Lab 1. Digital Signal Processing of Analog Signals

Objectives: Ideal C/D conversion. D/A conversion by zero-order hold. Digital Interpolation. Effect of oversampling in D/A conversion.

Set-up:

1. Copy the folder *PDS_practical1* to the folder *Alumnos*. Set folder *Alumnos> PDS_practical1* as working folder in Matlab. The following files must be in this folder:

`parametros11.m`	`parametros12.m`	`parametros13.m`
`practical1.m`	`practica12.m`	`practica13.m`

- File *parametros 1x.m* has the parameters of simulation *#x*.
- File *practica1x.m* runs simulation *x* and plots the results in a set of figures.

2. Add to Matlab path the folder *PDS_practical1_LIB,* which contains a library with all routines needed during the sessions. *File>Set Path>Add Folder,* and then *Close.*

1.1. Ideal C/D conversion

Let us consider the system in Fig. 1, where the continuous-time signal *xc(t)* is generated and converted to digital by using an ideal D/C converter.

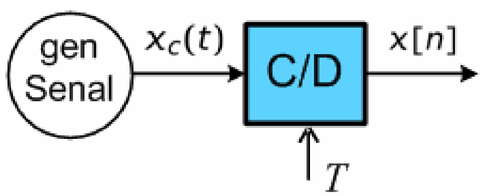

Figure 1. System "Sistema_11".

System parameters for simulating system "sistema11" are included in the file *parametros11.m*, and listed in the top of next page.

Run a simulation for the system "ejecutando" using the script *practica11*. The representation of the signals *xc(t)* and *x[n]*, both in time and frequency domains, should appear in a set of figures.

Question 1. With the considered set of parameters: is Nyquist criterion met? What's the minimum sampling frequency that can be used to sample *xc(t)* without aliasing?.

meets Nyquist? (yes/no) [ ] Minimum frequency = [ ]

List 1. *parametros11.m*

```
%---------------------------------------------
% Parametros 11
%---------------------------------------------

%---------------------------------------------
% System parameters
%---------------------------------------------
Fp = 20000;   %  xc(t) bandwidth (Hz)
Fs = 48000;   % sampling frequency (Hz)
T=1/Fs;       % sampling period

%---------------------------------------------
% Simulation parameters
%---------------------------------------------
longT=0.1;  % Length of simulation (in time - seconds)
formaEspectro='triangular';% Spectrum shape (triangle/flat)
%formaEspectro='plano';

%---------------------------------------------
%  Plot parameters
%---------------------------------------------
% Fraction of time-domain signals plotted (s)
tinicio= longT/2;  % window start (s)
tventana=0.0002;   % window end (s)
```

Question 2. With the set of parameters previously considered, what discrete frequency ω_n corresponds to the maximum frequency of the signal *xc(t)* from a theoretical perspective? Confirm your guess by observing this value in the plots.

Discrete freq. ω_n (normalized to π)	
Measured value (in plot)	

Question 3. Re-run the simulation for a signal *xc(t)* with bandwidth *Fp=5kHz*. Answer again Question #2 in this new situation.

Discrete freq. ω_n (normalized to π)	
Measured value (in plot)	

Question 4. Re-run the simulation for a signal *xc(t)* with bandwidth *Fp=30 kHz*. How would you explain the results now obtained?

Interpretation:

1.2. Ideal C/D conversion and real D/A conversion

Let us consider the system in Fig. 2, on which the continuous-time signal *xc(t)* is generated and converted into the discrete-time signal *x[n]*, first by using an ideal C/D converter, and then coverted back into the continuous-time domain by using a real D/A converter. See that two new signals are defined in the figure: *ximp(t)*, i.e., the output of the sequence to impulse converter, and *xret(t)*, i.e., the output of the reconstruction filter of zero-order hold.

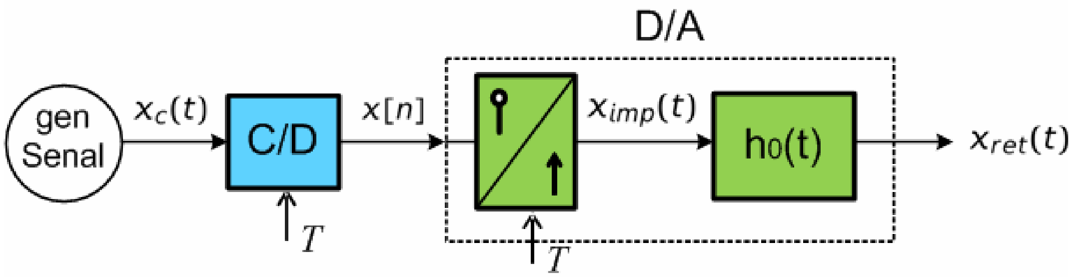

Figure 2. System "Sistema12".

Simulation parameters for the system "sistema12" are included in the file *parametros12.m*, and also listed below. See that a flat-spectrum signal is now being used.

List 2. *parametros12.m*

```
%-----------------------------------------------
% Parametros 12
%-----------------------------------------------

%-----------------------------------------------
% System parameters
%-----------------------------------------------
Fp = 20000;  % Signal Bandwith (Hz)
Fs = 48000;  % Sampling frequency (Hz)
T=1/Fs;      % sampling period (s)

%-----------------------------------------------
% Simulation parameters
%-----------------------------------------------
longT=0.1;  % Simulation length (s)
%formaEspectro='triangular';
formaEspectro='plano';

%-----------------------------------------------
% Graphical representation parameters
%-----------------------------------------------
% Window size for time-domain signals
tinicio= longT/2;  % window start (s)
tventana=0.0002;   % window end(s)
```

Run the simulation for the system "sistema12" using script *practica12*. Three figures should pop-up: In the first one, a portion of the 4 continuous-time signals in Figure 2 is depicted. In the second and third ones, the spectrums of these signals are represented in linear scale and log-scale, respectively. In the plots of the spectrums of *ximp(t)* and *xret(t)*, the frequency response of the zero-order hold filter is superimposed in red color.

Question 5. Seeing the spectrum of *xret(t)*: What's the frequency on which the first image replica *begins*? What's the *center* frequency for the image replica? Are these values coincident with those predicted by theory?

Freq. (Hz) image replica begins (theory)	
Center freq. (Hz) image band (theory)	
Freq. (Hz) image replica begins (plot)	
Center freq. (Hz) image band (plot)	

Question 6. According to theory: What's the attenuation caused by the zero-order hold filter at the beginning of the image replica? Double-check that this result is coincident with that deduced from the plot of the spectrum of *xret(t)*.

Attenuation (dB) theory	
Attenuation (dB) plot	

Question 7. Re-run the simulation for a signal *xc(t)* with bandwidth $Fp=5kHz$. As in Q5-Q6, fill the blanks with the requested information.

Freq. (Hz) image replica begins (theory)	
Center freq. (Hz) image band (theory)	
Freq. (Hz) image replica begins (plot)	
Center freq. (Hz) image band (plot)	
Attenuation (dB) theory	
Attenuation (dB) plot	

1.3. Ideal C/D conversion & D/A conversion with oversampling

We now address the effect of oversampling on D/A conversion. Let us consider the system in Fig. 3, where we have now included an interpolation by L stage prior to D/A conversion. An analog anti-imaging filter is also added in order to complete the system functionality.

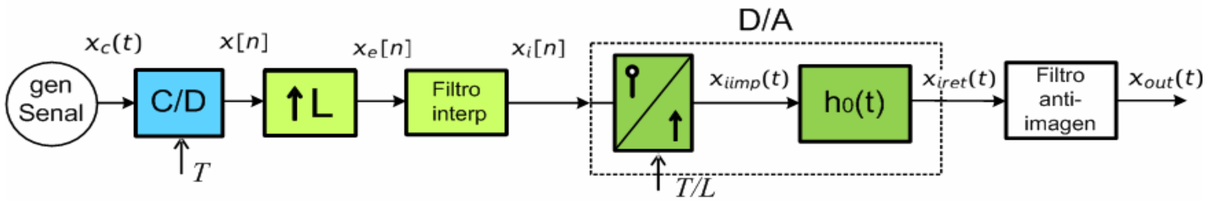

Figure 3. System "Sistema13"

Simulation parameters for the system "sistema13" are included in the file *parametros13.m.* These parameters are listed below (in addition to those previously listed in the last section).

List 3. *parametros13 .m*

```
%-----------------------------------------------
 Parameters 13 (besides those in parametros11)
%-----------------------------------------------
%-----------------------------------------------
%  System parameters
%-----------------------------------------------
L=2;          % Oversampling factor

% Specifications for the interpolation filter
ordenInterp=150; % filter order

% Anti-imaging filter specifications
Rp=3;            % Band pass ripple: dB (+-)
Rs=30;           % Reject band ripple: (dB)
Fpaso=Fp;        % Band pass limit frequency
Fstop=L*Fs-Fp;   % Reject band limit frequency
```

Run the simulation for the system "sistema13" using script *practica13*; three figures should show up.

Three figures should pop-up: In the first one, a portion of the 6 continuous-time signals in Figure 3 is depicted. In the second and third ones, the spectrums of these signals are represented in linear scale and log-scale, respectively. In the plots of the spectrums of *ximp(t)* and *xret(t)*, the frequency response of the zero-order hold filter is superimposed in red color.

In the plots of the spectrum of *xout(t)*, the frequency response of the anti-imaging filter is also superimposed in red color.

Question 8 Describe the main differences between the signals *xret(t)* in system "sistema12" and *xiret(t)*, both in time and frequency domains.

Explanation:

Question 9. If we were to choose a larger value of *L*, would it be beneficial or detrimental for the reconstruction of the signal? Verify your guess by running the simulation with *L=4*.

Explanation:

Question 10. Type `help expande` to see the header of the function in charge of expanding the signal. Following these guidelines, write your own Matlab function "expande2" that performs the expansion of a signal by a factor L. Check that it works with a toy example, and then add it up to the library. Change the instances to "expande" in "sistema13" to use your newly designed function. Run the simulation of the system ""sistema13" to see that everything works fine.

Matlab code for function expande2.m