

Deber Tercer Parcial

Grabar una señal de voz con frecuencia de muestreo de 16kHz guardar esa señal en un archivo y graficarla en el tiempo y en la frecuencia. A la misma señal cambiar de frecuencia de muestreo a 8kHz, 4kHz, 2kHz y 1kHz. A esas señales guardarlas y graficarlas en tiempo y frecuencia.

Contents

- Resolución
- Parametros iniciales
- Grabacion
- Grafica señal a 16kHz

Resolucion

Se cierra figuras, se limpia variables y consola.

```
close all
clear all
clc
```

Parametros iniciales

Se asignan valores a 3 parametros

1. Fs : la frecuencia de muestreo.
2. Bits : numero de bits, con los que se otendra la grabacion.
3. Canal : cuantos canales se audio de entrada se usaran.

```
Fs = 16000; % Frecuencia de muestreo 16kHz.
Bits = 16;
Canal = 1; % Un unico canal audio mono.
```

Grabacion

Usando la funcion *audiorecorder()* se graba la informacion del audio en un objeto llamado **recObj**. Usando la funcion *recordblocking()* ordenamos que grabe unicamente 5 segundos.

```
disp('Comienza grabacion REC'); % Mensaje de inicio en consola.
recObj = audiorecorder (Fs, Bits, Canal);
recordblocking (recObj, 5);
disp('Finaliza grabacion STOP'); % Mensaje de finalizado en consola.
```

Usando *getaudiodata()* convertimos la informacion grabada en **recObj** en un vector y se la almacena en la variable audio. *audio=getaudiodata(recObj)*; Se guarda el audio a Fs=16kHz con formato .wav.

```
audiowrite ('audio16kHz.wav', audio , Fs);
```

Comienza grabacion REC.

Finaliza grabacion STOP.

Grafica senal a 16kHz

Se crea un eje temporal t que permita ver la senal en 5 segundos, por ello al vector que va de 0 hasta 79999 se lo divide para $F_s = 16000$ para que el vector valla desde 0 hasta 4.9, entonces el audio estaria en funcion del tiempo.

```
t=(0:length(audio)-1)/Fs;
```

```
f=Fs/length(audio)*(0:length(audio)-1);
Y=real(fft(audio)/length(audio));
```

```
figure
subplot(1,2,1)
plot (t,audio,'Color',[0,0,1]);
xlim([0 5])
ylim([-1.25 1.25])
xlabel('Tiempo[s]');
ylabel('Amplitud')
title('Senal a 16k[Hz]')
```

```
subplot(1,2,2)
plot (f,Y,'Color',[0,0.5,1])
xlim([0 8000])
ylim([0 0.005])
xlabel('Frecuencia 16k[Hz]');
ylabel('|Y(f)|')
title('Espectro de Frecuencia')
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%primero a 8kHz
```

```

audiowrite ('audio1.wav', audio, (Fs/2));

audio(2:2:end)=[];
t1=(0:length(audio)-1)/(Fs/2);

f1=(Fs/2)/length(audio)*(0:length(audio)-1);
Y=real(fft(audio)/length(audio));

figure
subplot(1,2,1)
plot(t1,audio,'Color',[0.5,0.5,1])
xlim([0 5])
ylim([-1.25 1.25])
xlabel('Tiempo[s]');
ylabel('Amplitud')
title('Senal a 8k[Hz]')

subplot(1,2,2)
plot (f1,Y,'Color',[0.5,1,1])
xlim([0 4000])
ylim([0 0.005])
xlabel('Frecuencia 8k[Hz]');
ylabel('|Y(f)|')
title('Espectro de Frecuencia')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%primero a 4kHz

audiowrite ('audio2.wav', audio, (Fs/4));

audio(2:2:end)=[];
t2=(0:length(audio)-1)/(Fs/4);

f2=(Fs/4)/length(audio)*(0:length(audio)-1);
Y=real(fft(audio)/length(audio));

figure
subplot(1,2,1)
plot(t2,audio,'Color',[1,0,1])
xlim([0 5])
ylim([-1.25 1.25])
xlabel('Tiempo[s]');
ylabel('Amplitud')
title('Senal a 4k[Hz]')

subplot(1,2,2)

```

```

plot (f2,Y,'Color',[1 0.4 0.6])
xlim([0 2000])
ylim([0 0.005])
xlabel('Frecuencia[Hz]');
ylabel('|Y(f)|')
title('Espectro de Frecuencia')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%primo a 2kHz

audiowrite ('audio3.wav', audio, (Fs/8));

audio(2:2:end)=[];
t3=(0:length(audio)-1)/(Fs/8);

f3=(Fs/8)/length(audio)*(0:length(audio)-1);
Y=real(fft(audio)/length(audio));

figure
subplot(1,2,1)
plot(t3,audio,'Color',[0.3 1 0.7])
xlim([0 5])
ylim([-1.25 1.25])
xlabel('Tiempo[s]');
ylabel('Amplitud')
title('Senal a 2k[Hz]')

subplot(1,2,2)
plot (f3,Y,'Color',[0.5 1 0])
xlim([0 1000])
ylim([0 0.005])
xlabel('Frecuencia[Hz]');
ylabel('|Y(f)|')
title('Espectro de Frecuencia')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%primo a 1kHz

audiowrite ('audio4.wav', audio, (Fs/16));

audio(2:2:end)=[];
t4=(0:length(audio)-1)/(Fs/16);

f4=(Fs/16)/length(audio)*(0:length(audio)-1);
Y=abs(fft(audio)/length(audio));

figure

```

```

subplot(1,2,1)
plot(t4, audio, 'Color', [1 0 0.5])
xlim([0 5])
ylim([-1.25 1.25])
xlabel('Tiempo[s]');
ylabel('Amplitud')
title('Senal a 2k[Hz]')

```

```

subplot(1,2,2)
plot (f4,Y,'Color',[1 0.5 0])
xlim([0 500])
ylim([0 0.005])
xlabel('Frecuencia[Hz]');
ylabel('|Y(f)|')
title('Espectro de Frecuencia')

```





