Advanced signal processing techniques
Fourth assignment

Theodoros Katzalis
AEM: 9282
`katzalis@auth.gr`

Aristotle University of Thessaloniki

May 31, 2021

1 Cepstrum via Homomorphic filtering

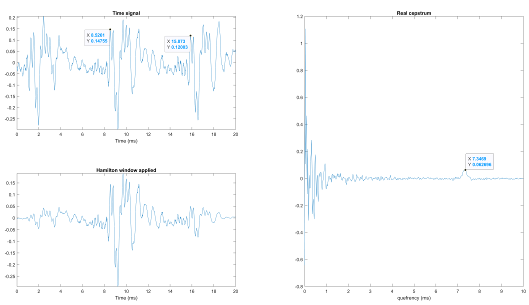
”Acquire voice samples. In this part, please either record or find at least 10 voice samples of a male and female individual making the five vowel sounds – “a”, “e”, “i”, “o”, “u”. If you are going to record them yourself or using a friend, then exaggerate the sounds a little and keep your voice extended for a while. Please make note of the conditions used to obtain the voice samples (e.g., what smartphone, what type of speaker – built-in or microphone, using which software program, or where or from whom the files were obtained). You should have 10 files in the end. If you want to be keen and impressive, you can get more than one male and female voice to obtain a better understanding of differences between both signals in the cepstral domain. Have fun with this.”

2

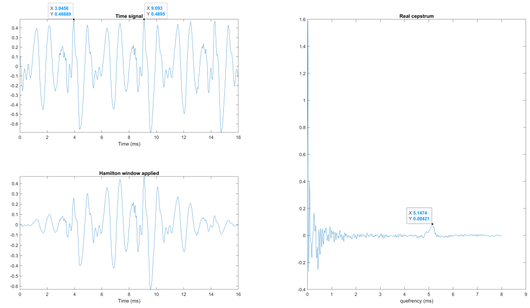
”Compute the cepstrum of each voice signal and discuss any difference qualitatively and quantitatively amongst male and female voices in general and amongst the different vowel sounds. This is an important component of the project, so please be creative and as comprehensive as possible. Your report should provide figures with original time domain signals as well as cepstrum signals. Female voices should generally have more peaks than male voices in the cepstrum domain. You should discuss why you think this would be the case.”

Solution

We have collected 10 voice samples of ”a”, ”e”, ”i”, ”o”, ”u” for one male and one female. We present a window of the voice signal of around 20ms, that is roughly equal to 3-4 pitch periods, along with the application of the Hamilton window, the corresponding real cepstrum, and finally the discrete fourier transform of these samples. For the calculation of the complex and real cepstrum, the `Matlab` functions `cceps` and `rceps` had been used.

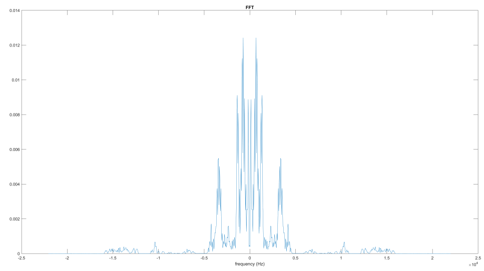


(a) Male Pitch = 7.34ms

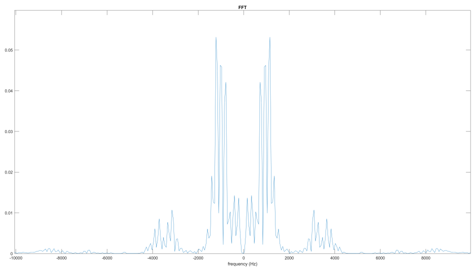


(b) Female Pitch = 5.14ms

Figure 1: Letter 'a'

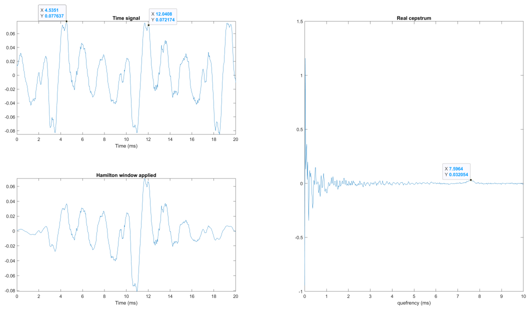


(a) Male

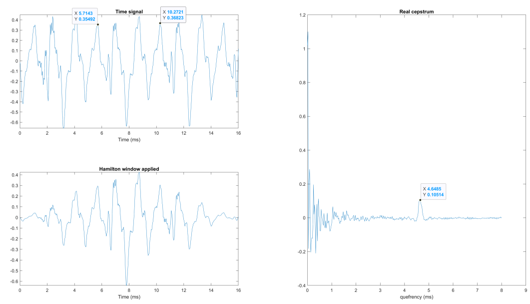


(b) Female

Figure 2: Letter 'a', DFT

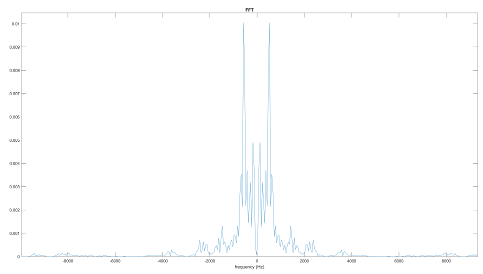


(a) Male Pitch = 7.59ms

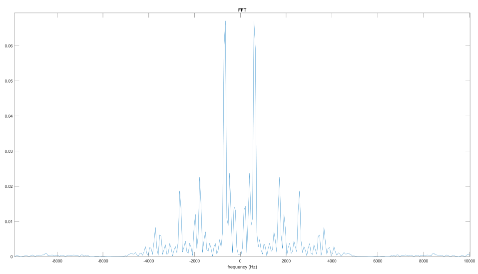


(b) Female Pitch = 4.64ms

Figure 3: Letter 'e'

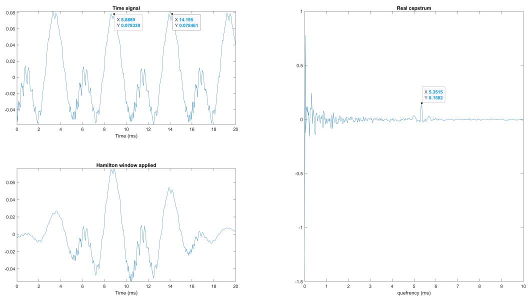


(a) Male

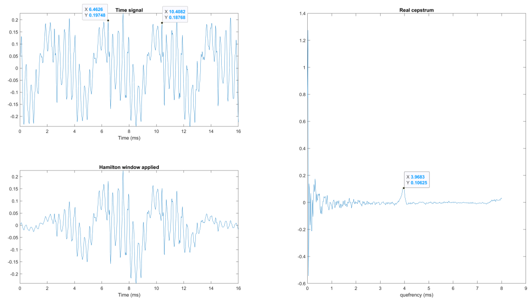


(b) Female

Figure 4: Letter 'e', DFT

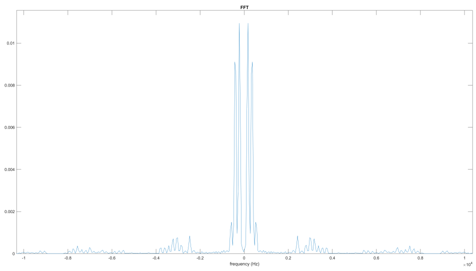


(a) Male Pitch = 5.35ms

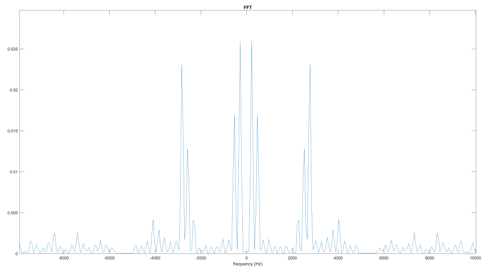


(b) Female Pitch = 3.96ms

Figure 5: Letter 'i'

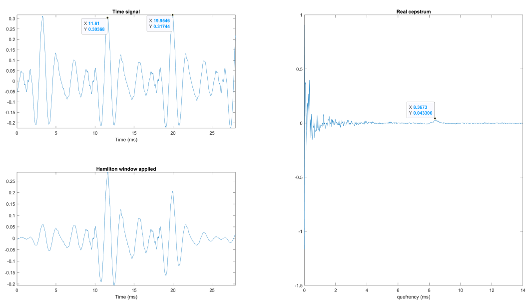


(a) Male

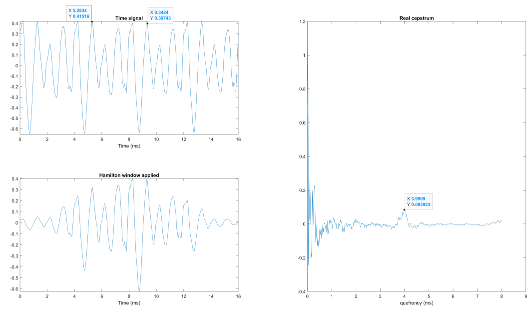


(b) Female

Figure 6: Letter 'i', DFT

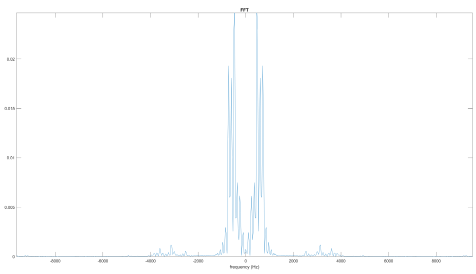


(a) Male Pitch = 8.36ms

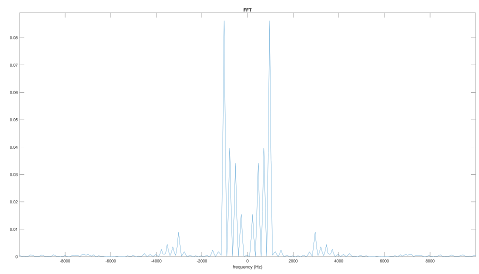


(b) Female Pitch = 4ms

Figure 7: Letter 'o'

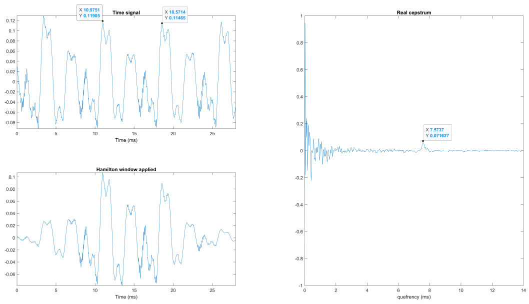


(a) Male

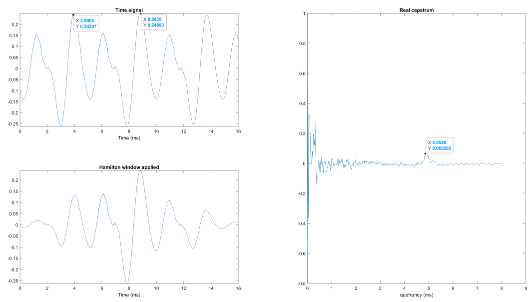


(b) Female

Figure 8: Letter 'o', DFT

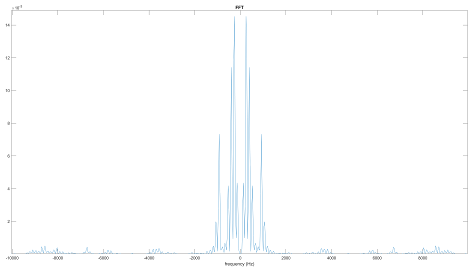


(a) Male Pitch = 7.57ms

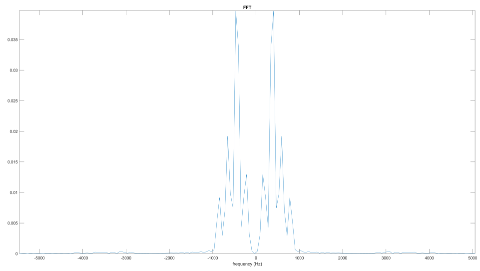


(b) Female Pitch = 4.85ms

Figure 9: Letter 'u'



(a) Male



(b) Female

Figure 10: Letter 'u', DFT

2.1 Conclusions

- We can see clearly the periodicity of the voice signal and how real cepstrum is able to capture the pitch. We can also claim that the real cepstrum compared to the complex spectrum has better performance for pitch detection with sharper peaks.
- It is known that the pitch in females is higher than males. This is determined by the fundamental frequency of the vocal fold vibration. Thus we can observe a shorter time period in the analysis of female vocal sounds and a corresponding spike to appear sooner in quefrency.
- The window of the voice signal should be 2-3 periods, a condition imposed by the window application in order for the approximation $s[n] = w[n](h[n] \circledast p[n]) \xrightarrow{cepstrum} \hat{s}[n] = \hat{p}[n] + \hat{h}[n]$ to hold, where $w[n]$ the window function, $p[n]$ excitation and $h[n]$ impulse response.
- The pitch between letters doesn't change significantly.

3

"Lifter the cepstrum domain signals. Design a window (length is an important design parameter and you should discuss how and what you select – it can be the same or different for each speech sample depending on what you would like to experiment with) to remove the transfer function dependency. Then, compute the time domain signal of the corresponding windowed result to obtain the deconvolved signal. Plot the deconvolved result. Is there anything you can say about the signal and its difference from the original time domain recorded sample? Again, your discussion is an important part of the report."

Solution

The following analysis is based to a specific sample of the ten total, the voice sound of 'o' female. For speech **deconvolution** with the aim of the **reconstruction**, we used the complex spectrum of the windowed voice signal. About liftering, we chose a rectangular pulse with size equal to `2*pitchEstimation`. The `pitchEstimation` is determined by the location of the first peak of the real cepstrum, that determines the cepstrum of the impulse response that we want to remove, making use of `Matlab` function `findpeaks()`. We can see visually in the following figures the application of the lifter in the complex cepstrum and the final outcome.

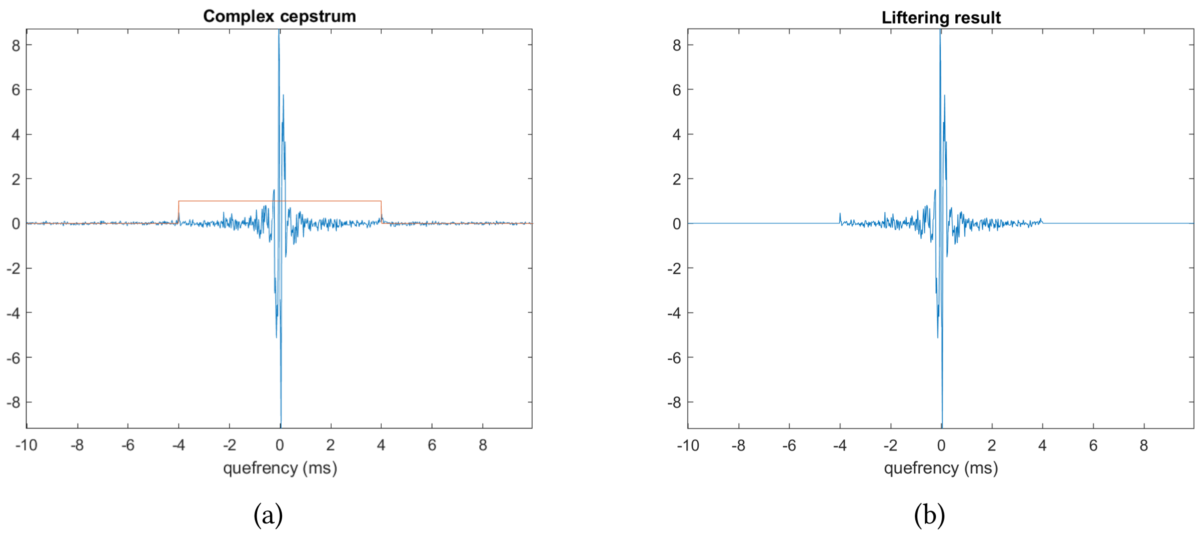


Figure 11: Letter 'o', DFT

After liftering, we inversed the complex cepstrum using `Matlab` function `icceps` and we obtained an estimation of the impulse response. We can observe that the obtained signal is similar with the fundamental portion of the original time signal that repeats itself.

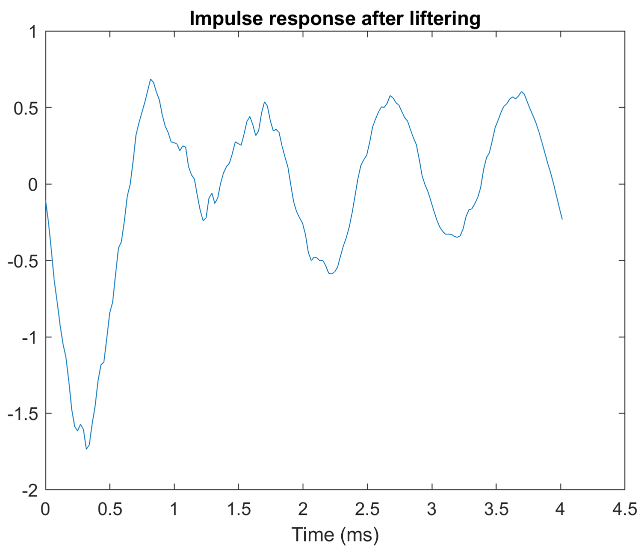


Figure 12: Impulse response estimate invresing the liftered complex cepstrum

4

"Try to synthesize back the voiced signals as follows and comment on your results."

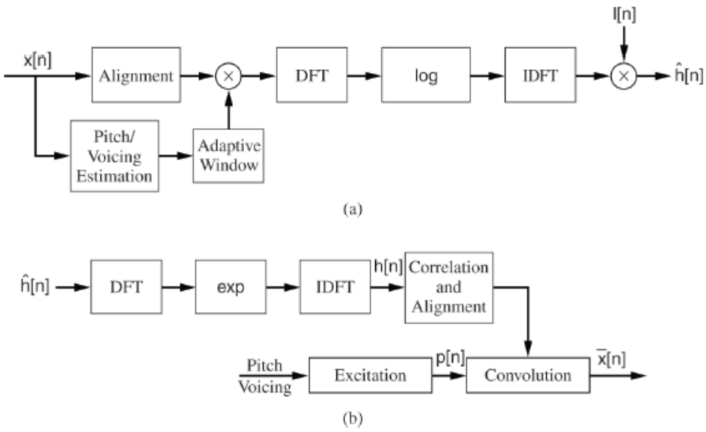
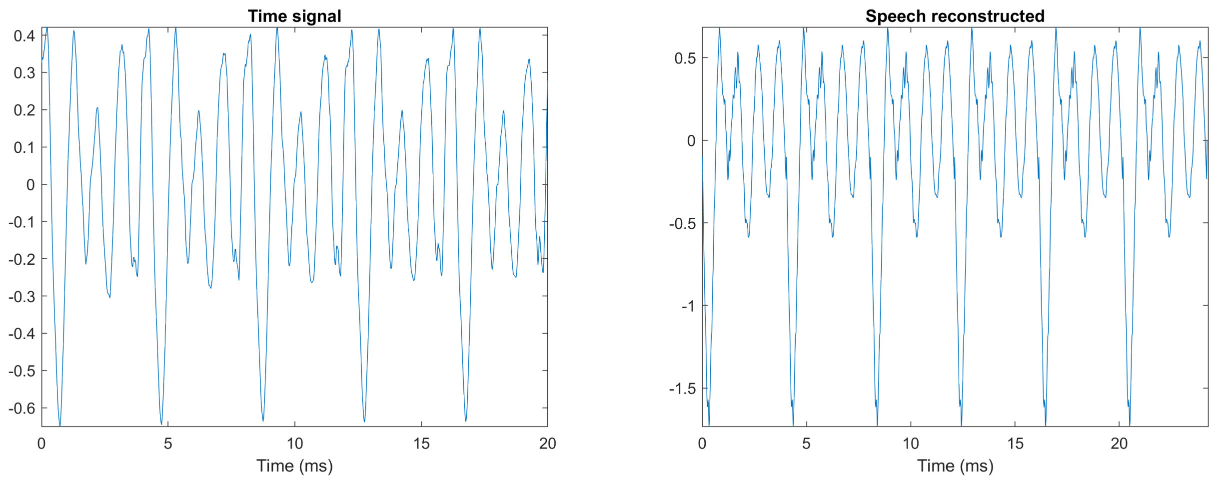


Figure 13: Speech reconstruction methodology

Using the convolution of an impulse train with the previous estimated impulse response, we obtained the following reconstructed voice signal which resembles adequately the original one.



(a) (b)

Figure 14: Original vs Reconstructed voice signal

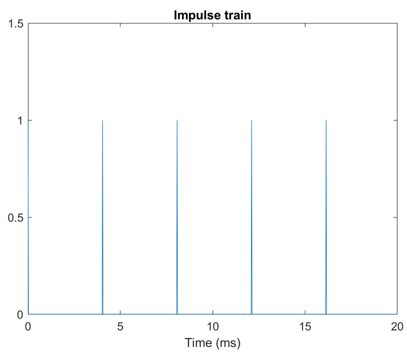


Figure 15: Impulse train

Although it should be noted that the impulse train can be estimated using a high pass lifter in the complex cepstrum analogous to the lowpass used for the impulse response.

5 Matlab

All the `Matlab` files can be found here.