

2. Transmisión en Banda Base

Sesión 3

1

Sesión 3

Sesión 3: Análisis de la señal muestreada

Conocimientos y Habilidades: Relacionar los parámetros de los tres procesos de la conversión digital a analógica con la calidad de la señal resultante. Utilización del espectrograma, una extensión de la transformada de Fourier, para analizar una señal. Uso de Octave/Matlab para diseñar filtros y para procesar señales.

Actividades	
TIE	BCD
Leer los tutoriales de procesamiento de audio y de filtrado de señales con Matlab. Leer la información sobre Red Book Audio en Wikipedia. Resolver la tarea 2.	Relacionar los números calculados en la sesión anterior con las características de la señal muestreada. Partiendo de una señal de alta calidad (por ejemplo, audio de un disco compacto), variar algunos de sus parámetros (tasa de muestreo, frecuencia máxima, bits por muestra, etc.) y escuchar la calidad de la señal. Relacionar los parámetros con el tiempo de transmisión de la señal. Utilizar el espectrograma para visualizar el efecto del muestreo en las señales.

2

Retroalimentación de la tarea 1

- What bandwidth is needed to transmit a PCM encoded signal?
 - Example: $0.5\%V_p$ for a 3.4 KHz audio signal

$$\max_error = \left(\frac{\Delta}{2}\right) = \frac{V_p}{L} \rightarrow \frac{V_p}{L} = \frac{0.5}{100} V_p \rightarrow L = 200 < 2^8$$
- Nyquist sample rate:

$$F_s = 2(3400) = 6800 \text{ Hz, then we need } 6800 * 8,$$

$$\text{bit rate} = 54400 \text{ bits/sec}$$
- With the fact that a bandlimited signal can convey two symbols per Hz
- For binary PCM, we need $54400/2 = 27200 \text{ Hz}$
- For practical applications, we sample faster than Nyquist rate, for example 8 kHz , then we generate 64000 bits/sec and the required bandwidth is 32 kHz
- Theoretical and Measured Quantization error
 - Con la Tarea 1 o ver el archivo en Matlab: SQNR.m

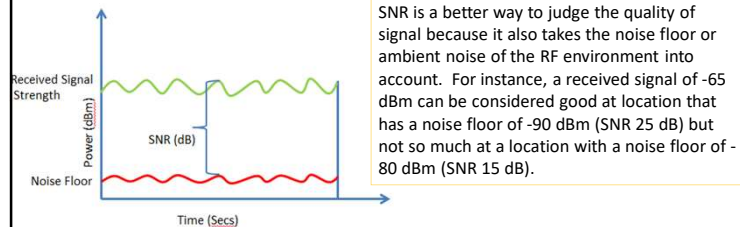
3

Logarithmic Units

- In communication we often measure ratios using log's
- The unit bel (B) is the \log_{10} of a ratio. More useful is the decibel (dB): $\frac{x}{y}$ in dB is $10 \log_{10} \frac{x}{y}$
- Examples: $2 \leftrightarrow 3 \text{ dB}$, $4 \leftrightarrow 6 \text{ dB}$
- Why dB's? Many Tx media have attenuation that is exponential in length. Thus the signal loss in dB is proportional to length.
- Reference value of 0 dB. Examples for power:
 - dBW: 0 dB = 1W, Watts
 - dBm: 0 dB = 1 mW, mili Watts
- Audio levels (rms of sound pressure) for hearing damage
 - 85 dB
 - Which NFL stadium is the loudest?
 - Noise level if you are standing 30 feet from a jet at take-off

4

SNR and Wireless Signal Strength



Your client connection

Client IP	10.92.132.86
Client MAC	3c:15:c2:9b:3c79
AP radio	2
Channel	44 (20 MHz wide)
Mode	802.11ac
Max bitrate	173 Mbps
Signal	-65 dBm

Generally, a signal with an SNR value of 20 dB or more is recommended for data networks where as an SNR value of 25 dB or more is recommended for networks that use voice applications.

Source: Cisco

5

Análisis de Tutoriales

- <http://www.findhdmusic.com/high-res-audio/free-music/>
- <https://sampleswap.org/>
- Instrucciones en Matlab para Audio
 - Revise la ayuda en Matlab para las siguientes funciones:
- `[y,Fs]=audioread(filename);`
 - `filename` = 'nombre.wav';
 - `y` es una matriz con una columna por cada canal de audio
 - `Fs` es la frecuencia de muestreo (especificada en el .WAV)

BitsPerSample	Data Type of y	Data Range of y
8	uint8	$0 \leq y \leq 255$
16	int16	$-32768 \leq y \leq +32767$
24	int32	$-2^{31} \leq y \leq 2^{31}-1$
32	int32	$-2^{31} \leq y \leq 2^{31}-1$
32	single	$-1.0 \leq y \leq +1.0$
64	double	$-1.0 \leq y \leq +1.0$

- También se pueden leer archivos MP3 (Ver ayuda en Matlab)

6

Análisis de Tutoriales

- Información sobre el archivo de audio:
 - `info = audiinfo(filename)`
- Escribir un archivo de audio:


```
load handel.mat
filename = 'handel.flac';
audiowrite(filename,y,Fs,'BitsPerSample',24,... 'Comment','This is my
new audio file.');
```

```
clear y Fs
info = audiinfo(filename) % Verificar
```
- Reproducción del audio:
 - `sound(x,fs)` % Sin normalizar
 - `soundsc(x,fs)` % Normalizado por hardware entre 1 y -1 volts

7

Ejemplo de funciones de audio

```
load handel.mat
filename='handel.wav';
audiowrite(filename,y,Fs);
% clear y Fs
[y,Fs]=audioread('handel.wav'); %Observe que y tiene valores entre 1 y -1
sound(y,Fs);
soundsc(y,Fs) % Normalización a la salida de la tarjeta de audio

[y,Fs]=audioread(filename,'native'); % En enteros la y

%Únicamente 2 segundos
samples =[1,2*Fs]; % [start,finish].
clear y Fs
[y,Fs]=audioread(filename,samples)
%[y,Fs]=audioread(filename,samples,'native'); %En enteros la y
```

Software: Audacity, **A free multi-track audio editor and recorder**

8

Convertir a enteros positivos

```
clear all
load handel.mat           %Recuperar la señal "y" en valores [-1 1]
b=16;                     % Cantidad de bits
%% Cuantizar a entero y expresión en binario
swing = (2^b-1)/2;
xq_int = round(y*swing+swing);
%xq_bin = dec2bin(xq_int); %como caracter cada digito binario
%xq_bin = int8(flip(de2bi(xq_int,2))); %como INT cada digito binario
xq_bin = de2bi(xq_int,16,'left-msb');
```

9

Frecuencias en instrumentos musicales

¿Considera adecuado un muestreo con calidad de CD a 44,100 Hz?

Instrumento	Fundamental	Armónicos
Flauta	261-2349	3-8 KHz
Oboe	261-1568	2-12 KHz
Clarinete	165-1568	2-10 KHz
Fagot	62-587	1-7 KHz
Trompeta	165-988	1-7.5 KHz
Trombón	73-587	1-4 KHz
Tuba	49-587	1-4 KHz
Tambor	100-200	1-20 KHz
Bombo	30-147	1-6 KHz
Platillos	300-587	1-15 KHz
Violín	196-3136	4-15 KHz
Viola	131-1175	2-8.5 KHz
Cello	65-698	1-6.5 KHz
Bajo acústico	41-294	1-5KHz
Bajo eléctrico	41-300	1-7 KHz
Guitarra acústica	82-988	1-15 KHz
Guitarra eléctrica (amplif.)	82-1319	1-3.5 KHz
Guitarra eléctrica (directa)	82-1319	1-15 KHz
Piano	28-4196	5-8 KHz
Saxo Soprano	247-1175	2-12 KHz
Saxo alto	175-698	2-12 KHz
Saxo tenor	131-494	1-12 KHz
Cantante	87-392	1-12 KHz

10

Ejemplo de cuantización en imágenes

```
load lena512.mat % https://www.ece.rice.edu/~wakin/images/
a=lena512;      % Lena en escala cargada tipo double
imshow(uint8(a)); %Casting uint8 bits para escala de grises
gsc1=8;
an=a/(2^8-1); % Normalizar entre 0 y 1
%an = an + 0.1*rand(size(an)); Dithering effect
b=3; % Cantidad de bits para representar
xq=round(an*(2^b-1)); %Cuantificar usando b bits;
xq=round(xq*(2^gsc1-1)/max(max(xq))); % Normalizar al máximo
% para escala de grises
figure(2); imshow(uint8(xq)) % mostrar la figura
```



11

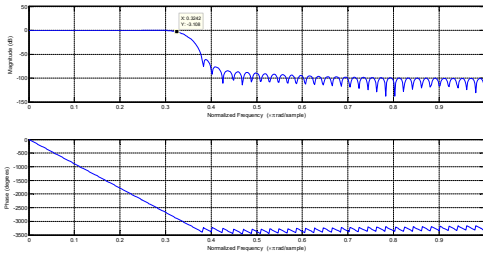
Análisis de tutoriales

- Véase el minitutorial de filtrado en Matlab

```
Fs=2000; Ts = 1/Fs; t = 0;Ts:1; % intervalo de muestreo y vector de tiempo
x = cos(2*pi*100*t)+cos(2*pi*500*t)+cos(2*pi*800*t);
f = [0 1/3 1/3 1];
m = [1 1 0 0]; o = 99;
b = fir2(o,f,m);
fvtool(b); % Espectro en frecuencia del filtro.
y = conv(x,b); % filtrado de x con filtro b o usando:
%y = filter(b,1,[x zeros(1,numel(b)-1)]); % filtrado de x con filtro b,
función filter
```

12

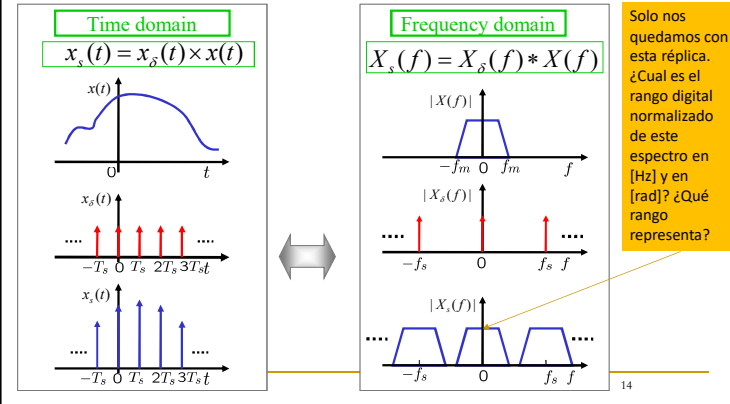
Respuesta del filtro y salida



13

Características del Espectro de una Señal Muestreada

- Sampling can be modeled as a point-by-point multiplication in the time domain by a pulse train (a sequence of impulses)



14

Características del Espectro de una Señal Muestreada

Time

$$x_s(t) = x(t) \sum_{k=-\infty}^{\infty} \delta(t - kT_s)$$

$$= \sum_{k=-\infty}^{\infty} x(t) \delta(t - kT_s)$$

$$= \sum_{k=-\infty}^{\infty} x(kT_s) \delta(t - kT_s)$$

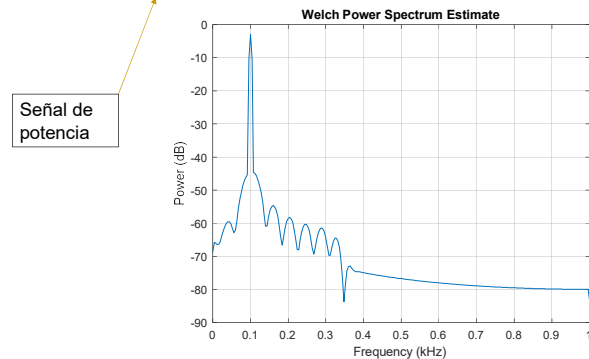
Frequency

$$X_s(f) = F_s \sum_{n=-\infty}^{\infty} X(f - nF_s)$$

15

Función para graficar espectro

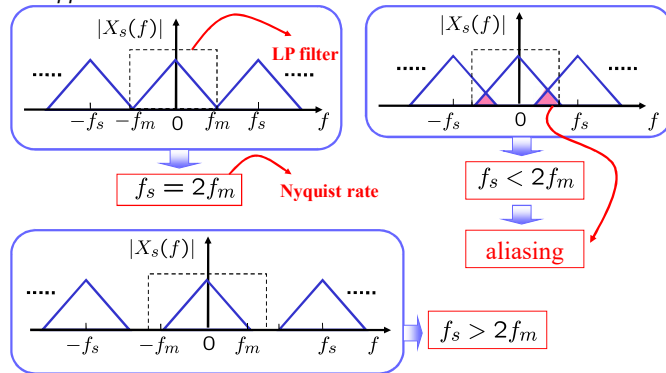
- % Análisis Espectral, Función pwelch.m
- pwelch(y, 500, 300, 500, 'one-side', 'power', Fs)



16

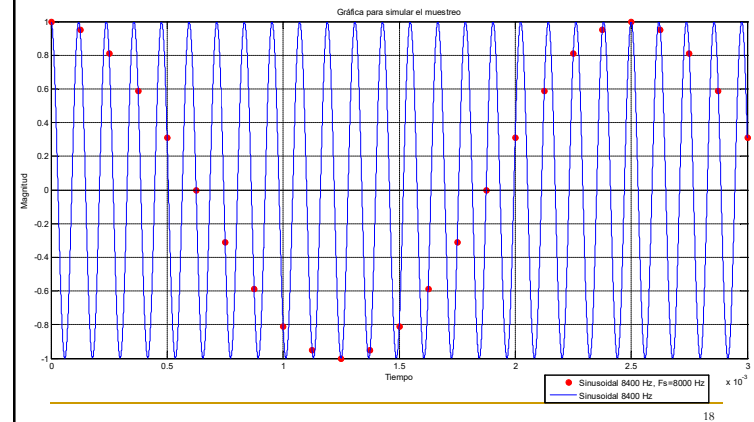
Características del Espectro de una Señal Muestreada

- **Aliasing:** If the sampling is done «too slowly», aliasing effect appears.



17

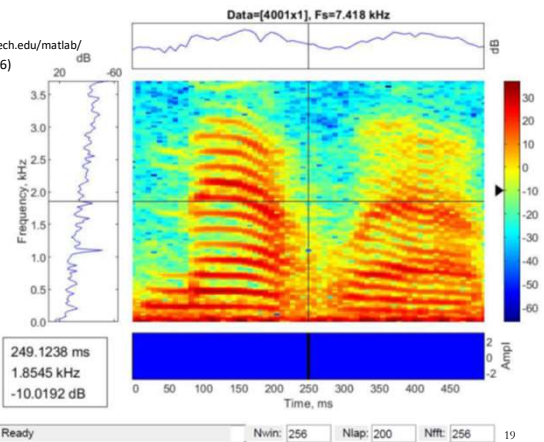
Ejemplo de "Aliasing"



18

Espectrograma

- ¿Qué es?
- Spectrogramdemo
 - <http://www.dspfirst.gatech.edu/matlab/>
- Spectrogram (Matlab 2016)



19

Matlab Exercise: Spectrogram View of Dial Tone Signal

- Create a signal, sampled at 4 kHz, that resembles dialing all the keys of a digital telephone. Create a Matlab script with this code:

```
Fs = 4e3; Ts=1/Fs; tones = [];
t = 0:Ts:0.5-Ts;
ver = [697 770 852 941];
hor = [1209 1336 1477];
for k = 1:length(ver)
    for l = 1:length(hor)
        tone = sum(sin(2*pi*[ver(k);hor(l)].*t))';
        tones = [tones; tone; zeros(size(tone))];
    end
end
soundsc(tones,Fs); % To hear
S = timetable(seconds(0:length(tones)-1)/Fs, tones);
```

Cjo al copiar los
caracteres tilde

- Open **Signal Analyzer**: Start the app by choosing it from the **Apps** tab
- Drag the timetable (S variable created in Matlab) to a display.
- Click Time-Frequency to add a spectrogram view
- On the **Spectrogram** tab, under **Time Resolution**, select **Specify**.
- Enter a time resolution of 0.5 second and zero overlap
- Analyze the results

20

Tarea

- Realizar la Tarea 2
 - Para completar el aprendizaje sobre la frecuencia de muestreo en audio, checa el sitio:
 - Ruido de intermodulación
 - <http://productionadvice.co.uk/high-sample-rates-make-your-music-sound-worse/>
 - Realiza las pruebas que ahí se sugieren con tu tarjeta de audio.

21

Realizar la Tarea 2

22

Análisis de la Función spectrum_Baz.m

```
function [ssf,afxs,pfxs] = spectrum_Baz(x,Ts,N,win,dp,lang,clip,pctrl)

% spectrum_Baz.m
%
% Author: Luis Miguel Bazdresch Sierra
%   Departamento de Electronica, Sistemas e Informática
%   Universidad ITESO
% Modified by: Omar Longoria 2012, for Matlab version
%
% Finds the spectrum of a signal, using the Fast Fourier Transform. Originally
% based on 'plotspec.m' from the book Telecommunications Breakdown, by Johnson
% and Sothares, this function allows control over many of the fft and plotting
% parameters, and adds phase spectrum plotting.
%
% Input arguments:
%
% x : signal
% Ts : sampling period (real > 0)
%     0.5 : (default)
%     y : Ts = y
% N : fft size (integer > 0)
%     0 : (default) N = 2*nextpow2(length(x))
%     y : N = y
% win : window input data (0 - 1)
%       0 : (default) no windowing
%       1 : hanning window
%       2 : kaiser window
% dp : control plotting (0 - 4)
%       0 : (default) no plotting
%       1 : plot mag spec and phase spec in two figures
%       2 : plot mag spec only
%       3 : plot phase spec only
%       4 : plot mag spec and phase spec in one figure

% lang : language of plot labels
%       0 : (default) english
%       1 : spanish
%       -1 : do not write plot labels
% clip : clip amplitudes before finding phase (0 - 100)
%       0 : (default) no clipping
%       y : clip to zero if amplitude is less than y% of max
%           amplitude
% pctrl : which command to plot with
%        0 : (default) use stem
%        1 : use plot
%        2 : use semilogy
%
% output:
%
% ssf : vector of frequencies
% afix : magnitude spectrum
% pfxs : phase spectrum
%
% Spectrums may be plotted, for example, with plot(ssf,afxs)
```

Eeng 360 23