

PROBAR LOS ALGORITMOS EN MATLAB

Es conveniente utilizar Matlab para comprobar el funcionamiento de la entrada de sonido, y también para probar los algoritmos de detección de los silbidos antes de implementarlos en C.

Las siguientes funcionalidades ayudan:

Almacenar en archivo de texto y leer desde Matlab:

En el programa en C se van almacenando los datos leídos en un archivo de texto (ej: sonido.txt) desde la función waveInProc().

En Matlab se puede leer y utilizar de forma sencilla los datos de ese archivo con las órdenes:

```
» load sonido.txt
» plot(sonido);
» specgram(sonido, [], frecmuestreo);
```

Esta última orden es interesante porque muestra un espectrograma, esto es, la evolución de la distribución frecuencial a lo largo del tiempo, con lo que fácilmente podemos comprobar qué frecuencias son interesantes en cada momento.

E/S de sonido desde Matlab:

Si se tiene instalado el toolbox de adquisición de datos (daq), se pueden utilizar las siguientes funciones para leer del micrófono y enviar a los altavoces directamente desde Matlab:

soundin.m

```
function data=soundin(duration,samplerate)
% function data=soundin(duration,samplerate=8000)
% lee datos del micrófono
% duration = duracion en segundos
% samplerate = frecuencia de muestreo en Hz (8000 por defecto)
if (~exist('samplerate'))
    samplerate=8000;
elseif isempty(samplerate)
    samplerate=8000;
end

ai=analoginput('winsound');
addchannel(ai,1);
set(ai,'StandardSampleRates','off');
set(ai,'SampleRate',samplerate);
set(ai,'SamplesPerTrigger',duration*samplerate);

input('Pulse tecla para grabar');
start(ai);
waittilstop(ai,duration*1.1);
disp('Grabacion terminada');
data=getdata(ai);
delete(ai);
```

soundout.m

```
function soundout(data,samplerate)
% function soundout(data,samplerate=8000)
% envia datos a los altavoces
% data = tabla de datos entre -1 y 1
% samplerate = frecuencia de muestreo en Hz (8000 por defecto)

if (~exist('samplerate'))
    samplerate=8000;
elseif isempty(samplerate)
    samplerate=8000;
end
ao=analogoutput('winsound');
addchannel(ao,1);
set(ao,'StandardSampleRates','off');
set(ao,'SampleRate',samplerate);
putdata(ao,data);
start(ao);

duration=length(data)/samplerate;
waittilstop(ao,duration*1.1);
delete(ao);
```

ATENCIÓN: Cuando se tiene instalado un modem además de la tarjeta de sonido, es posible que el primero sea reconocido como dispositivo de audio por defecto. En tal caso la E/S no funcionaría bien. Para comprobarlo, se puede ejecutar la orden:

```
» daqhwinfo('winsound')
```

La cual devolverá una respuesta como:

```
ans =
```

```
AdaptorDllName: 'C:\MATLAB6p5p1\toolbox\daq\daq\private\mwwinsound.dll'
AdaptorDllVersion: 'Version 2.2 (R13) 28-Jun-2002'
AdaptorName: 'winsound'
BoardNames: {'Línea #0 del módem' 'Auricular #0 del módem' 'Creative
SB AudioPCI'}
InstalledBoardIds: {'0' '1' '2'}
ObjectConstructorName: {3x2 cell}
```

El dispositivo que se abrirá por defecto es el indicado en el boardName con identificador 0, en este caso la línea #0 del modem. Para que, en este caso, utilice el dispositivo 2 (la tarjeta de sonido), deberemos añadir ese valor como 2º parámetro en las funciones analoginput y analogoutput de los ejemplos anteriores:

```
ai=analoginput('winsound',2);
ao=analogoutput('winsound',2);
```