Computational Physics III (PHYS-332) 2019–2020
O. Yazyev, K. Cernevics, Y. Guan, J. Colbois, J. Felisaz

Exercise sessions 2–3: Discrete Fourier transform. Spectral and cepstral analysis of an audio signal

Exercise 1: Discrete Fourier Transform [10 pts]

Files: `test.m`

The Fourier transform relates real and reciprocal space representations of a function. While the conventional Fourier transform operatres with continuous functions $f(x)$, the discrete Fourider transform (DFT) operates with discretized data f_n, $n \in \mathbb{N}$.

1. Write a matlab function which implements the discrete Fourier transform (DFT) as explained in the lecture. Please follow the conventions on naming and the types of input/output data.
2. Verify your DFT implementation by transforming simple model functions and comparing your results with MatLab's function `fft`. See supplied script test.m for examples. Make sure that your implementation handles correctly different possible input arrays.

Listing 1: 'mydft.m'

```
function result=mydft(input_array)

% mydft computes the discrete Fourier transform.
%
% Arguments:
%
%     input_array (1D array complex): data to transform;
%
% Returns: 1D array complex, transformed data of the same shape as
% an input array.

% Following is a template. This should be replaced with your own code.
result = dft(input_array);

end
```

Submit: file `mydft.m` with your implementation of the DFT.

Exercise 2: Spectral and cepstral analysis [20 pts]

Files: `audio1.wav, audio2.wav, audio3.wav, audio4.wav`

With the advent of deep learning, speech recognition techniques have become widespread and are used by the general public daily. One of the easiest speech recognition technique, however, doesn't require deep learning. It simply relies on computing the power *ceps*trum of a signal, and at the heart of this analysis, one needs to use Fourier transforms.

Initially introduced by Bogert et al. [1] to detect echo, the power cepstrum of a signal is defined as

$$\mathcal{C}(f) = \left| \mathcal{F}^{-1} \left\{ \ln \left(\left| \mathcal{F} \left\{ f(t) \right\} \right|^2 \right) \right\} \right|^2 \tag{1}$$

where $\mathcal{F}$ stands for the Fourier transform, and where $f(t)$ is a signal given as a function of time. Taking the Fourier transform of a signal, one can see what are the excitation frequencies that compose it. Typically, there will be a fundamental frequency (the lowest frequency for which the Fourier transform of the signal has a peak), and a few peaks at higher frequencies. The frequencies at which these peaks occur are called the harmonics. The idea is that the amplitude at which the various harmonics are excited is as well a strong signature of the signal (this is what musicians call the timbre). Looking at the amplitude of the harmonics can therefore give some new information about the signal; the cepstrum does this by taking the log of the signal then taking the inverse Fourier transform $\mathcal{F}^{-1}$.

The cepstrum can as well be applied in speech recognition, where the initial signal is a convolution of the effects of the vocal excitations (or pitch) and of the phonems composing the words. In Fourier space, this convolution will result in a product of the two spectra. Taking the logarithm makes these components additive, and the inverse Fourier transform pushes them to different parts of the *quefre*ncy domain. This is the source of the application of the cepstrum for pitch detection.

In this exercise, you are going to implement a function that computes the power cepstrum of a given signal, and apply it to a small range of problems.

1. Inverse Fourier transform and cepstrum

   (a) Write a function `mydftinverse.m` which performs the inverse Fourier transform of a signal.

   (b) Check it by taking the inverse Fourier transform of the Fourier transform of a random set of numbers.

   (c) Write a function `mypowercepstrum.m` which computes the power cepstrum of a signal.

   (d) The real cepstrum is defined as

   $$\mathcal{RC}(f) = \mathcal{F}^{-1}\left\{\ln\left(\left|\mathcal{F}\left\{f(t)\right\}\right|\right)\right\}. \tag{2}$$

   The real cepstrum is related to the power cepstrum by

   $$4(\mathcal{RC})^2 = \mathcal{C} \tag{3}$$

   Use the `rceps` function from Matlab to compute the real cepstrum and check your function `mypowercepstrum.m`.

2. Spectral analysis of a signal

   (a) Load the audio file `audio1.wav` using the MatLabs `audioread` function.

   (b) You can play the signal using the MatLab `sound` function.

   (c) Plot the signal as a function of time. For this, you will need to understand the output of the audioread function. Hint: you can type `help audioread` in the command window to see how to use the function. (The comments at the beginning of any function you write will be displayed in this way as well. It is a good habit to take to document your code.)

   (d) Compute the Fourier transform of the signal. Plot the absolute value of the Fourier transform as a function of the frequency, and find the fundamental frequency.

   (e) To verify your guess, generate a sinusoidal signal `sincheck`

   $$x(t) = 0.25\sin(2\pi f t) \tag{4}$$

   with the frequency from the previous point. Play it to verify that you get the same sound. Plot the absolute value of its spectrum agains the one of the `audio1.wav` file and show that the fundamental frequencies match.

   (f) Perform the same exercise on `audio2.wav`, `audio3.wav` and `audio4.wav` and comment on the fundamental frequencies of each recording.

3. Cepstral analysis of a signal

   (a) Load the audio files `audio1.wav` and `audio4.wav`.

   (b) For both files, compute and plot the log of the absolute value of the Fourier transform.

   (c) Compute the power cepstrum for both cases, and plot them on the same graph. You can renormalize one of them to make them comparable. For readability, only plot the quefrencies from $2 \cdot 10^{-4}$ to 0.016.

   (d) What are the quefrencies units?

   (e) Comment on the differences between the two quefrencies spectra and how they could be interpreted.

Submit: your report as a PDF file as well as your scripts. The points are distributed in the following way:

1. Implementation of inverse Fourier transform - 3 pts
2. Implementation of the power cepstrum - 3 pts
3. Plot of the signal as a function of time (4 plots) - 2 pts
4. Plot of the Fourier transform of the signals and establishing the correct fundamental frequencies (4 plots) - 4 pts
5. Plot of `sincheck` (4 plots)- 2 pts
6. Plot of the logarithm of the Fourier transform - 2 pts
7. Plot of the cepstra - 2 pts
8. Comment on the quefrencies - 2 pts

References

[1] B.P.Bogert, M.J.R. Healy, J.W. Tukey, The quefrency alanysis of Time Series for Echoes: cepstrum, pseudo-autocovariance, cross-cepstrum and saphe cracking, Proceedings of the Symposium on Time Series Analysis Chapt.15, 209-243, New York: Wiley, 1963