Signal Processing

Lab 2 – Speech processing

Pôle

Numérique

Author	: Guillaume GIBERT
Version	: 1 R 0
Promotion	: 2026
Program	: EENG3
Date	: March 24th 2024

1. Introduction

The goals of this lab are:

- To use the Fourier transform to analyse the spectral content of a signal;
- To use Time-frequency plots (e.g., spectrogram) to analyse the spectral content of a non-stationary signal;
- To study the effects of (down-)sampling on a signal;
- To experiment the filtering differences of a FIR (Finite Impulse Response) *vs* IIR (Infinite Impulse Response) filters;
- To create a channel vocoder to modify speech sound and create a robotic voice;
- To practise coding in Octave/Matlab;
- To control the versioning of a software using git/gitflow.

To do so, the following equipment is provided:

- A laptop running a Linux OS;
- A headset with a microphone.

2. Setup

For this lab, you will be using PC running a Linux OS (Ubuntu). However, the same code should work on Windows or Mac OS as you will be using Octave which is a scripted programming language available on multiple platforms.

If the signal package is not installed yet, open a terminal and type:

```
$ sudo apt-get install octave-signal
```

Please create a directory EENG in the home directory: /home/ros/EENG if it is not already the case.

Please write your code in this directory.

3. Speech processing

Producing speech is a complex phenomenon involving several synchronized physical activities [1]:

- To contract lungs to push air out through the oral and nasal cavities;
- To generate oscillations of vocal folds to produce voice sounds (e.g., vowels);
- To modify the shape of the vocal and nasal cavities using tongue, velum, lip and jaw movements.

Speech signal is a non-stationary kind of signal: its spectral characteristics vary over time (see Figure 2). Indeed, it is composed of periodic and non-periodic parts. The periodic parts correspond to the air coming from the lungs modulated by the vocal cords. The oscillations of the vocal folds create a physical phenomenon which is perceived as the speaker's pitch. This pitch is in the range of [80-160] Hz for men in general whereas it is in the range of [160-255] Hz for women. Children have a higher pitch in the range of [300 -450] Hz. The vocal tract is shaped by the position and movements of the tongue, velum, lip and jaw (see Figure 1). Depending on this shape, the resonances and anti-resonances of the acoustic space will be modified, thus generating different speech sounds by enhancing or reducing some frequencies.

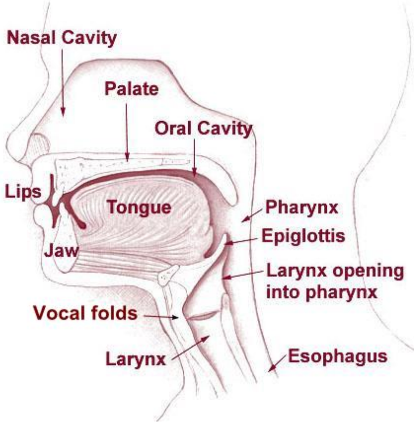


Figure 1 : Side view of the speech production organs (from [1]).

Vowels sounds have a higher energy than consonant sounds (see Figure 2) since there is no obstruction in the air flow. Consonants, on the contrary, are characterized by a partial or a full obstruction of some parts of the vocal tract. These obstructions generate turbulences leading to impulse and/or noise-like characteristics.

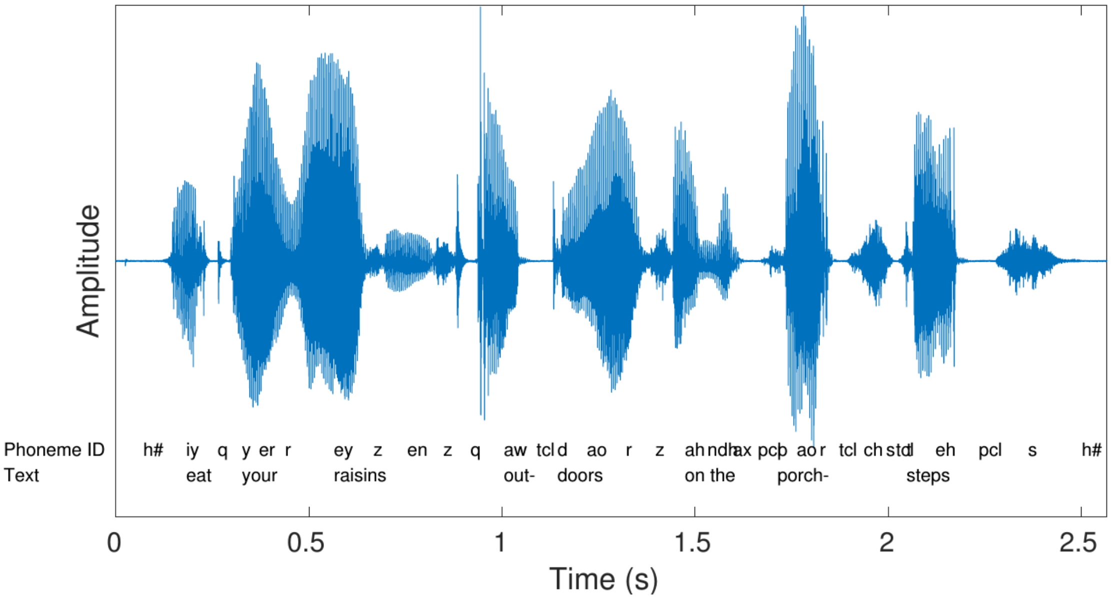
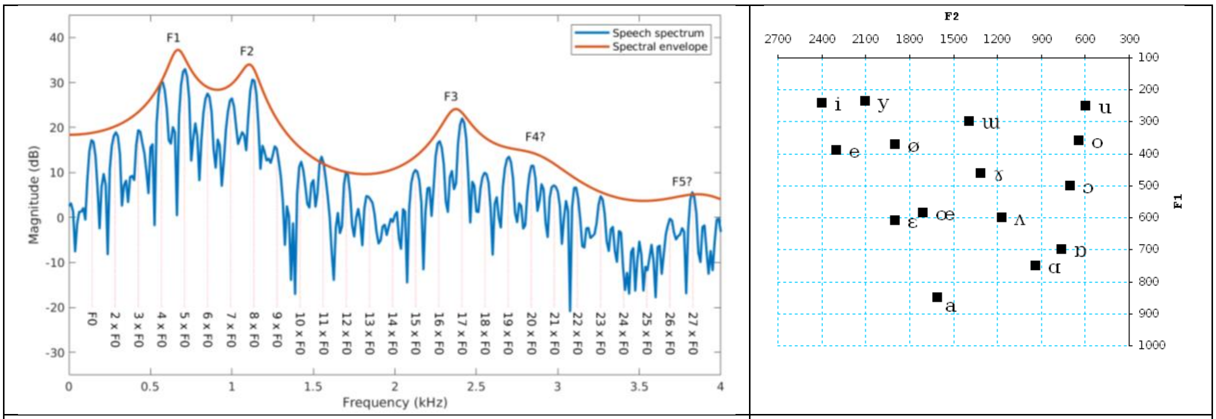


Figure 2: Variation of the amplitude of a speech signal over time for the pronounced sentence "eat your raisins outdoors on the porch steps" (from [1]).

Sounds perceived by humans vary in the frequency range of [20-20000] Hz. However, speech signals covers frequencies from 30 to 10000 Hz, most of the energy being in the range [200 – 3500] Hz [2].

The three main characteristics of a speech signal are:

- The **fundamental frequency** entitled **F0**. If it is present, the current speech sound (phoneme) is said to be voiced. If it is absent (i.e. vocal folds are not oscillating) the speech sound (phoneme) is said to be unvoiced.
- The signal **amplitude**. Vowels have higher energy than consonants, thus higher amplitude on the temporal trace. Plosive consonants such as /p/ are characterized by a short period of silence (amplitude is quasi null because air flow is stopped) followed by a burst of energy (air flow is released).
- The **resonances of the vocal tract** entitled **formants** (**F1**, **F2**, **F3**...). The formants are the broad spectral maxima (see Figure 3a) that result from the acoustic resonance of the vocal tract [3]. The first two formants can be used to determine the pronounced vowels using the "vowel quadrilateral" (see Figure 3) [4].



(a) Power spectrum of a vowel sound (from [1])

(b) Average vowel quadrilateral for English (from [3])

Figure 3 : Formants correspond to broad spectral maxima (a). A diagram of average vowel formants (b) along F1 and F2 allows separating the vowels.

4. Exercises

4.1. Setup

4.1.1. Git repository

Open a web browser at this address: https://gitarero.ecam.fr/user/login

Log in with your Ecam credentials

Create a new git repository entitled SignalLab2 (keep the default configuration)

Open a terminal

Go to the target directory using the *cd* command

Type the following command to clone the git repository you have just created:

```
$ git clone https://gitarero.ecam.fr/[username]/SignalLab2.git
```

Update the description of the project "Speech processing" in the README.md file and create a commit

Create a develop branch

4.1.2. Octave code

Create a file entitled speech_analysis.m (using a text editor e.g. scite, vim, gedit…) that loads the signal package, using the following command:

```
pkg load signal
```

Create a commit

4.2. Speech analysis

Load a speech signal using the **audioread()** function (see section 6 for the link to Octave documentation).

An example of speech signal modulator22.wav is available on Moodle.

Save the speech signal you just loaded modifying the sampling frequency information (half of the current one for instance) using the **audiowrite()** function. Listen to the generated sound.

Q. What is the effect of modifying the sampling rate information and keeping the same signal?

4.2.1. Temporal analysis

Create a plot of the temporal variations of the signal amplitude.

Do not forget to provide axis labels with units!

Q. Is this signal stationary?

Q. If not, what is the typical duration on which you may consider it as stationary?

4.2.2. Spectral analysis

Compute the power spectrum of the signal using the Discrete Fourier Transform (DFT). You may use the provided **frequencySpectrum()** function.

Q. What information(s) on the signal can you retrieve from this plot?

Compute the power spectrum of the signal using the Fast Fourier Transform (FFT). You may use the provided **frequencySpectrum()** function.

Q. What parameter(s) should you modify to apply a FFT instead of a DFT? Please explain.

Estimate the duration to compute the FFT for this speech signal.

Compare it with the duration to compute the DFT for the same signal. You may want to repeat this measurement several times to estimate the averages and standard deviations.

Q. Is the FFT faster than the DFT? Please explain.

Compute and display the spectrogram of this speech signal using the provided **spectrogram()** function which is based on the **specgram()** function from the **signal** package (see section 6 for the link to the documentation).

Set the parameter window_size to 30 ms and step_size to 5 ms.

Q. What is a spectrogram? Please explain.

Q. What are the differences with a DFT? Please explain.

Set the parameter window_size to 5 ms and step_size to 5 ms.

Q. What is the impact of the window_size parameter? Please explain.

Q. What would be the optimal value for speech analysis? Please explain.

The phonetic transcription of the first cardinal numbers is:

- **one** wʌn
- **two** tu:
- **three** θri:

Select the speech signals (i.e. determine the start and end samples) that correspond to the vowels /ʌ/, /uː/ and /iː/ of the cardinal numbers 1, 2, 3.

Compute and display the power spectrum of these signals.

Determine the fundamental frequency F0 and also the formants F1 and F2 for these vowels and compare them to the average vowel quadrilateral.

You may want to apply a low-pass filter to the power spectrum to retrieve the envelope and determine the formants frequencies.

Q. What is the value of this speaker's speech?

Q. Are the formants placed at the same place as the average one?

Q. Can you easily categorize the three vowels based on F1 and F2?

4.2.3. Downsampling

Use the function **downsample()** to downsample the speech signal to a sampling frequency equal to 4000 Hz.

Use the function **decimate()** to downsample the speech signal to a sampling frequency equal to 4000 Hz.

Compare the output signals (temporal variation and also sound quality). You can generate an audio file with the function **audiowrite()**. The audio file (.wav) can be played with the following command:

```
$ aplay <audiofile.wav>
```

Q. What is the difference between the two functions?

Q. What are the distortions generated by the downsample() function? Please explain.

Implement a low-pass FIR filter of order 30 and cut-off frequency equal to1000 Hz. You may want to use the function **fir1()**.

Implement a low-pass IIR filter (Butterworth) of order 8 and cut-off frequency equal to1000 Hz. You may want to use the function **butter()**.

Check the stability for both filters.

Q. How do you check the stability?

Q. Are the filters stable? Please explain

Display the frequency response (magnitude and phase) of the FIR and IIR filters using the **freqz()** function.

Apply the FIR and IIR filters on the original signal.

Compare the output signals (temporal variation and also sound quality).

Use the function **downsample()** to downsample the filtered speech signals to a sampling frequency equal to 4000 Hz.

Compare the output signals (temporal variation and also sound quality).

Q. What can you conclude?

Create a commit.

Push it to the remote repository.

Merge your code into the main branch.

Push it to the remote repository.

Check the graph of the branches using the following command:

```
$git log  --graph
```

N.B.: You could repeat the same signal processing techniques on your own speech sample and compare the results.

4.3. Speech processing

The aim of this last section is to create a channel vocoder to modify the speech sound so it becomes more "robotic-like". To do so, three signal processing techniques (that you already know) must be used: spectral analysis, filtering and modulation. An Octave/Matlab function entitled chanvocoder.m was developed by William Sethares (https://sethares.engr.wisc.edu/vocoders/channelvocoder.html). This function takes two audio files: one is considered as the modulator (i.e. voice) and another one as the carrier (i.e. synthesizer, noise). Three parameters can be adjusted: the number of channels (half of the length of the FFT), the number of bands and the window overlap (between successive FFT frames). Sound files are provided, they were taken from https://github.com/borsboom/vocoder which is a C implementation of the channel vocoder.

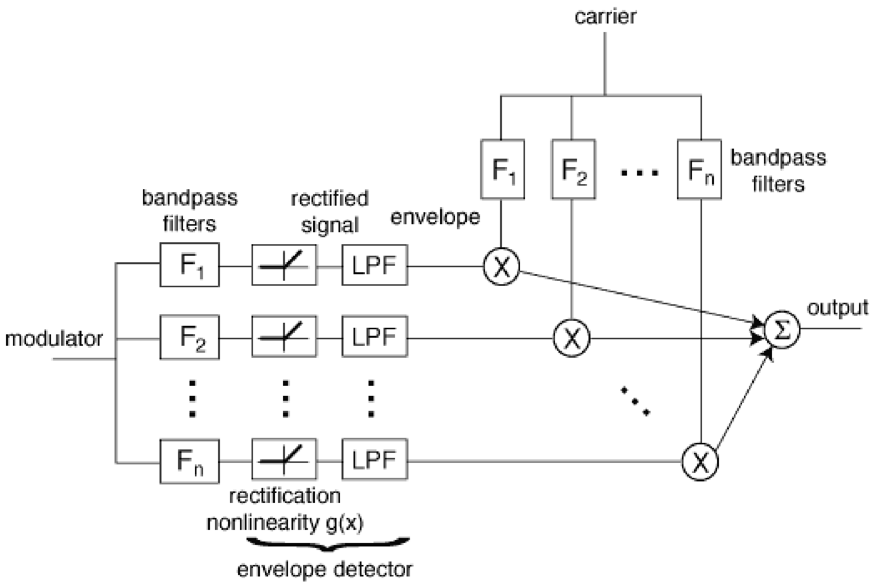


Figure 4 : Channel vocoder corresponds to a series of band pass filters applied to two signals: modulator and carrier. The envelope of the modulator (for each frequency band) is extracted thanks to a low-pass filter. Then, the envelope is imposed to the carrier waveform (from [5]).

Move to the develop branch.

Create a file entitled vocoder.m (using a text editor e.g. scite, vim, gedit...) that loads the signal package, using the following command:

pkg load signal

Create a commit.

Test the **chanvocoder()** function as explained in :

https://sethares.engr.wisc.edu/vocoders/channelvocoder.html

Plot on the same figure the temporal variations of the modulator, the carrier and the output signals. You may want to use the function **subplot().**

Compute the spectrogram of the two input and the ouput signals and compare them.

Test it with different carriers: synthesizer, white noise, periodic white noise.

Q. Based on the different spectrograms, could you explain the effects of the channel vocoder on speech sounds?

Q. What are the effects of the three parameters: number of channels, number of bands and window overlap?

Instead of applying the chanvocoder() function on recorded samples, your task will be to grab some sound signals from the microphone using the **record()** function and feed the **chanvocoder()** function with this "modulator" signal. Finally use the **audioplayer()** function to play the generated speech sound.

Plug the headset and microphone in the laptop jacks sockets and test your function.

5. Evaluation

A report should be submitted on Moodle by:

- March 31st 2024 11.59pm for Group E;
- April 3rd 2024 11.59pm for Group F.

The report should be submitted as a pdf file generated from the provided Word template. This report must contain the following sections: Methods, Results, Discussion.

Tips:

- The report is not a diary;
- Do not list a series of questions/answers;
- Check the axis labels of your figures (do not forget units!);
- Provide references (even for ChatGPT);
- State which parts were written by a bot.

6. Links

https://speechprocessingbook.aalto.fi/index.html

https://octave.sourceforge.io/list_functions.php?sort=alphabetic

https://octave.sourceforge.io/signal/overview.html

https://sethares.engr.wisc.edu/vocoders/channelvocoder.html

https://github.com/borsboom/vocoder

7. References

[1] T. Bäckström, O. Räsänen, A. Zewoudie, P. Pérez Zarazaga, L. Koivusalo, S. Das, E. Gómez Mellado, M. Bouafif Mansali, D. Ramos, S. Kadiri et P. Alku, Introduction to Speech Processing, 2 éd., 2022.

[2] Wikipedia, «Wikipedia,» December 2022. [En ligne]. Available: https://en.wikipedia.org/wiki/Voice_frequency. [Accès le March 2023].

[3] Wikipedia, «Wikipedia,» November 2022. [En ligne]. Available: https://en.wikipedia.org/wiki/Formant. [Accès le March 2023].

[4] EduHK, «Praat Manual,» 2023. [En ligne]. Available: https://corpus.eduhk.hk/english_pronunciation/index.php/2-2-formants-of-vowels/. [Accès le March 2023].

[5] A. A. Sethares, «Channel vocoder,» [En ligne]. Available: https://sethares.engr.wisc.edu/vocoders/channelvocoder.html.

[6] P. Ladefoged, «American English,» chez *Handbook of the International Phonetic Association*, Cambridge, Cambridge University Press, pp. 41-44.