Lab2,EE4C5,DSP

Prof. Naomi Harte, Dr Jean-Baptiste Monteil

October 3, 2023

1 Introduction

1.1 Model

We consider the following continuous signal:

$$x(t) = \sin(2\pi f_0 t) \tag{1}$$

It is a pure sinusoid with frequency f_0, amplitude 1, phase 0. The signal corresponds to a music note. The higher the frequency is, the more high-pitched the sound is. For instance, a sinusoid with frequency $f_0 = 440$ Hz corresponds to A/*la*. This continuous signal cannot be studied in MATLAB (it contains a infinite number of samples and values are not quantized), hence we can only study the sampled and quantized version of this signal. We define the numeric signal $x[n]$ sampled with sampling frequency f_s, which contains N samples, as:

$$x[n] = \sin(2\pi f_0 t[n]) \tag{2}$$

The corresponding times $t[n]$ are defined as:

$$t[n] = \frac{n}{f_s} \quad for \quad 0 \leq n \leq N-1 \tag{3}$$

By default, MATLAB quantizes all values with 64 bits. In this lab, the considered signal has the following features: sampling frequency $f_s = 8000$ Hz, fundamental frequency $f_0 = 800$ Hz, duration $d = 2$ secs.

1.2 Nyquist criterion

For this subsection, create a Matlab script called lab2Part1.m

1. According to Nyquist, what is the minimum value of the sampling frequency that can be used to not degrade the signal above, $x(t)$?

2. To illustrate the Nyquist criterion, choose relevant values of f_s (choose multiple values above and below the minimum you identified) and listen to the obtained numeric signals $x[n]$. Can your verify the theory? Please write down the values of f_s you chose and explain what impact you have observed.

3. Let's suppose once more that $f_s = 8000$ Hz. Plot the sampled version of two sinewaves, one with frequency $f_0 = 800$ Hz, the other with frequency $f_0 = 7200$ Hz, both sampled with $f_s = 8000$ Hz. You can also listen to both. What do you observe? Comment on the results with reference to the theory you covered in lectures.

4. Load the 'handel' audio .wav file in your script and listen to it. What is the sampling rate of this signal? Now undersample at successively decreasing sampling frequencies until you notice a change. What is the frequency where you first perceive aliasing? Describe what the effect of aliasing sounds like to you. Plot the original audio file and its under-sampled version using SUBPLOT. As usual, label everything appropriately.

```
load handel.mat
filename = 'handel.wav';
audiowrite(filename,y,Fs);
clear y Fs
[y, Fs] = audioread('handel.wav');
```

1.3 Create a note

We have considered so far a sine wave with fundamental frequency f_0, with no concern about the significance of f_0. However, a musical instrument emits a frequency when producing a note. For instance, a piano has a finite range of notes/frequencies.

We can associate an integer with each note and we compute its fundamental frequency:

$$f_0^{\text{note}} = 440 \times 2^{\frac{\text{note}-69}{12}} \tag{4}$$

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: keyboard[1]

In the above table, you can find some notes of the keyboard, their corresponding number and frequency:

1. Create in MATLAB an empty script named lab2Part2.m where you define the values of f_s, d and $note$. You can select the values of d and $note$, use $f_s = 8000$ Hz.

2. Create a function createNote.m taking as input a duration d, a note number $note$, and a sampling frequency f_s. The output should be the signal x, corresponding to the sine wave of duration d and fundamental frequency f_0^{note}, and associated with a time vector t. **In MATLAB, the name of the script must be the same as the name of the function.** In the case where note $= -1$, the output is a signal of zeros of duration d.

```
function [out1,out2] = functionName(in1,in2,in3)
```

3. In the script lab2Part2.m, test the function and generate different notes with different frequency and different duration. Inlcude plots in your write-up that show the code is working.

4. Test the case note $= -1$ and show your output in this case is as per the specification above.

1.4 Create a melody

A melody is a sequence of notes. We can model multiple notes each with specific duration into vectors.

1. With a loop over a set of integers of your choice (note numbers), create a vector containing the corresponding fundamentals.

2. With inspiration from createNote.m, create a new function createMelody.m taking as input a vector of duration D_VECT, a vector of note numbers NOTE_VECT and a sampling frequency f_s. The output is the concatenation of several sine waves whose duration and height are defined in the vectors D_VECT and NOTE_VECT. For simplicity, assume all notes have the same duration and start at the same time.

```
y = []; %create an empty vector
y = [y1 y2]; %concatenate row vectors y1 et y2
y = [y1 ; y2]; %concatenate column vectors y1 and y2
N = length(y); %length of vector y
y = zeros(N,1); %create a column vector of length N with zeros
y = zeros(1,N); %create a row ...

%for loop syntax in MATLAB
j=0;
for i=1:3
    j=j+1;
end
```

3. Test your function in the script lab2Part3.m by selecting vectors D_VECT and NOTE_VECT of your choice. Listen to the melody. Save it as a wav. file. Plot the signal against time and include in your writeup. Clearly indicate the notes and duration that you have chosen.

4. Please modify this melody so its height is one octave up. *An octave is a gap of* 8 *tons/* 12 *notes, between two notes. For instance, an octave separates the note* 45 *(A octave* 2*) from the note* 57 *(A octave* 3*).* Explain how you do it. Please make the melody go 50% faster. Explain what you need to change. Save it as another wav. file, with a suitably informative name.

5. **Optional Challenge.** Write a new function createMusic.m that can play notes with different duration and starting at different times. These could be random for example, within certain constraints. Test your function, and save the created music in a wav. file. Comment on how melodic your music is (or is not!)

1.5 Upsampling/downsampling

It is often necessary to change the sampling rate of a discrete-time signal to obtain a new discrete-time representation of the underlying continuous-time signal of the form

$$x_1[n] = x_c(nT_1), \tag{5}$$

where $T \neq T_1$. This operation is called resampling.

1. Pick a note/frequency from Figure 1. Design the pure sine wave of that frequency, by selecting the sampling frequency and the duration. Upsample your signal by a factor of 2. Plot both your signal and the upsampled version. How does upsampling affect your signal? Try different sampling factors.

2. Reload the 'handel' wav file. Upsample it by an integer factor of 2, and then 4. Listen to the output. Repeat this upsampling process, until you perceive artefacts in the music. Plot the original and the downsampled signals. Listen to the original and distorted upsampled signal. At what level of upsampling do you begin to perceive distortions? What is the effective sampling frequency? How does this relate to the theory you covered in lectures? Save the resulting wav file and include in your submission.

Crux: The *resample* built-in function in MATLAB uses an antialiasing lowpass filter. Since the goal of this question is to observe aliasing, you must upsample the signal without any anti-aliasing filter, i.e. this task will not work if you use *resample*. You need to first expand the signal - by putting zero-valued samples between actual samples to increase the sampling rate. Then you need to conisder how to implement the interpolation part. You may get inspiration from the following code snippet.

```
L=2; %the sampling factor
y=zeros(1,L*length(x)); %an array with length the original
   signal x multiplied by the sampling factor.
y(1:L:length(y))=x; %In the upsampled signal y, every L steps,
   you can insert a value of the original signal x.
```

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.

References

[1] Fundamental frequencies of notes in western music.
https://auditoryneuroscience.com/pitch/fundamental-frequencies-notes-western-music.