Acoustic Engineering
Seminar 2: Sound analysis and synthesis

Bart Bijnens - bart.bijnens@upf.edu

Adriana Modrego - adriana.modrego@upf.edu

Genís Plaja - genis.plaja@upf.edu

Zat Pros - zat.pros01@estudiant.upf.edu

October 2024

1 Introduction

In this second seminar, we will go a bit deeper in the spectral analysis topic and tackle two objectives:

- Clearly understand what information is contained in the spectral representation of sound, and why it is useful for further tasks and operations. **Make sure you have a complete understanding of both spectrum and spectrogram and the information contained in them.** Answer and discuss the following questions:
  - Are your sounds harmonic or inharmonic?
  - **Chose an audio signal with multiple notes.** Can you visualise these in the spectrogram?
  - **What will happen if you filter out or set to value 0 the frequencies above a certain threshold? Let's say, above 2kHz.**
- Perform sound synthesis by capturing and reproducing the most salient spectral components of our sound signals.

2 Analysis and synthesis of a sound

As already discussed in the theory lectures, the Fourier Transform disentangles our sound as a combination of several sinusoids. Every sound can be represented as a sum of sinusoids oscillating at different frequencies. Having this information at hand, we may create an easy (and extremely simple) synthesizer inspired on real sound examples.

For this experiment we will select audio signals containing a single note played by instruments of the chosen family. The pipeline will work as follows:

- We start by implementing a function to easily create a sinusoid. Inputs should be amplitude, frequency, sampling frequency, and phase phi.

- We perform analysis of the audio signals containing a single note. Since we only have a single note, and the audios are going to be short, we can use the spectrum.
- From the spectrum, we can find the principal **harmonics** of our sound. Find and note down the frequency of the first 8-10 harmonics and their relative amplitude.
- You may then simulate the sound by artificially generating a sinusoid per each harmonic having the amplitude and frequency at hand. You may ignore the phase for this experiment. Create a function to do that. **Hint:** *You know the relative distance between frequency values and the relative difference in amplitude, so you should be able to create a function that receives a frequency value as input (a note), and generate the synthesized sound for that note.*
- Compare and discuss the resulting sound with the original one. What is missing? What are the characteristics of the missing aspects? Describe clearly the generated sound and why it sounds the way it sounds.

One could argue that you have a synthesizer now! **Can you take an audio of your original source and copy the same melody with the fake instrument you synthesized? Now plot the two waveforms and spectrograms and compare.**

3 Submission

Seminar submissions are always due one week after the class. You should submit a **.ipynb Jupyter notebook** file with the implemented code and experiments. For the submission of Seminar 2, we expect these aspects to be implemented and discussed:

- Analysis and discussion of a single-note sound.
- A working algorithm to synthesize single notes of the analyzed instrument using the analyzed and manually extracted peaks.
- Synthesize a similar melody as an original one from a downloaded example from your sound family. Discussion on the difference between the original and the synthesized sound signal.

Make sure all the code you write and execute works. No need to extensively comment or test the code, but make sure to discuss all the requested aspects and contents. Import all necessary dependencies and remember that **Jupyter notebooks store the variables until you close or you kill the kernel**. Make sure you don't keep unwanted values to important variables. Please, **do submit the notebook in Aula Global with your surnames in the filename,** e.g. plaja_modrego_seminar2.ipynb.