Basics of signal processing (1)

Prof. Febo CINCOTTI, febo.cincotti@uniroma1.it

Dept. of Computer, Control and Management Engineering

(DIAG, Via Ariosto)

Material for this section of the course

- Matlab notebooks available here:
  - https://drive.matlab.com/sharing/d5ad1819-5e50-442a-81fc-6017505d91f3

- Not a textbook, but readings for those who want to have some context:
  - Steven W. Smith
    The Scientist and Engineer's Guide to Digital Signal Processing
    https://www.dspguide.com/pdfbook.htm

Neuroengineering - Analog to Digital Conversion

Contents:

- sampling, Shannon's or Nyquist's sampling theorem, aliasing, antialiasing filters
- quantization, number of bits, q. noise, (saturation high-pass filter)

See also https://www.dspguide.com/pdfbook.htm, Chapter 3

Sampling

Try the following:

1. Choose a low F_0 (e.g. 5 Hz) and assess that you can visually interpolate the samples to recover the analog waveform
2. Increase F_0 (e.g. 30 Hz) and note that the visual interpolation becomes much harder. Nevertheless the reconstruction using the `sinc()` is successful.
3. Set F_0 just below 50 Hz (e.g. 49 Hz) and note that the visual interpolation is misleading (two low-frequency sinewaves seem to appear). Even in this case the `sinc()` reconstruction is successful.
4. Set F_0 just above 50 Hz (e.g. 51 Hz) and note that the visual interpolation is qualitatively not different from the previous case. Despite this, the `sinc()` reconstruction fails (the reconstructed sinewave seems to be at a lower frequency).
5. Set F_0 just below 100 Hz (e.g. 99 Hz) and note that the visual interpolation is misleading (a very low-frequency sinewave appears). In this case, the `sinc()` reconstruction fails in a way that is consistent with the visual interpolation (the same very low frequency sinewave is reconstructed).

```
% Generate the desired signal (sinewave)
F_0 =5;  % Hz
sig = @(t) cos(2*pi*F_0*t);  % anonymous function

t = (0:.001:1)';    % s, proxy of continuous time
x = sig(t);         % proxy of analog signal for plotting

% Sample the signal
DO_SAMPLING = true;
if DO_SAMPLING
    F_samp = 100;  % Samples/second (S/s), sampling frequeny
    T_samp = 1/F_samp;  % s, sampling interval
    t_sampled = (0:T_samp:1)';   % sampling times
    x_sampled = sig(t_sampled);  % sampled signal
    %     [x_sampled2,t_sampled2] = resample(x,t,F_samp);
end% if

% Reconstruct an analog signal from the samples
DO_RECONSTRUCTION = true;
if DO_RECONSTRUCTION
```

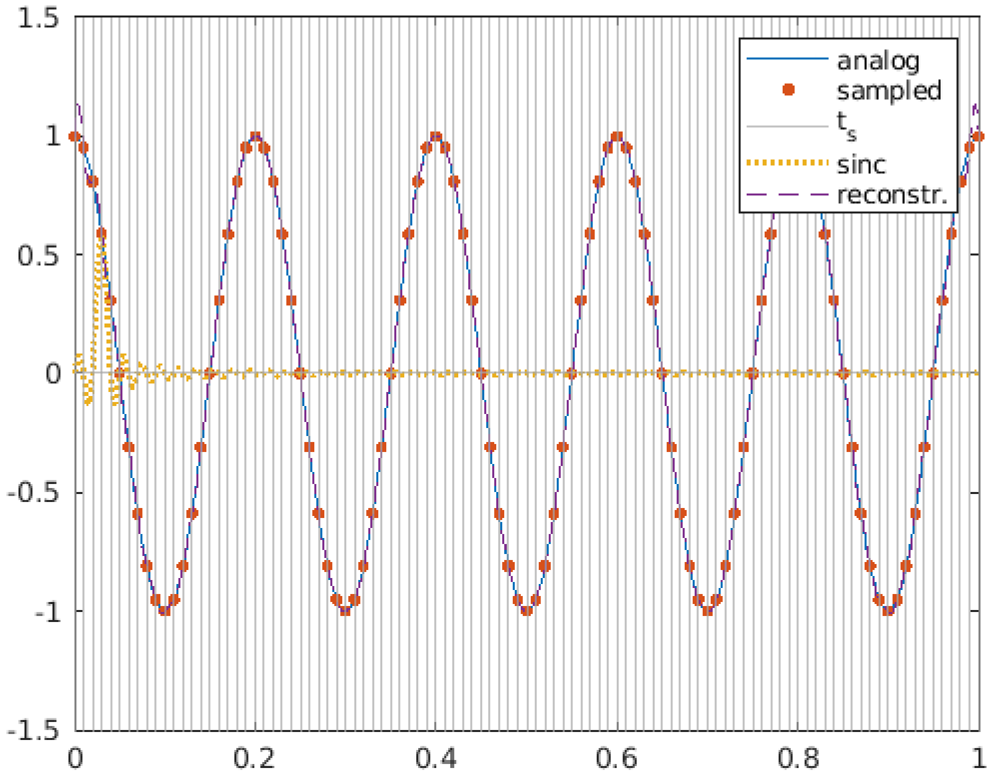
```
    % convoluted syntax to avoid for-cicles for efficiency
    % basically, it generates a sinc() on each sample
    % and sums them all
    [t_grid,ts_grid] = ndgrid(t,t_sampled);
    sincs = sinc((t_grid - ts_grid) / T_samp);
    x_reconstructed = sincs * x_sampled;
end


% === CREATE THE FIGURE ===
figure(1)
clf

% Show the analog signal
plot(t,x, "DisplayName", "analog")
yline(0, "Color", "#aaa",'HandleVisibility','off')
xlim([0,1])  % width of the figure's time axis

% Show the samples
if DO_SAMPLING
    hold on
    plot(t_sampled, x_sampled, '.', "MarkerSize",10, "DisplayName","sampled")
    hold off
    if true% show sampling timepoints
        for tt=t_sampled', h = xline(tt, "Color", "#aaa",'HandleVisibility','off'); end
        h.HandleVisibility = "on";
        h.DisplayName = "t_s";
    end
end% if

% Show the reconstructed analog signal
if DO_RECONSTRUCTION
    hold on
    if true% show one of the sinc's
        SINC_SAMPLE =4;  % which sinc will be plotted
        plot(t, sincs(:,SINC_SAMPLE)*x_sampled(SINC_SAMPLE), ':', ...
            "LineWidth",2, "DisplayName","sinc"),
    end
    % the reconstructed signal is the sum of all sincs
    plot(t, x_reconstructed, '--', "DisplayName","reconstr.")
    hold off
end
legend
```



Sampling theorem

(a.k.a. Shannon's theorem, a.k.a. Nyquist's theorem):

A continuous signal can be *properly* sampled, only if it does not contain frequency components above one-half of the sampling rate.

(*Properly* meas that the analog signal can be correctly reconstructed from its samples.)

One-half the sampling frequency is named the *Nyquist frequency.*

Failing to sample in the conditions of the Shannon's theorem, produces a phenomenon called **aliasing**, i.e. the appearance of "ghost" frequencies below the Nyquist frequency, which are in fact *mirrored* from higher frequency bands. In this example ($f_{Nyquist} = 50Hz$) , a sinewave at 51 Hz is reconstructed as a sinewave at 49 Hz, and a sinewave at 99 Hz is reconstructed as a sinewave at 1 Hz.

Sampling a signal containing (non-interesting) components with arbitrarly high frequency, requires an *analog* low-pass filter to condition the analog signal (**anti-alias filter**) so that for a given sampling frequency the Shannon's theorem is fulfilled. By doing so, the high frequency component is lost, but at least it does not contaminate through aliasing the useful signal components at lower frequencies.

-------------------------------- 01/04/2020 ---------------------------