

MANIPULACIÓN DE SEÑALES DE AUDIO

Soporte de audio

Objetos de entrada/Salida de audio	
audiodevinfo	Información del dispositivo de audio
audioplayer	Crea un objeto para reproducir audio
audiorecorder	Crea un objeto para grabar audio

Controladores de hardware de sonido	
sound	Reproduce un vector como sonido
soundsc	Autoescala y reproduce un vector como sonido
wavplay	Reproduce un sonido usando un dispositivo de salida de audio de Windows
wavrecord	Graba un sonido usando un dispositivo de entrada de audio de Windows

Importar y exportar archivos de audio

auread	Lee archivos de sonido .au
auwrite	Escribe archivos de sonido .au
wavread	Lee archivo de sonido de Microsoft WAVE (.wav)
wavwrite	Escribe archivo de sonido de Microsoft WAVE (.wav)

Utilidades

lin2mu	Convierte una señal lineal a codificación μ -Law
mu2lin	Convierte archivo codificado μ -Law en una señal lineal

Ejemplo de datos de audio (archivos .mat)

chirp	Barrido de frecuencia
gong	Gong
handel	Coro aleluya
laughter	Risas de una multitud
splat	Caída (pito) y reventada (splat)
train	Silbato del tren

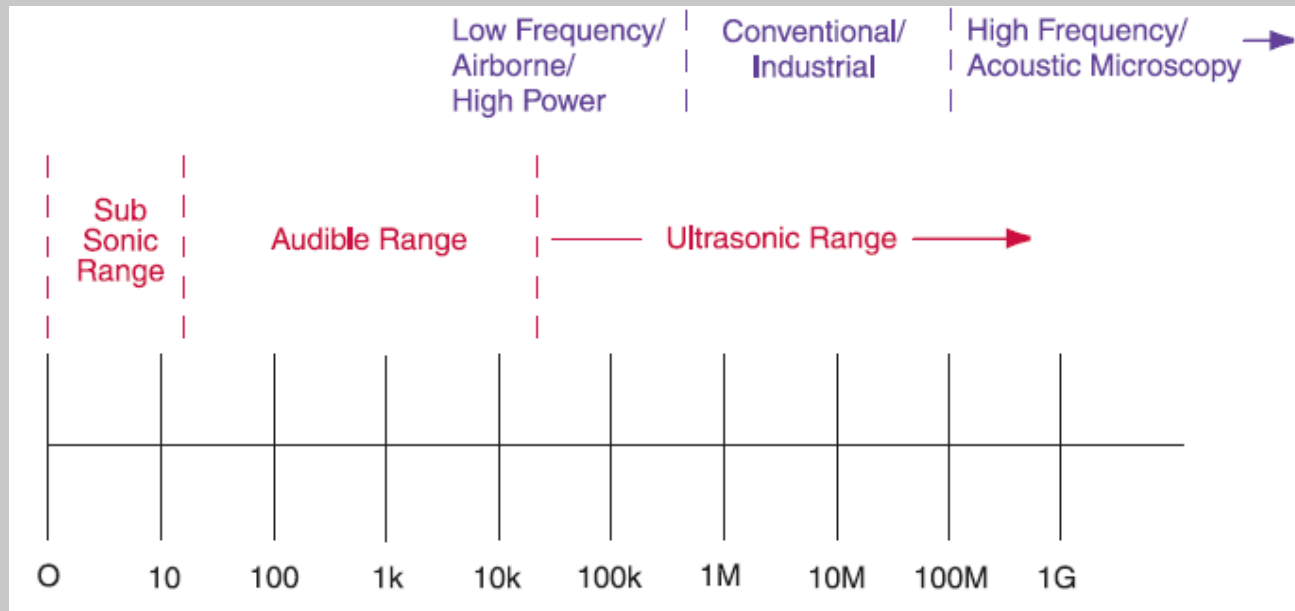
Ejemplo:

```
>> load laughter  
>> sound(y,Fs)
```

Ejercicio:

Grafique tres de las señales anteriores en el tiempo

Espectro sonoro



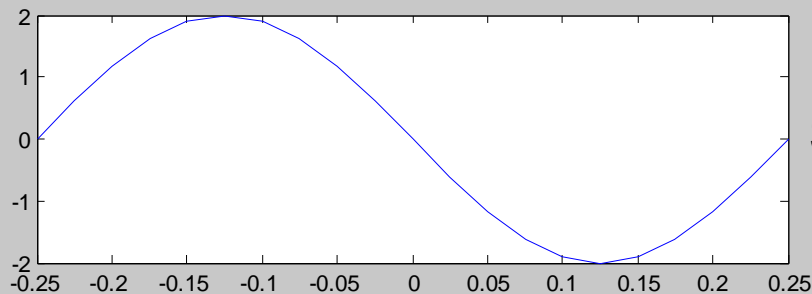
Tono: Es la cualidad que nos lo hace percibir como agudo o grave, y depende de la cantidad de veces en un período de tiempo, en el cual se repite la vibración, o sea, la frecuencia.

Tono agudo: Frecuencias mayores a 5000 Hz

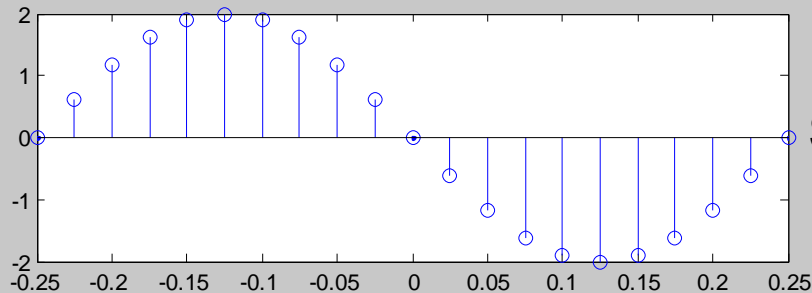
Tono grave: Frecuencias menores 300 Hz

Frecuencia de muestreo

Se define como la cantidad de muestras que se tienen de una señal en una unidad de tiempo y se mide en Hz(ciclos por segundo).



Señal continua (análoga)



Señal digital (discreta)

La frecuencia de muestreo para una señal de determinada frecuencia debe de ser mayor que el doble de la señal a esto se le conoce como la frecuencia de Nyquist y también para evitar el aliasing.

Frecuencias de muestreo típicas para señales de audio

Muestras/s	Descripción
8000	Teléfonos, adecuado para la voz humana pero no para la reproducción musical.
22050	Radio En la práctica permite reproducir señales con componentes de hasta 10 kHz.
44100	CD, En la práctica permite reproducir señales con componentes de hasta 20 kHz. También común en audio en formatos MPEG-1 (VCD, SVCD, MP3).
48000	Sonido digital utilizado en la televisión digital, DVD, formato de películas, audio profesional y sistemas DAT.

Generación y reproducción de una señal de audio

Se parte de formas de onda sinusoidales definidas por la ecuación:

$$y(t) = A \sin(\omega t + \phi)$$

para generar formas de onda compuestas por sumas de estas señales sinusoidales.

El siguiente programa permite crear una señal de audio y grabar la señal en un archivo .WAV, el cual podrá ser escuchado posteriormente con la función *sound* de Matlab (y con cualquier otro programa reproductor, como el *mplayer.exe* de Windows)

Copie el siguiente código en el editor de Matlab y córralo, verifique que los computadores tengan habilitados los parlantes.

```
clc
clear all
f=200;
fs=20*f; % verificar que sucede cuando se cambia el valor numérico
Ts=1/fs;
Amp=2;
t_total=2;
t=0:Ts:t_total;
s=Amp*sin(2*pi*f*t);
s_norm=s/max(abs(s));
Num_per=3; % cantidad de periodos a graficar
plot(t(1:(Num_per*1/f)/Ts),s_norm(1:(Num_per*1/f)/Ts))
grid on
wavwrite(s_norm,fs,'sonido_seno_1.wav');
sound(s_norm,fs) % generar sonido
```

Nota: consulte en el help de Matlab las funciones “abs”, “wavwrite” y “sound”

Interactúe con sus compañeros con el objetivo de identificar que se hace en cada línea de código (instrucción).

Ejercicio

Genere, reproduzca y visualice una señal de audio a partir de la suma de tres (3) señales sinusoidales de distinta amplitud y frecuencia.

Captura de señales de audio a partir del micrófono del PC.

wavrecord ;

```
Fs = 11025;  
y = wavrecord(num_seg*Fs, Fs, 'int16');
```

en la cual:

Fs= frecuencia de muestreo

num_seg = cantidad de segundos que estamos grabando

'int16' se usa para el numero de bits (se puede obviar)

Copie el siguiente código en el editor de Matlab y córralo, verifique que los computadores tengan habilitados los parlantes.

```
clear all
clc
prompt = {'Entre el numero de segundos','Entre la frecuencia de muestreo en Hz'};
dlg_title = 'Parametros de grabacion';
num_lines= 1;
def = {'2','8000'};
answer = inputdlg(prompt,dlg_title,num_lines,def);
n=str2double(answer(1));
f_muestreo=str2double(answer(2));
N_canales=2;
y = wavrecord(n*f_muestreo,f_muestreo,N_canales,'int16');
```

Con los valores almacenados en el Worspace, los archivos de audio se pueden reproducir con:

wavplay(y, Fs); si es que hemos colocado el 'dtype' o
sound(y,Fs) si es que no hemos utilizado el 'dtype' (ver el help)

Sí al código anterior le añadimos:

```
.  
.   
.   
[file,path] = uiputfile('*.wav','Grabar en:');  
y = wavrecord(n*f_muestreo,f_muestreo,N_canales,'int16');  
wavwrite(y,f_muestreo,strcat(path,'/',file));  
% wavwrite(y,'jkl'); % para grabar en el directorio de trabajo
```

Leer (cargar) archivo de audio .wav

wavread ;

```
y = wavread(filename)
[y, Fs, nbits] = wavread(filename)
[...] = wavread(filename, N)
[...] = wavread(filename,[N1 N2])
```

en la cual:

y= vector en el cual se almacenan (cargan) los datos muestreados

Fs= frecuencia de muestreo

nbits = número de bits utilizado en la grabación

N= se reproducen las primeras N muestras

N1,N2= se reproducen las muestras desde N1 hasta N2

Copie el siguiente código en el editor de Matlab y córralo, verifique que los computadores tengan habilitados los parlantes.

1. Desde el Command Window

```
>> y=wavread('filename'); % esta se utiliza cuando el archivo esta guardado  
en el directorio de trabajo  
>> wavplay (y)  
>> sound(y)
```

2. Puede ser desde el editor, creando um archivo .m

```
[filename, pathname] = uigetfile('*.wav', 'cargar archivo de audio');  
[y, f_muestreo,nbits]=wavread(strcat(pathname,'/',filename));  
wavplay(y)
```