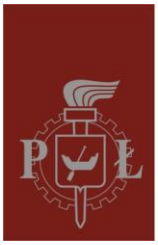

Politechnika Łódzka
Instytut Elektroniki

SIGNAL PROCESSING

Laboratory #4:

Plotting signals in Python (time scale and amplitude resolution)

M. Kociński, P. Strumiłło

Medical Electronics Division
Institute of Electronics

PURPOSE:

Plotting signals in Python

TASKS:

1. Plotting sampled sinusoidal signals in a correct time scale:
   a. Write a script defining N=2000 samples of a harmonic signal $x(t)=A\sin(2\pi f_x t)$, where A=10, f_{x1}=10 Hz, sampled at a rate of f_s=1000 Hz. Plot this signal in an interpolated and sampled version using the `plot` command (see: plot?)

   b. On a single figure, plot the sum of three sinusoids of equal amplitudes A=10 and the following frequencies: f_{x1}=10 Hz, f_{x2}=20 Hz, f_{x3}=25 Hz sampled at a rate of f_s=1000 Hz. Plot first N=2000 samples of the sum of the defined sinusoids.

2. Load `ecg_mit.mat` signal available from link:
   hwww.eletel.p.lodz.pl/pstrumil/sig_proc/signals.rar
   a. Plot first N=2000 samples of this ECG signal in a correct time (in seconds) and amplitude (in mV) scale, given the sampling rate is fs=360 Hz and an 11-bit analog to digital (A/D) converter was applied to sample and code the ECG signal recorded in the voltage range of -5mV ÷ +5mV.

3. Write a function `quantize_ecg(vec,b)` that will simulate a lower number of bits of the A/D converter. The `vec` parameter is the vector containing the source 11-bit ECG signal and `b` is the parameter indicating by how many bits we decrease the resolution of an 11-bit ECG signal, e.g. if `b=2` the function should return an ECG signal coded with 9 bits (i.e.11-b=9)

   □10/27/2012