



PERCEPCIÓN COMPUTACIONAL

Gonzalo Pajares

Práctica 10

1) Generación, grabación y reproducción de sonido

- a) Grabar la secuencia “**Percepción Computacional**” utilizando el código siguiente. Reproduciendo el sonido grabado mediante **play**. La señal queda almacenada en el vector (variable) **x**. Esta señal se puede representar gráficamente mediante la función **plot**. Mediante este procedimiento se crea el objeto *recObjeto* con diferentes propiedades, que pueden visualizarse. Una de ellas es la frecuencia de grabación F_s , que se obtiene como sigue: $F_s = \text{recObjeto.SampleRate}$. Tras lo cual se puede grabar en un fichero (.wav o de otro tipo) mediante la función **audiowrite**.

```
% Graba la voz durante t segundos.
t = 5;
recObjeto = audiorecorder;
disp('Comienza grabación')
recordblocking(recObjeto, t);
disp('Finaliza grabación');

% Reproducir grabación.
play(recObjeto);

% Almacenar la señal en un vector x.
x = getaudiodata(recObjeto);

% Dibujar sonido.
figure; plot(1:size(x,1),x);

% Grabar la señal en un fichero wav
Fs = recObjeto.SampleRate; % frecuencia de muestreo
audiowrite('MiVoz.wav',x,Fs);
```

- b) Grabar otros dos ficheros (PC11025.wav y PC22050.wav) más a las frecuencias 11025 Hz y 22050 Hz. Tras la grabación, utilizar la función $[y1, Fs] = \text{audioread}('PC11025.wav')$; $[y2, Fs] = \text{audioread}('PC22050.wav')$ y reproducir ambas mediante **soundsc**(y1, Fs) y la misma función para y2.

2) Generación de ruido y filtrado

- a) Leer el fichero **PC11025.wav**, grabado a la frecuencia $F_s = \text{recObjeto.SampleRate}$ como sigue $[y1, Fs] = \text{audioread}('MiVoz.wav')$ y reproducirlo mediante **soundsc**(y, Fs). Añadir a la señal ruido aleatorio como sigue, representándola en pantalla con **plot**.

```
yruidosa = y + 0.005.*randn(size(y,1),1);
soundsc(yruidosa, Fs);
figure; plot(1:size(yruidosa,1),yruidosa);
```

- b) Diseñar un filtro de Butterworth filtrándola posteriormente como sigue, volviéndola a reproducir y dibujar

```
[b,a] = butter(8,0.1,'low');
yfiltrada_butterworth= filtfilt(b, a, yruidosa);
soundsc(yfiltrada_butterworth, Fs);
figure; plot(1:size(yfiltrada_butterworth,1),yfiltrada_butterworth);
```

3) Características de la señal

- a) Leer el fichero **PC11025.wav**, a la frecuencia de 11025 kHz con $[x, Fs] = \text{audioread}(\text{'fichero'})$. Seleccionar de la señal x las muestras comprendidas entre $y = x(10000:1:10010)$, visualizar su contenido a través del espacio de trabajo de Matlab y calcular los pasos por cero y crear una variable indicando dónde se han producido dichos tránsitos. Comprobar el resultado marcando los pasos por cero con la función plot de Matlab que dibujará la señal y , así como sus pasos por cero. Realizar el mismo proceso para la señal $y = x(10000:1:10200)$, sin comprobar el resultado.
- b) Programa de Matlab: **Caracteristicas.m**.

Leer el fichero **PC11025.wav**, a la frecuencia de 11025 kHz con $[x, Fs] = \text{audioread}(\text{'fichero'})$. Definir una ventana de Hamming con $L = 240$ utilizando la función de Matlab $w = \text{hamming}(L)$; definir el solapamiento para las ventanas como $a = 60$. Procesar la señal por tramos calculando su **magnitud, energía y entropía** (en sus dos modalidades).

4) Correlación cruzada

- a) Programa de Matlab: **CorrelacionCruzada.m**.

Leer el fichero **PC11025.wav**, a la frecuencia de 11025 kHz con $[x, Fs] = \text{audioread}(\text{'fichero'})$. Definir una ventana de Hamming con $L = 240$ utilizando la función de Matlab $w = \text{hamming}(L)$; definir el solapamiento para las ventanas como $a = 60$. Calcular la correlación cruzada mediante la función de Matlab: $c = \text{xcorr}(w,s)$ donde s es la señal en cada tramo.

- b) Programa de Matlab: **Reconocimiento.m**
Seleccionar uno de los ficheros de sonido contenidos en la carpeta Reconocimiento y ejecutar el programa.

5) Banco de filtros: Espectrograma o sonograma

Programa de Matlab: **espectrograma.m**.

Leer el fichero **PC11025.wav**. Obtener su espectrograma mediante el banco de filtros definidos por el filtro Paso Banda Butterworth de orden 4

6) Transformada de Fourier: Espectrograma o sonograma

- a) Leer el fichero **PC11025.wav**. Llamar a la función de Matlab **spectrogram**. Representar el espectrograma como se muestra a continuación

```
[y,f,t,p] = spectrogram(x);  
surf(t,f,10*log10(abs(p)), 'EdgeColor', 'none');  
axis xy; axis tight; colormap(jet); view(0,90);  
xlabel('Tiempo'); ylabel('Frecuencia (Hz)');
```

- c) Programa de Matlab: **Caracteristicas.m**.

Leer el fichero **PC11025.wav**, a la frecuencia de 11025 kHz con $[x, Fs] = \text{audioread}(\text{'fichero'})$. Definir una ventana de Hamming con $L = 240$ utilizando la función de Matlab $w = \text{hamming}(L)$; definir el solapamiento para las ventanas como $a = 60$. Procesar la señal por tramos calculando su **centroide, flujo espectral y RollOff**.