Signal processing labs

M1 PPN

A. Coillet

aurelien.coillet@u-bourgogne.fr

Academic year 2023-2024

Digital signal

The following tutorials will be performed on a computer, with Octave, an open source software for numerical simulations. The goal of this tutorial is both to use the concepts seen in the lectures, and to get familiar with digital signal processing, especially with the fast Fourier transform.

1 Brief introduction

GNU Octave is a software featuring a programming language mainly aimed at numerical computations, and similar to the one found in Matlab. In the framework of this lecture, we will use its abilities to work efficiently on arrays and matrices of numerics, using a wide variety of operations.

In order to efficiently use Octave, a small number of reflexes are necessary. The first consists in **go and look for documentation and help,** either in Octave's help (`help command`) or on the internet, on Octave and Matlab's websites, or on dedicated forums such as `stackexchange.com`. The second reflex is to **read carefully and correctly interpret the error messages.** They are your main source of information to understand what is not working in your code. Their grammar needs some time to getting used to, though.

Octave can be used in its interactive mode, using the command window. In this mode, each time the Enter key is pressed, the command is executed. The other mode that we will mostly use is the script mode, where a list of commands is saved in a text file (the script). Octave offers a dedicated editor to write these script files, from which we can execute it with the F5 key. This feature is very convenient and allows for a fast script development.

In Octave, variables are used to store data, numbers, arrays, matrices, ...

```
txt = 'Hello world!';      % Strings
var = 5.867;               % Numerics
tab = [1 3 5 19 0];        % Array or vector of numbers
mat = [1 i; -i 1];         % Matrices and complex numbers
```

The semicolon at the end of the line prevents the displaying of the result in the command window:

```
mat = [1 2 3; 4 5 6; 7 8 9]             % 3x3 matrix
    mat =
          1   2   3
          4   5   6
          7   8   9
```

One can access the elements of a matrix using the `(i,j)` notation:

```
mat(3,2)
    ans = 8
mat(2,:)
    ans =
          4   5   6
```

The `*` symbol, when applied to matrices or arrays corresponds to the matrix product. In order to obtain the element to element multiplication, we use the `.*` symbol:

```
A = [1 2 3]; B = [4 5 6];
A*B
    error: operator *: nonconformant arguments (op1 is 1x3, op2 is 1x3)
A.*B
    ans =
          4 10 18
```

Applying a function on a variable is done using `(var)`:

```
m = mean(tab)
    m =  5.6000
```

Some of Octave's features are available in various packages that need to be loaded in order to access these features. In particular, the `signal` package contains a number of very useful functions for our purpose. We will therefore use the following command at the beginning of our scripts:

```
pkg load signal
```

2 Sound

The processing of sound signals is particularly well suited for digital signal processing. The goal of this activity is to become familiar with sound signals as well as the use of Octave.

Musical note and timbre In the folder `TDSound`, you will find 4 audio files: `PianoLaPeriod.wav`, `TrumpetLaPeriod.wav`, `ViolinLaPeriod.wav` et `FluteLaPeriod.wav`. They correspond to the sound recording of a single period of an instrument playing the A note (La in French).

Q 2 - 1 Using Octave's help for the `audioread` function, find how to load each of these files, and store the audio signal as an array, and the sampling frequency as a float. Comment on the sampling frequency with regards to the Shannon-Nyquist theorem.

Q 2 - 2 What is the time interval between two values of the sound signal? From this, construct an array that stores the time value for each sound value. Use the documentation available on the internet to learn how to do that. Repeat this operation for each instrument.

Q 2 - 3 Using the `plot` function, draw on the same graph the four different sound signals. Add labels for the axes and a legend to the graph (this should be done for every graphs from now on).

Q 2 - 4 The length of these samples is too short to be able to listen to them. Build an array with 1000 periods, and play it using the `sound` function. Can you hear differences between the 4 sounds? Do they appear natural to you?

Q 2 - 5 We want to calculate the Fourier coeficients for these samples. Since they are periodic and with a power of 2 number of points, we can directly use the `fft` function to calculate the coefficients. Read the help page for `fft` and store the Fourier transform of each signal of the four samples into four variables.

Q 2 - 6 Build the frequency axis as seen in the lecture.

Q 2 - 7 Plot the Fourier spectrum of the four signals. We will use the `stem` function instead of the `plot` function, given the discrete nature of the Fourier series development.

Q 2 - 8 Comment on the graph. Which is the fundamental frequency? What is its value, and to which note does it correspond? Which instrument has the most harmonics? Link this latter observation to your perception of the sound of the instruments. What is the meaning of the right part of the spectrum? Is the Shannon-Nyquist criterion respected for the fours sound samples?

Spectrograms The samples we just studied do not sound very faithful to what we are used to when we talk about a piano, a violin or a trumpet.

Q 2 - 9 Propose an explanation for why the previous samples do not sound natural.

Q 2 - 10 The files `PianoLa.wav`, `TrumpetLa.aif`, `ViolinLa.aif` and `FluteLa.aif` correspond to the full recording of the sounds from which the previous period samples have been taken. Perform a similar analysis to what we did previously: load the files, find the frequency sample and comment it, play the sound. Is the sound closer to what you expect?

Q 2 - 11 Plot the audio signal for each one of the sounds. Note that each array contains two column corresponding to the left and right channel (stereo recording); for simplicity's sake, we will only study one of them. Using the zoom feature of the plot window of Octave, interpret the shape of the sound signals. Show that we can split the evolution of the signal in different phases, and link them to your sensation while listening to the sound. Compare the period of the sound for these different phases.

Q 2 - 12 Calculate the Fourier transform of the signals and create the frequency axis that matches. Plot the Fourier spectrum. In contrast to the previous case, this signal is not periodic, and this calculation is indeed a Fourier transform and no longer a Fourier series development.

Q 2 - 13 Discuss on the shape of the spectrum, in particular the following features:

- the existence of peaks,
- the structures inside each peak,
- the number of point of each curve.

When hearing these sounds, our feeling is that the sound evolves, according to the stages we described in the previous questions. The human ear is indeed a mixed detector: it is sensitive to notes, that are frequencies in the Fourier domain, but it can follow the evolution of the Fourier components *in time*. In order to reproduce this sensation, we will need a *spectro-temporal* analysis using the *spectrogram* tool. The principle behind such an analysis is to perform Fourier transforms on a time windowed version of the signal, and not on the signal itself. The time window is then successively shifted and the Fourier transform repeated. With this operation, we will obtain the evolution in time of the Fourier transform of a sample of the sound.

Q 2 - 14 Propose an order of magnitude of the brain's response time. Below this value, we will consider that the information content will be in the Fourier domain (interpreted as a frequency), and above this value, it will correspond to a time evolution. How many points `Nw` of the sound signal does it correspond to? To accommodate for the use of power of 2 in the fast Fourier transform operation, we will use the `nextpow2` function to get an appropriate number of points.

Q 2 - 15 To calculate the spectrogram, we will perform the FFT operation on `Nw` number of points, and then shift the window by `Nw/4`. To which time duration does that correspond to? How many shifts will we have to do in order to sweep the entire sound trace? Build a matrix full of zeroes whose lines will store the Fourier spectrum data, and whose columns will correspond to the successive times.

Q 2 - 16 Using a `for` (or `while`) loop, and the notation `A(i:j)` which allows to only keep the elements of `A` that are within `i` and `j`, fill in the previously initialised matrix with the spectrogram of the sound recordings.

Q 2 - 17 The next stage is now to find a way to visualize the content of this spectrogram. The `pcolor` function displays data from `A(i,j)` matrices using a colormap for the *z-axis*. Display the spectrogram using this function, and use a decibel scale for the colormap.

Q 2 - 18 Build the time and frequency axes for this graph, add labels to them, limit the display to an appropriate frequency range, and repeat this operation for the other 3 sounds. One can use the `subplot` function which allows to divide the figure window.

Q 2 - 19 These graphs are particularly complex to interpret, and one of the reasons for this comes from the use of a rectangular time window: the `A(i:j)` operation amounts to multiply the signal with a rectangular window. Modify your code to make use of the `hamming` and `hanning` windows. Discuss the differences between these window and the rectangular one.

Q 2 - 20 At this stage, we have obtained a rich and faithful representation of the sounds. What are their main characteristics? Cite some visible differences between the various instruments. Can you link these characteristics to physical properties of the instrument?

Complex sounds

Q 2 - 21 Perform the same analysis for the sound of a guitar, a cymbal, or whichever instrument (see the website `http://theremin.music.uiowa.edu/MIS.html` for a wide variety of sound samples). Discuss the shape of the spectrograms and the difference between instruments.

Q 2 - 22 Use the same method to analyse the sound `Piano.ogg` and discuss the resulting spectrogram.

In the standard music scale, one octave contains 12 equally spaced notes. These note intervals correspond to the ratio between the frequency of 2 notes, which we call r. Going from one note to the one above amounts to multiplying the frequency by r, and doing so 12 times (one octave) means multiplying the frequency by 2. This can be mathematically written as:

$$2 = r^{12} \tag{2.1}$$

The ratio between the frequencies of two successive notes is therefore $\sqrt[12]{2}$.

Q 2 - 23 The A_3 note has a frequency of 440 Hz. Calculate the frequencies of the notes of the following octave. Add these notes to the graph of the spectrogram that you just calculated. Deduce from that the score of the piano air.

3 Images and filters

In this lab, we will study the 2 dimensions Fourier transform, in the framework of image data processing.

Simple examples Let us begin to get familiar with 2D Fourier transform with simple images.

Q 3 - 1 Build a 256x256 matrix filled with zeroes, and then give it the following values:

$$\texttt{Image1(i,j)} = \sin\left(\frac{2\pi(i+j)}{5}\right) \tag{3.1}$$

Draw the image using the `imagesc`. Why does the image appear with colors? You can change these colors with the `colormap` function.

Q 3 - 2 Calculate the 2D Fourier transform of this image using the `fft2` function, and draw the amplitude and phase spectra. We will use the `fftshift` function to put the zero frequency in the center of the image. Interpret the spectra with your knowledge of the Fourier transform.

Q 3 - 3 Perform the same analysis with a Gaussian function centered on $O = (129, 129)$, another Gaussian center on $i = j = 100$, a 2D rectangular function centered on O and another one centered on $(100, 100)$. For each image, interpret the phase and amplitude spectra.

Real images and filters

Q 3 - 4 Load the image `LymphocyteGray.jpg` with the `imread` command. Plot this image as well as its amplitude and phase spectra. In order to better visualize the low amplitude spectral components, you can use a decibel scale. Discuss the graphs.

Q 3 - 5 From this Fourier spectrum, show that one can go back to the original image using the inverse Fourier transform `ifft2`. Perform the same `ifft2` operation on the absolute value of the spectrum, and conclude on the influence of the phase in image processing.

Q 3 - 6 Create a 256x256 matrix such that:

$$\texttt{Flp(i,j)} = \begin{cases} 1 & \text{si } (i-129)^2 + (j-129)^2 \leq 20^2 \\ 0 & \text{sinon} \end{cases} \tag{3.2}$$

Plot this matrix with `imagesc`. What type of filter does it correspond to? Apply this filter to the image, and display the resulting spectrum. Caclculate the inverse Fourier transform and draw it. Discuss the resulting image and the potential use of this filter.

Q 3 - 7 Perform the same operation as the previous question using the following filter:

$$\texttt{Fhp(i,j)} = \begin{cases} 0 & \text{if } (i-129)^2 + (j-129)^2 \leq 20^2 \\ 1 & \text{if not} \end{cases} \tag{3.3}$$

Which kind of filter is this? Which characteristic in the filter shape leads to the visible oscillations in the filtered image?

Q 3 - 8 Perform the same analysis for the Gaussian filter:

$$\texttt{Glp(i,j)} = \exp\left(\frac{-(i-129)^2 - (j-129)^2}{20^2}\right) \tag{3.4}$$

and for the Butterworth filter:

$$\texttt{Blp(i,j)} = \frac{1}{1 + \left(\frac{(i-129)^2}{20^2} + \frac{(j-129)^2}{20^2}\right)^n} \tag{3.5}$$

where you will change the order of the filter n. What are the advantages of such filters compared to the rectangular one?

Q 3 - 9 Load the other images from the folder and draw their amplitude spectra. Link their spectral characteristics to the spatial features of the images.

4 Radar

In this exercise, we simulate the principle of the radar. A radar system is composed of an antenna that emits radio waves and a detector for these waves. When an object reflects the waves, the detector records this echo with a time delay corresponding to the travel time of the waves. In order to be easily recognizable, the sent signal has to have a easy to use shape. A chirped sine function, where the period of the sine changes with time, is often used for that purpose:

$$x(t) = \sin(2\pi f t(t + 2\pi)) \tag{4.1}$$

Q 4 - 1 Create a time vector going from 0 to 20, and the vector `x` corresponding to the chirped signal, with a frequency $f = 0.1$. Plot this signal and check that the number of points is enough.

Q 4 - 2 Create a new vector going from 0 to 100 with the same time interval as in the previous question. This new array will be the base for the simulation of the detected signal. In order to take into account the contribution from environmental and detection noises, create a `noise` vector with the `randn` function.

Q 4 - 3 Create a detected signal `y` corresponding to the addition of the noise and a fraction (we will take 0.3) of the signal `x` delayed with an arbitrary `i` index. Draw both the `noise` and `y` signals. Can you spot the presence of the `x` signal in the detected signal `y`?

Q 4 - 4 Calculate the cross correlation between `x` and `noise` on one hand, and between `x` and `y` on the other hand. This operation is performed using the `xcorr` function. Reading the help page of xcorr is suggested, due to small differences between the continuous signal version of the correlation and this discrete and finite signals one. Plot these 2 correlations and find the value of the delay.

5 Optical pulses

Q 5 - 1 We can model a laser pulse by the product of a very fast (typically 100 THz) oscillation with a Gaussian curve with a typical duration of 1 ps. Create 2 vectors, the first one representing the time and going from −50 ps to 50 ps, and the other representing the laser pulse. We will use an oscillating frequency of 10 THz in order not to saturate the memory of the computer.

Q 5 - 2 Calculate the Fourier transform of the laser pulse, build the frequency axis, and draw the amplitude and phase spectra. One can use the `unwrap` function for the latter.

Q 5 - 3 In experiments, one can only record the power spectral density of an optical signal (using a spectrometer), and the phase information is therefore lost. Draw the power spectral density of the modelled pulse signal.

Q 5 - 4 Perform the same analysis as previously on a set of two pulses delayed by `t0` $= 6$ ps. We will take the same frequency for both pulses, but the phase difference ϕ will be arbitrary set. Discuss the shape of the power spectral density when the delay and phase are changed. Can these features be seen using experimental techniques? Justify that this phenomenon can be called interferences.

Q 5 - 5 Calculate the autocorrelation of this laser signal with 2 pulses using the `xcorr` function. Plot the result and build the appropriate x-axis array. What is the other way to calculate the autocorrelation? Show thanks to Octave that both methods give the same result.

In the past 10 years, an experimental technique has been developed to allow for the extremely fast recording of the spectrum of pulsed signals. This technique uses the dispersive properties of an optical fiber, where light propagates at different speeds depending on its wavelength. Hence, a short wavelength will arrive first (for instance) on a photodetector, before any other wavelength that is larger. This technique is called the Dispersive Fourier Transform (DFT) and is particularly useful for pulsed fiber lasers, where trains of pulses are emitted at each round-trip of the laser cavity. By using a fast oscilloscope, one can reconstruct the evolution of the pulse spectrum, round-trip after round-trip.

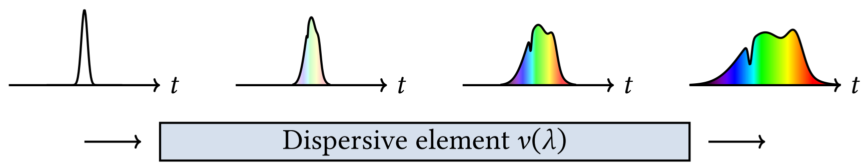

Figure 1: Principle of the dispersive Fourier transform.

Q 5 - 6 Justify the name *Dispersive Fourier Transform*.

Q 5 - 7 A recording of an experiment with a fiber laser and using the DFT technique is available in the `DFT.dat` file. Once `load` has been executed on it, you will obtain a matrix whose columns are the values of the spectrum, and the lines are the successive round-trips. The value of the optical frequencies of the columns are stored in the `freqTHz.dat` file. Load both files and use the `pcolor` function to visualize the evolution of the power spectral density. Compare the shape of the spectrum at one arbitrary round-trip to the simulated ones, and discuss.

Q 5 - 8 In order to check that there are two nearby pulses in this signal, we will calculate the Fourier transform of the power spectral density. To which physical quantity this Fourier transform will correspond? Plot the evolution of it with the round-trips. Do you confirm the results from the previous questions?

Q 5 - 9 What does the x-axis stand for? Build the appropriate array, and deduce from that the main characteristic of the pulse.