IST, Digital Signal Processing, Lab. #1 – Sampling
Pedro M. Q. Aguiar[1]
January, 2012

In this lab., we will become familiarized with the MatLab© software for the manipulation of digital signals, in particular acoustic ones, emphasizing the illustration of the impact of sampling. Between brackets ({`command`}), you will find suggestions of relevant commands for the proposed tasks. Naturally, you should use the system help to obtain the description of how to use the command {`help command`}.

1. We want to generate discrete-time signals $x[n]$, by sampling 1-second duration intervals of continuous-time signals $x_c(t)$, *i.e.*,

$$x[n] = x_c\left(nT_s\right), \quad 0 \leq nT_s \leq 1, \quad \text{with} \quad T_s = \frac{1}{F_s},$$

where the sampling frequency should be adequate for a wide range of acoustic signals, *e.g.*, F_s =20KHz.

For sinusoidal and square waves of several frequencies (*e.g.*, 100Hz, 440Hz, 1KHz), do the following:

a) Generate the corresponding discrete-time signal. {`sin`}, {`square`}

b) Plot (segments of) the signal. Comment. {`plot`}

c) Listen the signal. Comment. {`soundsc`}

2. A sinusoidal chirp, also termed sweep, is a signal $\sin\phi(t)$, whose instantaneous frequency, defined as

$$f_i = \frac{1}{2\pi}\frac{d\,\phi(t)}{d\,t},$$

varies with time. It is commonly used in sonar, radar, and spread spectrum communications.

a) Generate a discrete-time signal by sampling with F_s =20KHz a 2-second duration chirp whose instantaneous frequency increases linearly from 1KHz to 8KHz. {`chirp`}

b) Listen the signal. Comment.

c) Repeat a) and b), now with F_s =10KHz. Comment.

d) Repeat a) and b), now without using the command `chirp`, *i.e.*, generate the signal by using the command `sin` and the appropriate phase. Comment. (If you wish, generate and listen other sinusoidal chirps, with nonlinear instantaneous frequency, *e.g.*, itself a sinusoid, etc.)

3. Impact of an anti-aliasing filter.

a) Load the sound file *romanzasmall*. {`wavread`}

b) Listen the signal.

c) Re-sample the signal at one-tenth the original sampling rate. {`:`}

d) Listen the re-sampled signal. Comment, comparing with b).

e) Filter the original signal with a low-pass filter with cut-off frequency of $\pi/10$.
{ `>> filtered_signal = filter(fir1(100, 0.1), 1, original_signal);`}

f) Repeat b), c), and and d), now with the filtered signal. Comment.

[1]Parts of this document are based on a lab. assignment authored by João Sanches.