Lab4,EE4C5,DSP

Prof. Naomi Harte, Dr Jean-Baptiste Monteil

November 7, 2023

1 Introduction

1.1 Frequency domain

Observing a music signal, such as those we can create with the function createMusic.m (provided in Blackboard, Lab4 folder), in the time domain does not readily provide us with important information about the signal. We have noticed a few times already that we can only see detail over a very short time window. We will instead observe properties in the frequency domain in this Lab, by observing the discrete Fourier transform of our signal. In this way we are able to identify all the notes played in the music.

1.1.1 Observe a note

1. In MATLAB, create a script named lab4Part1.m in which you define the values of f_s, d, $note$ and $start$. You should use $f_s = 8000$ Hz, $note = 69$, $d = 2$ seconds, and $start = 0$ seconds.
2. Create the note with the help of the function createNote.m provided in Blackboard, in the Lab4 folder. Plot the discrete Fourier transform of the signal thanks to the function my_FFT.m, also provided in Blackboard Lab4 folder. Annotate your figure. Comment on what you observe and include the figure in your write-up
3. Among the notes listed in Fig. 1, which ones can we observe correctly with the DFT with the current set-up? Can we correctly observe the note 110 (note that 110 is not in the table of notes but still exists)? Give a reason why.

octave	C	C#	D	D#	E	F	F#	G	G#	A	A#	B
1	24 32.7	25 34.65	26 36.71	27 38.89	28 41.2	29 43.65	30 46.25	31 49	32 51.91	33 55	34 58.27	35 61.74
2	36 65.41	37 69.3	38 73.42	39 77.78	40 82.41	41 87.31	42 92.5	43 98	44 103.83	45 110	46 116.54	47 123.47
3	48 130.81	49 138.59	50 146.83	51 155.56	21 164.81	53 174.61	54 185	55 196	56 207.65	57 220	58 233.08	59 246.94
4	60 261.63	61 277.18	62 293.66	63 311.13	64 329.63	65 349.23	66 369.99	67 392	68 415.3	69 440	70 466.16	71 493.88
5	72 523.25	73 554.37	74 587.33	75 622.25	76 659.26	77 698.46	78 739.99	79 783.99	80 830.61	81 880	82 932.33	83 987.77
6	84 1046.5	85 1108.73	86 1174.66	87 1244.51	88 1318.51	89 1396.91	90 1479.98	91 1567.98	92 1661.22	93 1760	94 1864.66	95 1975.53
7	96 2093	97 2217.46	98 2349.32	99 2489.02	100 2637.02	101 2793.83	102 2959.96	103 3135.96	104 3322.44	105 3520	106 3729.31	107 3951.07

Figure 1: Notes

1.1.2 Observe a melody

1. In the same MATLAB script, create an audio signal corresponding to a snippet of music with 4 successive notes (select the note number $note$, the duration d, and the start $start$ for each note, all to be integers). Use $f_s = 8000$ Hz. Please call the function createMelody.m (provided in Blackboard Lab4 folder). Please refer to Fig. 1 to select the notes.
2. Plot the discrete Fourier transform of the resulting audio signal. Annotate your figure. Show that you can find which notes were played uniquely from the Fourier transform graph.

1.2 Separation of multiple sources

A major application area within signal processing is source separation - to take a signal containing a mixture of multiple signals and generate separate signals with the different original sources reconstructed, e.g. recover the speech of each member of the group from a recording of a group of people speaking at the same time. It is also a common problem in communications.
In our lab, we will add two audio signals, and recover each of them with the help of a linear filter. To this end, we select between an ideal lowpass filter (if we want to remove the high-pitched notes), an ideal highpass filter (to remove the low-pitch or deep notes), and an ideal bandpass filter (to only keep mid-range notes). The transfer functions depend on the cutoff frequency and are defined as follows:

- lowpass $H_{LP}(\omega) = \begin{cases} 1 & \text{if} \quad |\omega| < \omega_c \\ 0 & \text{otherwise} \end{cases}$

- highpass $H_{HP}(\omega) = \begin{cases} 0 & \text{if} \quad |\omega| < \omega_c \\ 1 & \text{otherwise} \end{cases}$

- bandpass $H_{BP}(\omega) = \begin{cases} 1 & \text{if} \quad \omega_{c1} < |\omega| < \omega_{c2} \\ 0 & \text{otherwise} \end{cases}$

1.2.1 Separate two notes

1. In a new script lab4Part2.m, create an audio signal containing two notes occurring simultaneously: one high-pitched, and one low-pitched, both sampled at $fs = 8000$ Hz. You can use the supplied function createMelody.m. Listen to the audio signal to ensure it contains the notes as you intended. Plot the discrete Fourier transform of your signal as before, and propose a cutoff frequency able to separate the two notes. Include your plots and decision in your write-up.
2. Create a vector HLP and a vector HHP corresponding to the transfer functions of the ideal low-pass and highpass filters. You must create vectors of the same size as the vector of frequencies f, the second output of the my_FFT.m function. The vectors HLP and HHP will contain 0 and 1.

```
%to modify the values of a vector through a condition, you can
   use
y(y<0) = 0; %all negative values of y are put to zero
z(y<3 & y>0) = 1; %all values between 0 and 3 are set to 1
```

3. Noting that you are working in the frequency domain, apply the ideal lowpass filter HLP to the DFT of your audio signal. Then apply the inverse Fourier transform to find the filtered audio signal (i.e. return to the time domain). Please use the function my_FFTinv.m to do so. Listen to the filtered audio and comment on how it sounds in your write-up. Plot the DFT of the filtered audio. Did you remove the note you expected with this filter? Please annotate your figure.

```
%to multiply two vectors element-wise, you can use:
z = y1.*y2;
```

4. Do the same as in 3. for the ideal highpass filter HHP.

1.2.2 Separate two sources

Let's now create two music signals, add them, then separate them back with the filters we have designed.

1. Create a signal $x1$ of duration 10 seconds, sampled at $f_s = 8000$ Hz, containing notes (at least 4) between 50 and 71. Listen to the obtained signal. Include a DFT of your signal in your write-up.

2. Create another signal $x2$ of duration 10 seconds, sampled at $f_s = 8000$ Hz, containing notes (at least 4) between 72 and 90. Listen to the obtained signal. Include a DFT of your signal in your write-up.

3. Create the signal $x = x1 + x2$. What cutoff frequency can separate the two signals $x1$ and $x2$? Verify by examining a DFT of the new signal and create your filters.

4. By using the designed filters, reconstruct from x the signals $y1$ and $y2$, which respectively estimate the original signals $x1$ and $x2$.

5. Listen to the obtained signals and compare them to the real sources. Compute the Euclidean distance between $x1$ and $y1$, $x2$ and $y2$.

6. Observe the spectrum of the signals $x1$, $x2$, $y1$ and $y2$.

7. Test other cutoff frequencies and comment.

8. **Optional challenge**. Listen to the melody2.wav file provided in blackboard Lab 4 folder. Observe its Fourier transform, propose and apply a strategy to obtain uniquely the melodic line.

1.3 Spectral analysis

1.3.1 Instruments

In Blackboard, within the Lab4 folder, you can find the music note C5 for different instruments (the piano, the trumpet, the violin, and the flute).

1. By using the script plot_spectogram_various.m, plot the spectogram of the note C5 for each instrument (consider the full note which lasts $2 \sim 3$ seconds). Comment on the differences and how the spectrogram captures the specific nature of each instrument. (You may need to read a little about each instrument for this.) Observe how the musical effect that the instrument produces is related to the frequencies in the spectogram.

2. Now plot the spectogram of a window of length 40 ms for each instrument. Comment on the differences between the spectogram of the full note and the one of the window.

1.3.2 Queen

In Blackboard Lab4 we provide a recording of the song 'Another One Bites the Dust' from Queen.

1. Consider the spectogram of this song for the time sequence 2'28 - 2'31. Plot and annotate your spectrum.
2. Make observations on the spectrogram. Can you identify the drums, Freddie Mercury singing, the electro effects? In your write-up, either with a clear explanation or annotation, identify all these events in your spectogram.
3. **Optional challenge**. Isolate another short portion of the song, and produce another annotated spectrum with notable features.

1.3.3 Speech example

The speech file 'OSR_us_000_0010_8k.wav' is provided in blackboard Lab 4

1. Look at the spectogram for different window lengths. Make observations on how the window length affects the spectogram. Please include at least 3 plots of different spectograms in your report.
2. Now corrupt the speech file with a super-imposed tone. Plot the spectogram for a window with length of your choice. Identify the noise on the spectogram, and the speech frequency content.

1.3.4 Heartbeat data

In blackboard Lab 4, we provide two datasets: one containing normal ECGs, the other containing ECGs from patients with arythmia.

1. Plot the ECG of one healthy patient against the ECG of a patient suffering from arythmia. Include the plots in your report.

```
T = readtable('ptbdb_normal.csv');
Tarray = table2array(T);
normalPatient = Tarray(1,1:end);
```

2. Observe the spectogram for each patient (the healthy one and the patient suffering from arythmia). Include the spectograms in your report. Can you observe anomalies more easily with the spectogram or the time-domain signal? Explain, with reference to the plots in your write-up.

2 Pre-clinic submission

Before your assigned clinic, you must upload a record of your "work in progress" on the lab material. This is rough-work and can include e.g. initial code, cut'n'paste from the MATLAB commandline, initial figures or draft code. This submission is required whether or not you attend your clinic. It is not marked, but is a record of your progress before the clinic.

3 Write-up for (post-clinic) submission

The purpose of this lab is that you engage with the assigned tasks, get a better appreciation of the real-world analysis of signals, and make connections between what you are learning in lectures and practical use of those concepts. Therefore, you should not be spending a huge amount of time on the write-up. The purpose of the submission is to show you have done all the assigned tasks, that you have thought about the questions asked, and that you are achieving the required level of proficiency in MATLAB.
With this in mind, you should submit:

- A single pdf with all required figures and commentary on the tasks in the lab. Bulleted comments are allowed.
- The pdf above must use a sans serif font, and adequate font size and spacing.
- You must include the declaration on plagiarism as per your course handbook.
- You can use snippets of code in the pdf to illustrate a point you wish to make, but all code must be submitted in .m files.
- Ultimately, we should be able to run your code and recreate your plots etc from what you submit.
- Include all relevant wav files, giving them informative names
- All code should be commented.

Note: All deadlines are detailed on Blackboard, and students must make themselves aware of when items are due.