First Lab Assignment - Sampling and Quantisation of media signals

Objectives: This work aims to allow the student to apply and consolidate knowledge concerning media sampling and quantisation.

Introduction

The objective of this work is to allow students to gain a better understanding on the principles and effects of sampling and quantisation of media signals, notably audio signals. The intention is to learn the effects that those processes have on the quality of the signal by varying the sampling rate and the number of quantisation levels used to represent the acquired samples.

To conduct the work, audio signals (speech or music) may be acquired with the mp3 format and stored with the Wave encapsulation format or to use already available signals with the Wave format (found in the Moodle of the course, in the section "Multimedia Resources", folder "Lab1 – Audio Files"). To those signals it will be applied sampling and quantisation operations with different values and using or not filters. This will be achieved by using scripts Matlab available on Moodle in the folder "scripts". By comparing the quality of the sound generated by the different algorithms and with the different parameters' values the students should acquired a better understanding of the role played by them on the final quality (sampling, quantisation, pre and post-filtering).

It is advised to use headphones to better evaluate the quality and not interfere with the experiences of the other colleagues.

Work to be developed

1. **Optional/*Introductory part that may be developed outside the classroom***:

Select a music file mp3, open the program VLC and select the option "`File>Convert/Stream…`"; in the section "`Choose Profile`" select "`Audio -mp3`" on the left and then click on the right "`Customize`"; to select the "`WAV`" encapsulation method; press the option "`Audio Codec`", select "`WAV`" for the codec and choose a sampling rate ("`Samplerate`") of 11025 Hz; press "`Apply`" and when going back to the previous window, select an mp3 file in the option "`Open media`" (you may also "`drag-and-drop`" the mp3 file you want to work with). Finally select "`Save as File`" indicating the desired filename and the location.

Repeat these procedures but now selecting a sampling rate of 44100Hz (audio CD quality).

Replay each of the stored files and compare their quality. Take note of the major differences that you have noticed. <u>Include that information in your report.</u>

2. Variation of the sampling frequency with or without filters

In this part you are going to work with Matlab using the scripts "*amostragemInterp_semFiltro.m*" e "*amostragemInterp_comFiltro.m*" available on the course Moodle. These programs are modified versions of programs obtained from the Web site of the course *Multimedia Communication Systems I* of the Polytechnic University in Brooklyn.

Before starting your work, analyse the code of the available programs to better understand the operations performed. To help you in this task, flowcharts of the programs are provided at the end of this report. Remember that, every time that you do not understand the operation of a given Matlab function or the type of parameters it expects to receive or the output it produces, you may write in the command window of Matlab "help <name_of_the_function>".

As input files you should use uncompressed audio files with the WAVE format ("`xxx.wav`"). You will find some examples on Moodle in the section "Multimedia Resources", folder "Lab1 – Audio Files". You may also convert your own mp3 file adopting the procedure explain in the previous point selecting the largest possible sampling rate (44100Hz).

Start Matlab and change to your own working directory. Copy all files (scripts and audio files) to that directory.

i) run the script "`amostragemInterp_semFiltro.m`" using a .wav file. For example, if you want to use the file "`inputSound.wav`", want to save the processed sound in the file "`outputSound.wav`" and perform a 4 to 1 sub-sampling, you should execute the following command on Matlab:

```
>>amostragemInterp_semFiltro('inputSound.wav', 'outputSound.wav', 4)
```

ii) Compare the original sound and the processed sounds after sub-sampling and interpolation in terms of perceptual quality, waveform and spectrum. Evaluate the obtained mean square error (note: *MSE=Max/10^(PSNR/20)*). Run the script again, now performing a subsampling with a factor 2. Compare the results obtained in the two experiences.

iii) Repeat now the same experiences but with the script "`amostragemInterp_comFiltro.m`". Compare the results from the perceptual and objective viewpoints. Which of the programs leads to a better results for the same value of k? Analyse the frequency response of the used filters. Which should be ideally the cut-off frequency of these filters? Verify if they present sufficient attenuation in the desired cut bands.

In the report you should provide answers to the questions raised above, present the relevant waveforms of the signals and comments of your perceptual and objective analysis, comparing the results obtained in the different experiences.

3. Quantisation experiences

In these experiments you will use the script "`quant_uniform.m`", which allows to quantise an audio signal using a variable number of quantisation levels or bits per sample, defined by the user as an input parameter.

Run the program "`quant_uniform.m`" using as an input signal an uncompressed sound with the wave format just like in the previous experiments but making sure that that signal has been digitised using 16 bits per sample (PCM format: 16 bits per sample and 44,1kHz for te sampling frequency). If you use file called "`inputSound.wav`", want to save the processed sound in the file "`outputSound.wav`" and perform a quantisation with 256 levels, you should execute the following command in Matlab:

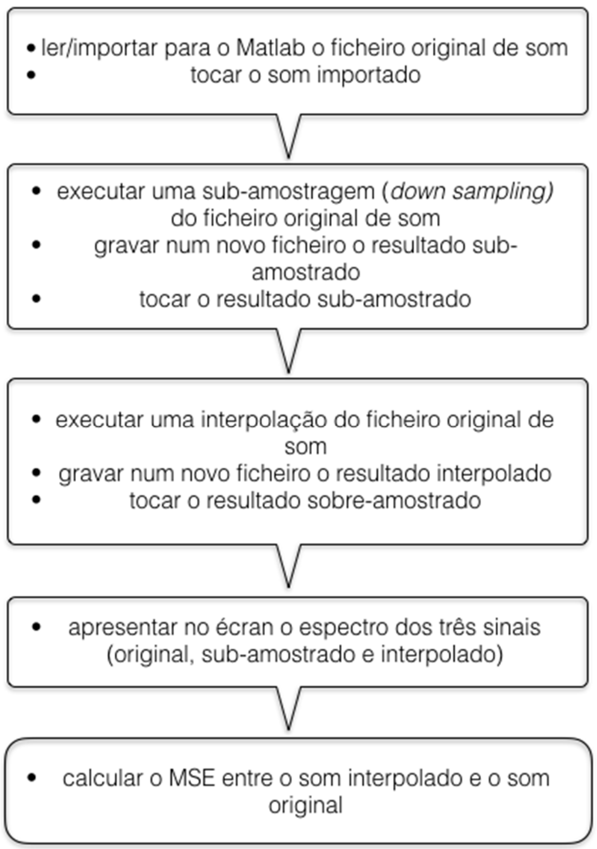
```
>>quant_uniform('inputSound.wav', 'outputSound.wav', 256)
```

Run the script twice using 256 and 16 quantisation levels. Compare the results among them and also individually against the original sound in terms of perceptual and objective quality. Compare the quantisation errors obtained.

Report should be delivered up to 28 February in Moodle.

Annex 1 - Flowcharts of the Matlab scripts

`amostragemInterp_semFiltro.m`



`quant_uniform.m`