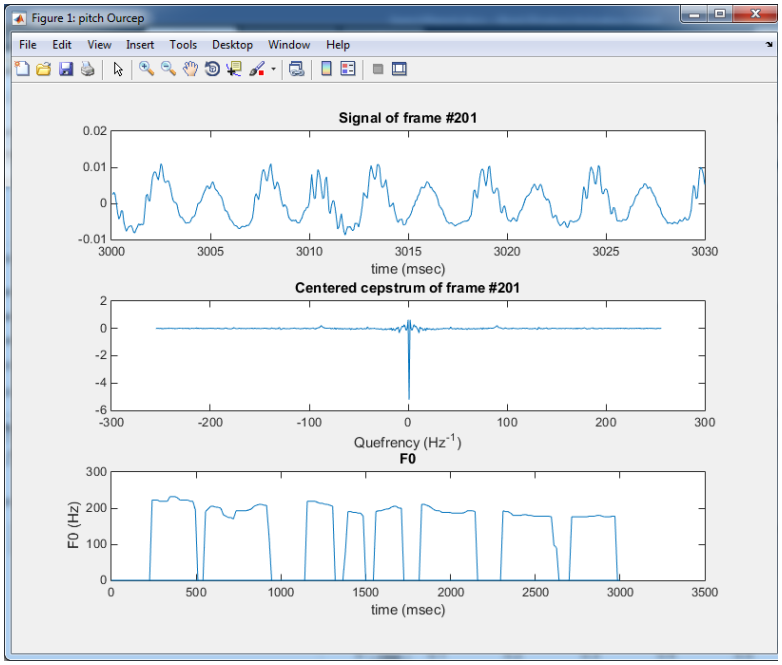

Figure 3

3.5. pitchcep2.m

The process is exactly as the pitchcep1, but the "cep" function is responsible of computing Cepstrum. The ongoing process is shown in Figure 4 and as it can be seen in Figure 6, there is no significant difference between the results.

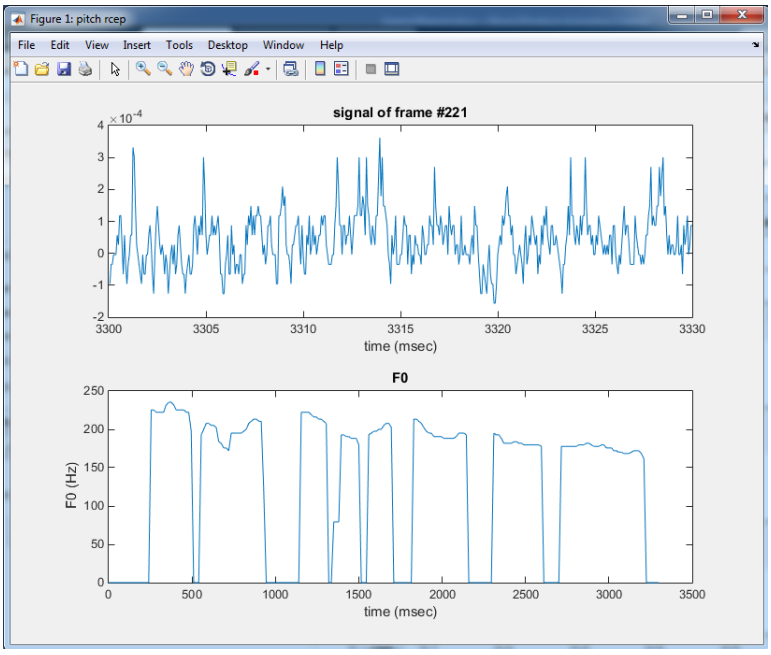

Figure 4

3.6. cep.m

“rceps” command is employed to calculate Cepstrum in cep function. We should again first check for being voiced with the use of “isvoicedthresh” (VUV threshold), as an input argument passed to this function. and then if so, $f_0 = f_s/(f_0 indmin + indmax)$

3.7. formantlpc.m

Formant extraction can be done by LPC method. LPC methods provide extremely accurate estimates of speech parameters, and does it efficiently. In general, Linear predictive models the signal as if it were generated by a signal of minimum energy being passed through a purely-recursive IIR filter. To find the formant frequencies from the filter, we need to find the locations of the resonances that make up the filter. This involves treating the filter coefficients as a polynomial and solving for the roots of the polynomial.

Generally we expect to find ~1 formant per 1000 Hz. So a general rule of thumb is to set the filter order to the sampling rate in kHz plus 2 • 2 for each expected formant, plus two to account for the effects of higher formants and/or the glottal spectrum.

What is done in the function is that a linear filter is defined with this number of coefficients. In Matlab, lpc(x,p) command finds the coefficients of a pth-order linear predictor, an FIR filter that predicts the current value of the real-valued time series x based on past samples. Finding roots of the polynomial is then followed by finding the desired root (only look for roots >0Hz up to fs/2) in the range and convert it to frequency.

Formants should be set to zero in unvoiced frames. To do so, we can call the pitchcep1 function to compute pitch, and then employ VUV or F0 (where it is zero) to be multiplied by the Formants. This is what happens at the last lines of this function before plotting the LP filter together with all three formants.

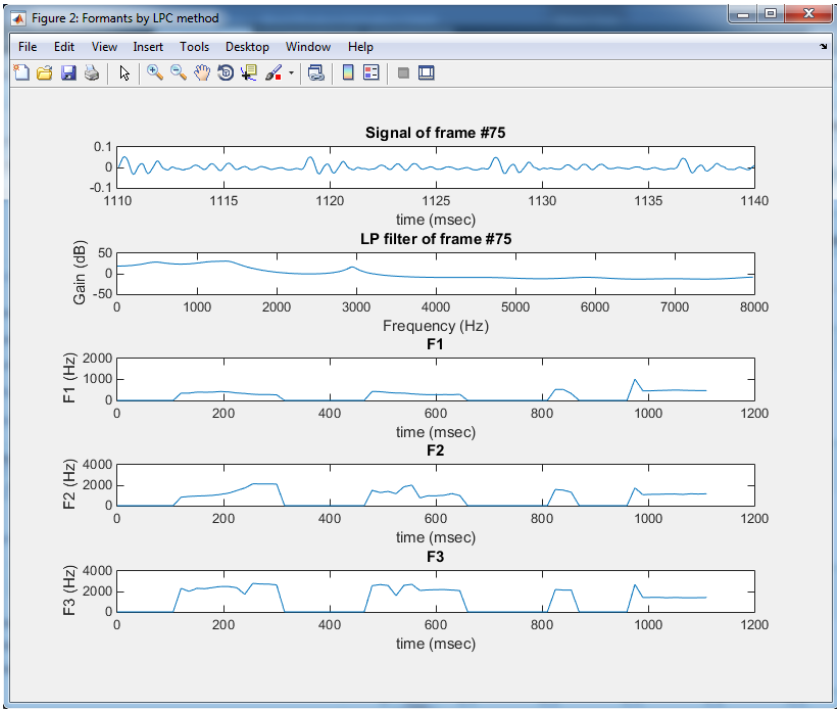

Figure 5

3.8. formantdlm.m

In voiced frames, the envelope of the log spectrum can be obtained by smoothing the log-magnitude of the spectrum of the windowed frame .The peaks of the smoothed Cepstrum correspond to the resonance frequencies of the transfer function of the speech production system. The first peak correspond to F0, where the next 3 peaks correspond to the first 3 formants (F1, F2, and F3).

Implementing this function is very similar to that of pitchcep1, although we have to stop before taking DFT^{-1} (See Figure 2). In this case the output is still in frequency domain and the task is to look for the peaks coming after the first peak (F0), and call them F1, F2, and so on. In this project, this is done by the function named "peakssearch". This function is called with the log of magnitude of DFT of the windowed signal which has been shifted, along with the corresponding frequencies array, and F0. F0 of this frame is passed to peakssearch function in order to enable us to start the search right after it.

Again, Formants are set to zero in unvoiced frames by multiplying them by VUV or F0 (where it is zero)

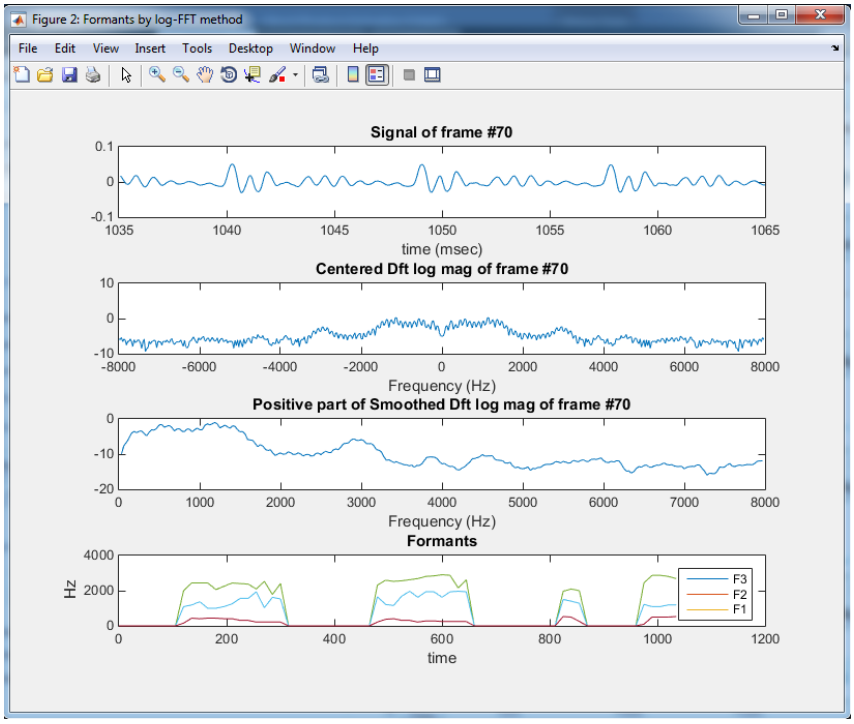

Figure 6

3.9. peakssearch.m

Search process is easy with the use of findpeaks command. If this command is called with the frequencies as the second input, it gives not only the peaks, but also the indices corresponding to these peaks. Hence, we have access to the frequencies accordingly. The idea is to choose an interval of 1000Hz for each Formant, then check for frequencies in these intervals to have their peaks, for each Formant, take the maximum in its range, and again we have index of this maximum that is in fact frequency's index. Note that this is a local index. We have to add the length of our previous search arrays to obtain the Global index, like what we did for pitch index in both pitchcep functions. The last thing to do is to take the frequency of this index.

4. Summary and Results

In this project, we focused on the computation of DFT-Log Spectrum and Cepstrum, by which we are able to detect pitch and formants of a signal. LPC method is also implemented, in order to compare the results. There is slight difference in the estimation of parameters by different methods. Some final results can be seen in Figures 6 to 8. The averages of the parameters is also calculated and can be seen in these Figures.

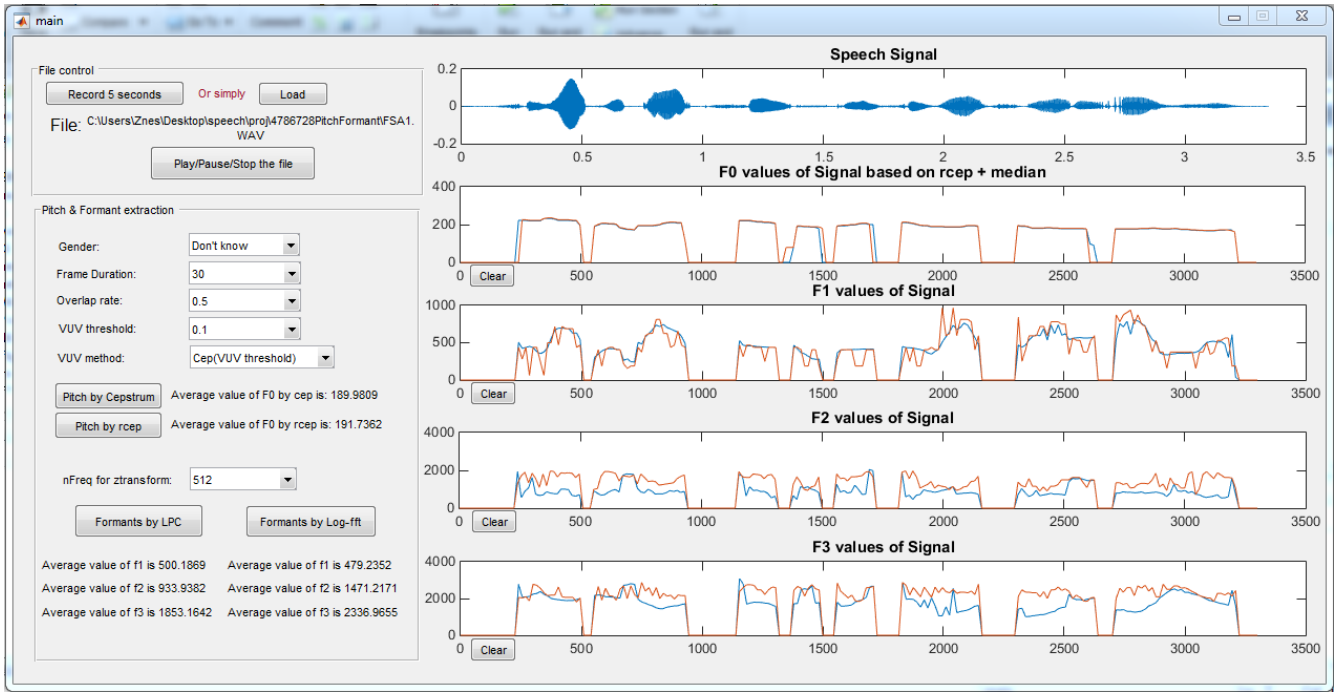

Figure 7

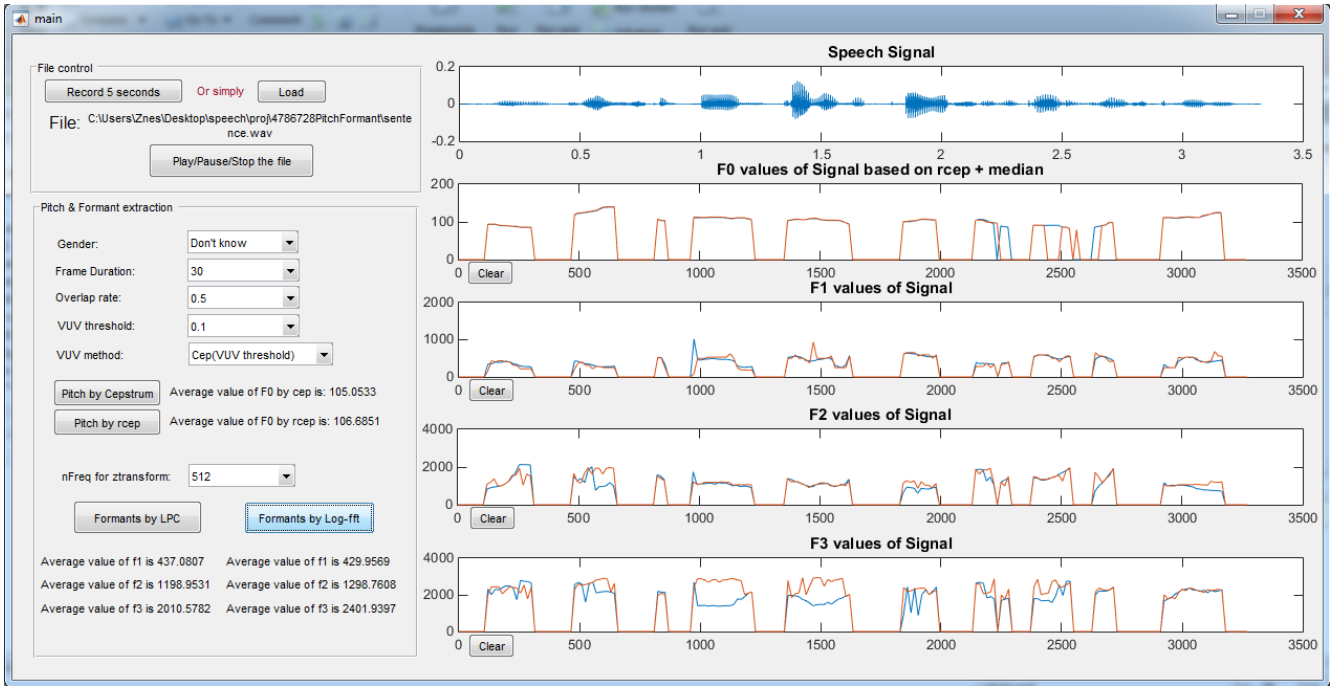

Figure 8

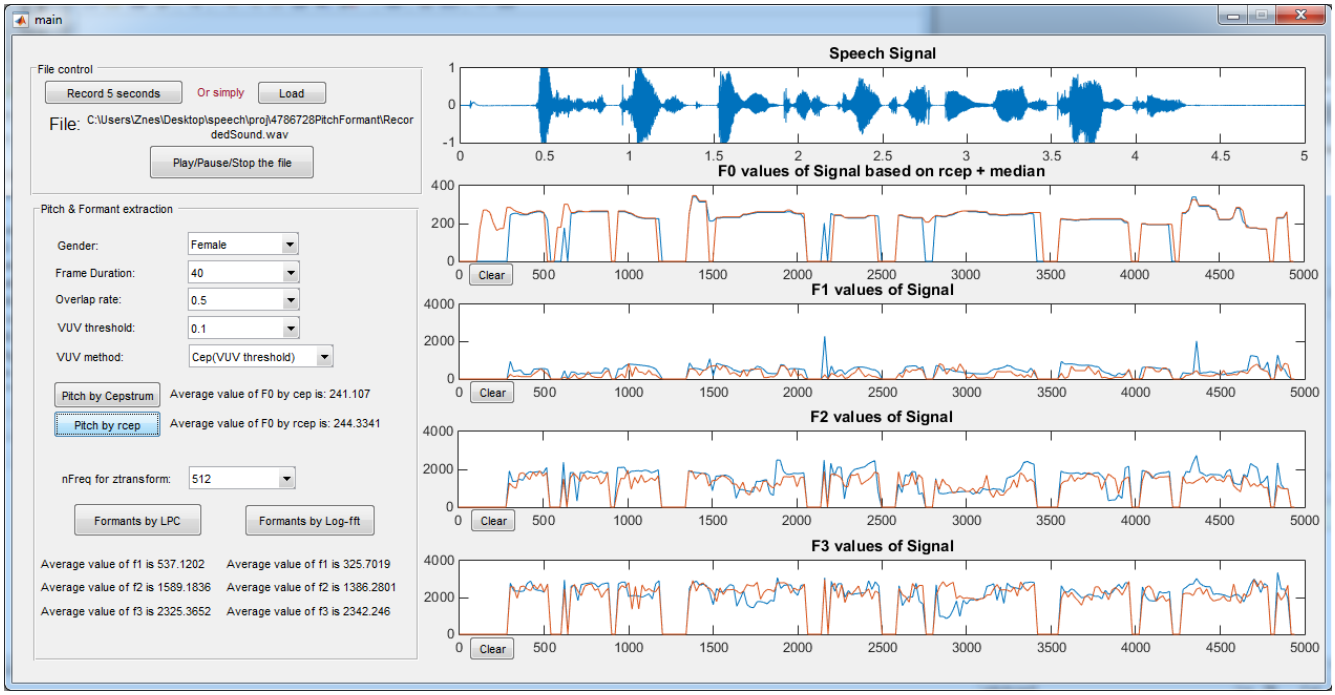

Figure 9

5. References

5.1. https://www.phon.ucl.ac.uk/courses/spsci/dsp/

5.2. https://2019.aulaweb.unige.it/course/view.php?id=5025

5.3. https://www.ece.ucsb.edu/Faculty/Rabiner/ece259/speech%20course.html

5.4. https://it.mathworks.com/

5.5. https://www.phon.ucl.ac.uk/courses/spsci/matlab/lect10.html