



Memoria práctica 2

EXPERIMENTOS DE PSICOACÚSTICA

NOMBRE: Juan José Martínez Cámara



Cámara anecoica



Introducción

Con esta práctica se pretende observar de manera práctica algunos de los efectos y fenómenos relativos al funcionamiento del sistema auditivo estudiados en teoría.

Respecto a los objetivos a alcanzar, los aspectos más importantes que deberá de poner de manifiesto la práctica mediante la creación de pequeños programas, serán los siguientes:

- Comprobación práctica de las curvas isofónicas.
- Experimentación con enmascaramiento

Simulación de batidos auditivos.

II.2 Curvas isofónicas

Para establecer un nivel de referencia se reproducirá en los auriculares un tono de amplitud 1, frecuencia 1KHz y duración 1 segundo, ajustando el nivel de salida mediante el software de la tarjeta de sonido para apreciar una sonoridad lo más alta posible sin llegar a molestar. A partir de este momento supondremos que este nivel se corresponde con 94 dB de nivel de presión sonora, nivel de presión de referencia SPLref. Nota: calcular la amplitud de los tonos en función del nivel de SPL y nivel de SPL de referencia $A=10[(SPL-SPLref)/20]$ La frecuencia de muestreo se considera para todas las simulaciones 44100Hz.

Se comenzará por realizar un programa que se llamará 'lea1.m'. Dicho programa realizará dos barridos



i. Se realizará un barrido en frecuencia utilizando tonos de frecuencias normalizadas de 1 octava desde 125 hasta 8000 Hz con un nivel de presión sonora de 88 dB. Comenzará por el tono de 125 Hz y posteriormente se oirán en los auriculares 500 ms del tono de 125 Hz concatenados con 500 ms del tono de 1KHz que servirán de referencia de sonoridad (es decir, sin pausa entre el tono de 125 Hz y el de 1KHz). A continuación se repetirá la misma secuencia para el tono de 250 Hz y así hasta llegar al de 8 KHz. Todos los tonos tendrán el mismo nivel de 88 dB. Generar un fichero wav en donde se recoja el resultado del barrido en frecuencia.

```
%definimos parametros iniciales
t_fin=0.5;[x,fs]=audioread('TonolKhz.wav');%cargamos nuestro tono
SPLref=94;SPL=88;%declaramos los niveles de presion

%definimos nuestro vector de tiempos
t=0:1/fs:t_fin;

%calculamos la amplitud
A=10^((SPL-SPLref)/20);

%reajustamos el tono de 1Khz
x=x(1:length(t));

%tenemos que buscar , 125-250-500-1000-2000-4000-8000, ya que estas son
%octavas de mi funcion
f=[125 250 500 1000 2000 4000 8000];
x_final=[];
for i=1:length(f)
    p=isofon(f(i),SPL);           %hacemos la curva isofonicas
    x2 = A*cos(2*pi*f(i)*t);      %definimos nuestra señal
    x_aux = [x2 x'];              %Realizamos la concatenacion con nue
    x_final = [x_final x_aux];    %Usamos para ir almacenando, ya que
end
%Con este comando reproducimos el audio cuando ejecutemos el scrip
soundsc(x_final,fs);
audiowrite('aprt1)125-8Khz.wav',x_final,fs);
```



ii. Se realizará un barrido de nivel utilizando saltos de 6 dB desde 76 hasta 94 dB.

Se usará un tono de 1000 Hz comenzando por 76 dB y posteriormente se oirán en los auriculares 500 ms del tono concatenados con 500 ms del tono de 88 dB que servirán de referencia de sonoridad (es decir, sin pausa entre el tono de 76 dB y el de 88 dB). A continuación se repetirá la misma secuencia para el tono de 82 dB y así hasta llegar al de 94 dB. Todos los tonos tendrán la misma frecuencia de 1000 Hz. Generar un fichero wav en donde se recoja el resultado del barrido de nivel.



```
%definimos parametros iniciales
t_fin=0.5;[x,fs]=audioread('TonolKhz.wav');%cargamos nuestro tono
SPLref=88;SPL=[ 76 82 88 94 ];%declaramos los niveles de presion
f=1000;

%definimos nuestro vector de tiempos
t=0:1/fs:t_fin;

%reajustamos nuestro tono
x=x(1:length(t));

for i=1:length(SPL)
    p=isofon(f,SPL(i));           %hacemos la curva isofonicas
    A=10^((SPL(i)-SPLref)/20);    %Modificamos la amplitud en funcior
    x2=A*sin(2*pi*f*t);           %redefinimos con la nueva amplitud
    x_aux=[x2 x'];                %Realizamos la concatenacion con n
    x_final=[x_final x_aux];      %Usamos para ir almacenando, ya que
end
sound(x_final,fs);
audiowrite('aprt2) 82-94SPL.wav',x_final,fs);
```

Con el barrido en frecuencia del programa 'lea1.m' y haciendo uso de la función isofon. realícese una tabla indicando la sonoridad de cada uno de los tonos analizados indicando cual de ellos es el más sonoro justificando la respuesta en términos del comportamiento del oído humano. En la misma tabla se ha indicar (haciendo uso de las las curvas de sonoridad) la sonoridad teórica esperada –en fonios- comprobando así que el valor predicho no se aleja del que proporciona la función isofon).

Tras realizar el barrido requerido y construido la tabla, tenemos el siguiente resultado :

SPL/FREQ	125 Hz	250 Hz	500 Hz	1000 Hz	2000 Hz	4000 Hz	8000 Hz
----------	--------	--------	--------	---------	---------	---------	---------



88 DBSPL	85.247F O	92.449 FO	91.499 FO	88.888 FO	94.722 FO	99.887 FO	83.443 FO
-----------------	--------------	--------------	--------------	--------------	--------------	--------------	--------------

Observamos que la escala en cuanto a sonoridad es la siguiente :

$$4000\text{Hz} > 2000\text{Hz} > 250\text{Hz} > 500\text{Hz} > 1000\text{Hz} > 125\text{Hz} > 8000\text{Hz}$$

Con lo cual a 4000Hz tenemos máxima sonoridad y en 8000Hz tenemos la mínima sonoridad.

Imagen en 3d:

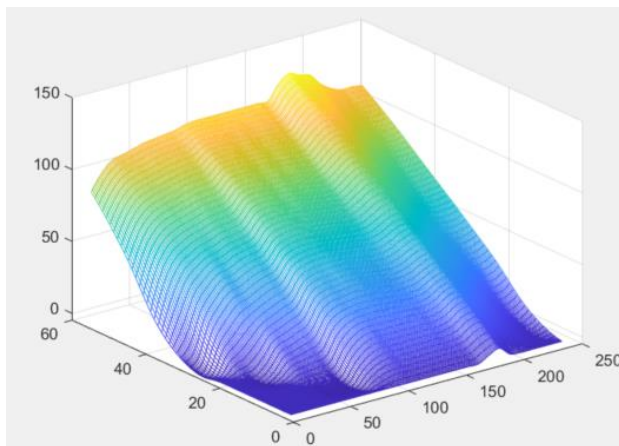
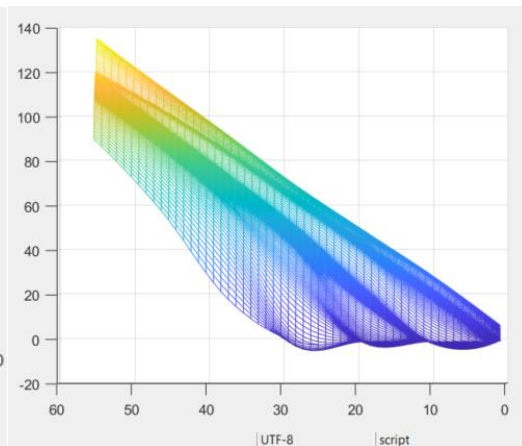


Imagen en 2d:



Con el barrido de nivel del programa 'lea1.m' funcionando correctamente, determínese aproximadamente cuantas veces más o menos sonoros son los tonos de 76, 82 y 94 respecto la sonoridad del tono de 88 dB. Realícese una tabla (tal y como se realizó en el barrido en frecuencia) indicando el valor de la sonoridad en fonos y compare con el valor teórico esperado.

Tras realizar el barrido de nivel, hemos obtenido los siguientes resultados:

Freq/SPL	76 DBSPL	82 DBSPL	88 DBSPL	94 DBSPL
----------	----------	----------	----------	----------



1000 Hz	75.5556 FO	82.2222 FO	88.8889 FO	93.3333 FO
---------	------------	------------	------------	------------

Como se puede observar que con 94 DBSPL , tenemos mas sonoridad que en cualquier otro y 76 DBSPL resulta ser el menos sonoro.

En este caso, la escala quedaría :

94 DBSPL>88 DBSPL>82 DBSPL>76 DBSPL

Imagen en 3d:

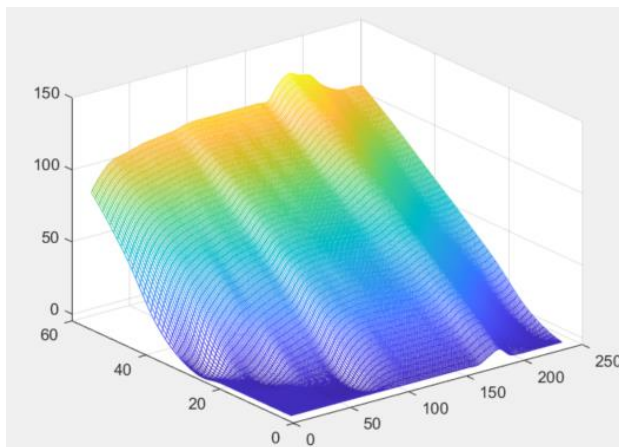
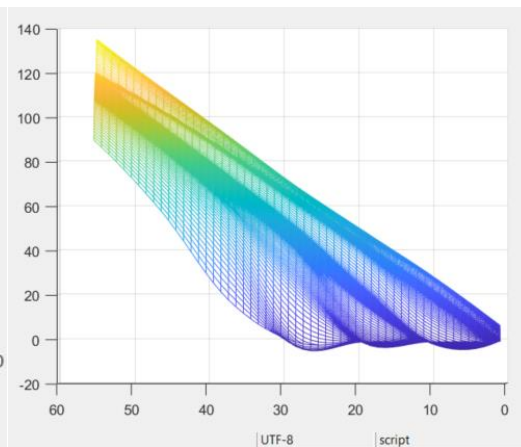


Imagen en 2d:



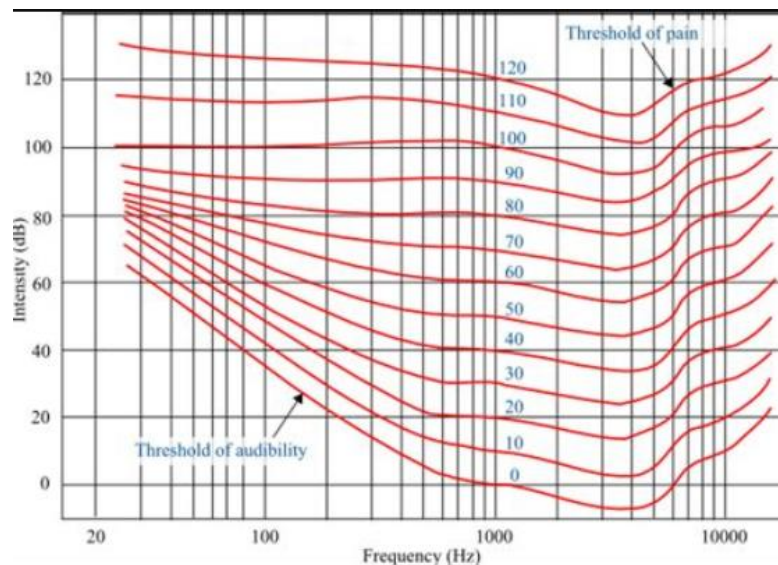
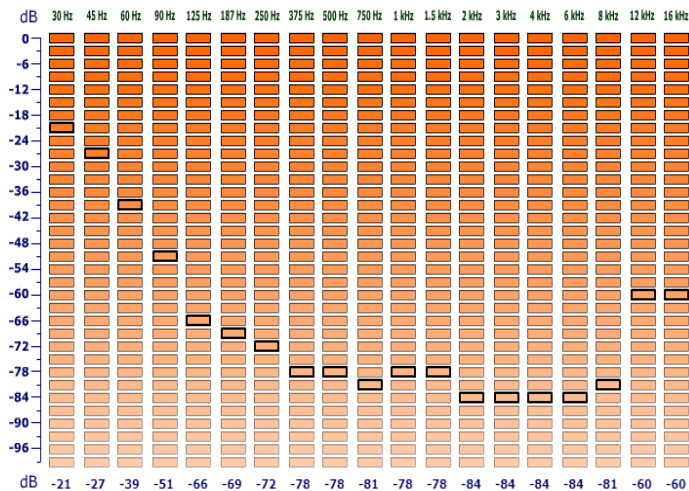
II.3 Comprobación práctica de las curvas isofónicas

Accediendo al siguiente link, realizar un test para comprobar los niveles de audiabilidad. Realizar el experimento para cada oído y obtener la curvas resultantes. <http://newt.phys.unsw.edu.au/jw/hearing.html> (no funciona en ordenadores Macintosh, pero sí en ordenadores con sistema operativo



Windows). Reportar el resultado obtenido por el alumno en la memoria de prácticas

Debemos de obtenemos una curva muy similar a la de fletcher Munson



II.4 Enmascaramiento de fuentes sonoras



Se trata de crear una función llamada 'lea2.m' que sea capaz de generar un tono (sonido enmascarante) superpuesto a un ruido de 100 Hz de ancho de banda (sonido enmascarado). Las variables de entrada serán:

f_{tono} frecuencia del tono.

SPL_{tono} nivel del tono.

f_{ruido} frecuencia central del ruido.

SPL_{ruido} nivel total del ruido.

Cada vez que se llame a la rutina 'lea2.m' deberá reproducir los sonidos enmascarante y enmascarado durante 1.5 s, y dar como salida la forma de onda conjunta (proporcionar los ficheros wav tal y como se indica a continuación).

```
%%II.4 Enmascaramiento de fuentes sonoras
function [x_final] = lea2(ftono,SPLtono,fruido,SPLruido)
%{ ... }
%definicion
SPLref = 94;fs=44100;f0=100;
t=0:1/fs:1.5;

%definicion de presiones
A1= 10^((SPLtono-SPLref)/20);
A2= 10^((SPLruido-SPLref)/20);

%generacion de tono
tono = A1*sin(2*pi*ftono*t);

%generamos el sonido enmascarado
enmascarador=A2*creaRuido(length(tono),fruido,f0,fs);
x_final = [tono' enmascarador tono'+enmascarador];%concatenamos nuestro
end
```



Cada vez que se llame a la rutina 'lea2.m' deberá reproducir los sonidos enmascarante y enmascarado durante 1.5 s, y dar como salida la forma de onda conjunta (proporcionar los ficheros wav tal y como se indica a continuación). Se creará ahora un programa 'lea3.m' que la siguiente secuencia de sonidos

- 1) ftono= 800 Hz, SPLtono= 94 dB, fruido= 1000 Hz, SPLruido= 14 dB
- 2) ftono= 800 Hz, SPLtono= 94 dB, fruido= 1000 Hz, SPLruido= 82 dB
- 3) ftono= 1200 Hz, SPLtono= 94 dB, fruido= 1000 Hz, SPLruido= 14 dB
- 4) ftono= 1200 Hz, SPLtono= 94 dB, fruido= 1000 Hz, SPLruido= 82 dB

Son 4 situaciones a considerar.

El alumno ha de reportar los ficheros wav producto de cada prueba, justificando en la memoria si se produce o no enmascaramiento. Para las 4 pruebas, el nombre de los ficheros a reportar son los siguientes:

"enmascaramiento1.wav", "enmascaramiento2.wav",
"enmascaramiento3.wav", "enmascaramiento4.wav".

Realice una discusión de los resultados y comentar si el tono es capaz de enmascarar al ruido



% { ... % }

fs=44100;

%% Primera situacion

f_tono_1=800;SPL_tono_1=94;fruido_1=1000;SPLruido_1=14;

[x_final_1] = lea2(f_tono_1,SPL_tono_1,fruido_1,SPLruido_1);

%% segunda situacion

f_tono_2=800;SPL_tono_2=94;fruido_2=1000;SPLruido_2=82;

[x_final_2] = lea2(f_tono_2,SPL_tono_2,fruido_2,SPLruido_2);

%% tercera situacion

f_tono_3=1200;SPL_tono_3=94;fruido_3=1000;SPLruido_3=14;

[x_final_3] = lea2(f_tono_3,SPL_tono_3,fruido_3,SPLruido_3);

%% cuarta situacion

f_tono_4=1200;SPL_tono_4=94;fruido_4=1000;SPLruido_4=82;

[x_final_4] = lea2(f_tono_4,SPL_tono_4,fruido_4,SPLruido_4);

%% cargamos los audiowrite

audiowrite("enmascaramiento1.wav",x_final_1,fs);

audiowrite("enmascaramiento2.wav",x_final_2,fs);

audiowrite("enmascaramiento3.wav",x_final_3,fs);

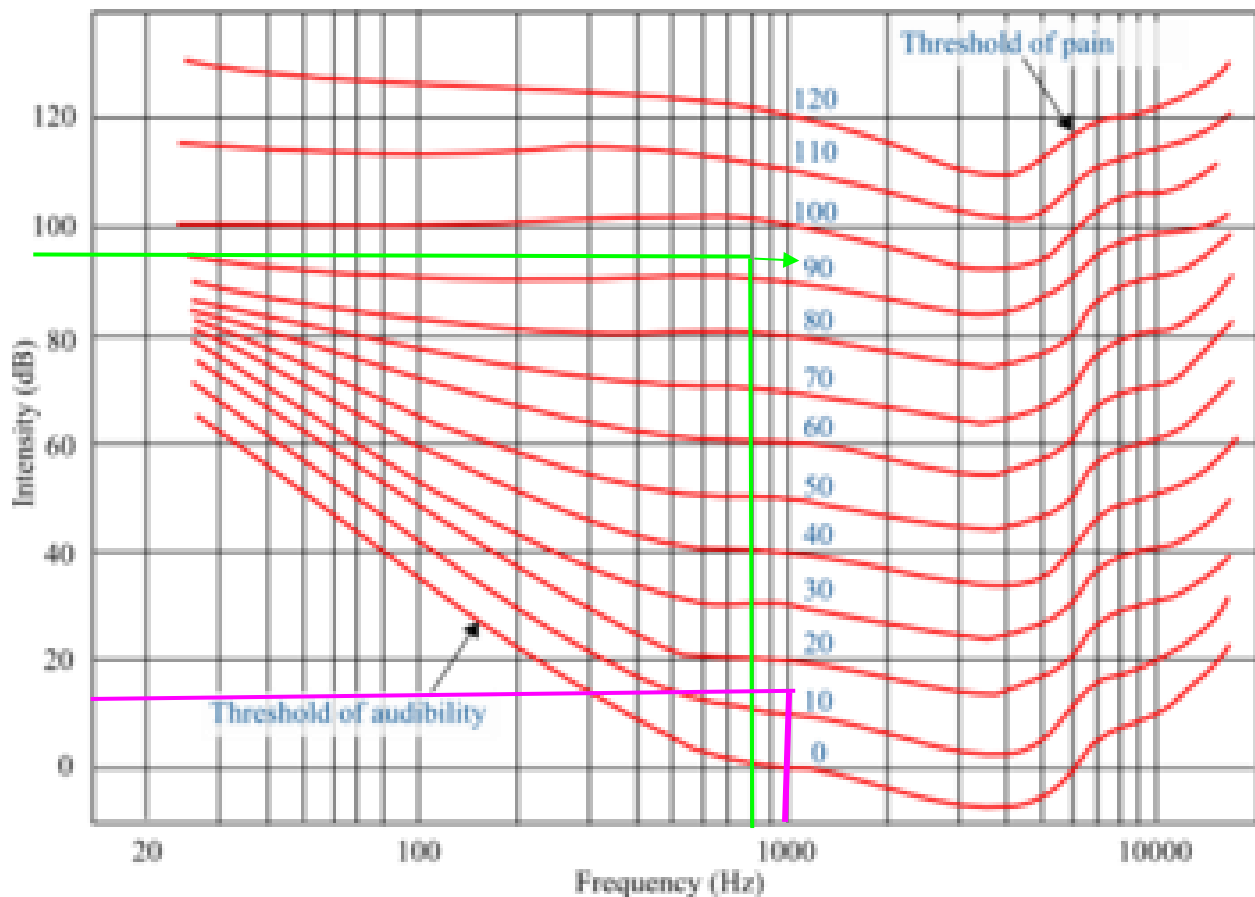
audiowrite("enmascaramiento4.wav",x_final_4,fs);



Vamos a realizar un análisis de las distintas situaciones, veamos que ocurre:

Leyenda	Tono	Ruido
Colores		

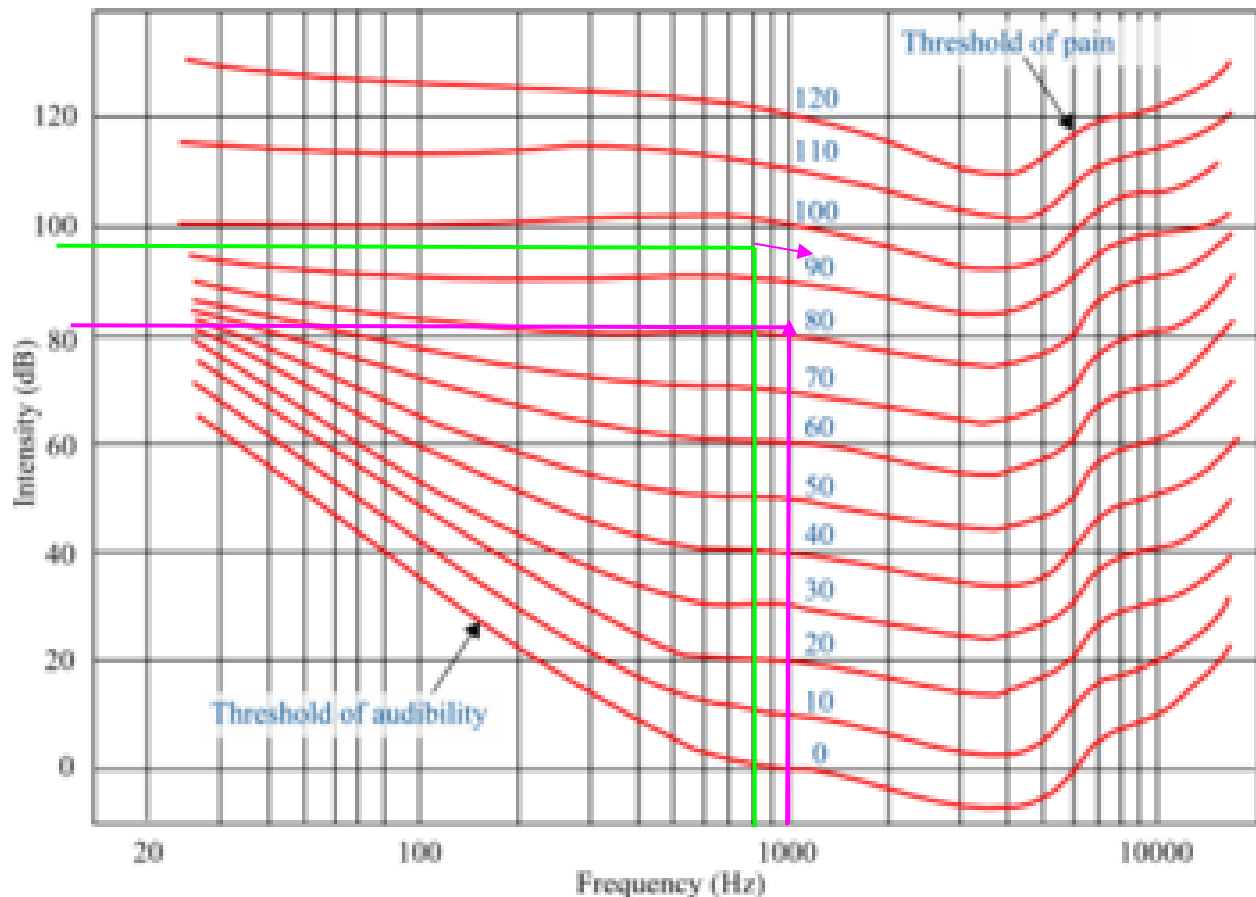
En nuestra primera situación hemos generado el archivo “enmascaramiento1.wav” teóricamente para que el tono enmascare al ruido la SPL del tono debe ser superior entre 28-21 dB. En este caso se cumple por lo que el tono enmascara al ruido. Si lo reproducimos podemos percibir únicamente un tono.



Como vemos, supera este nivel de ruido, muy por encima de los 28 dB requeridos.



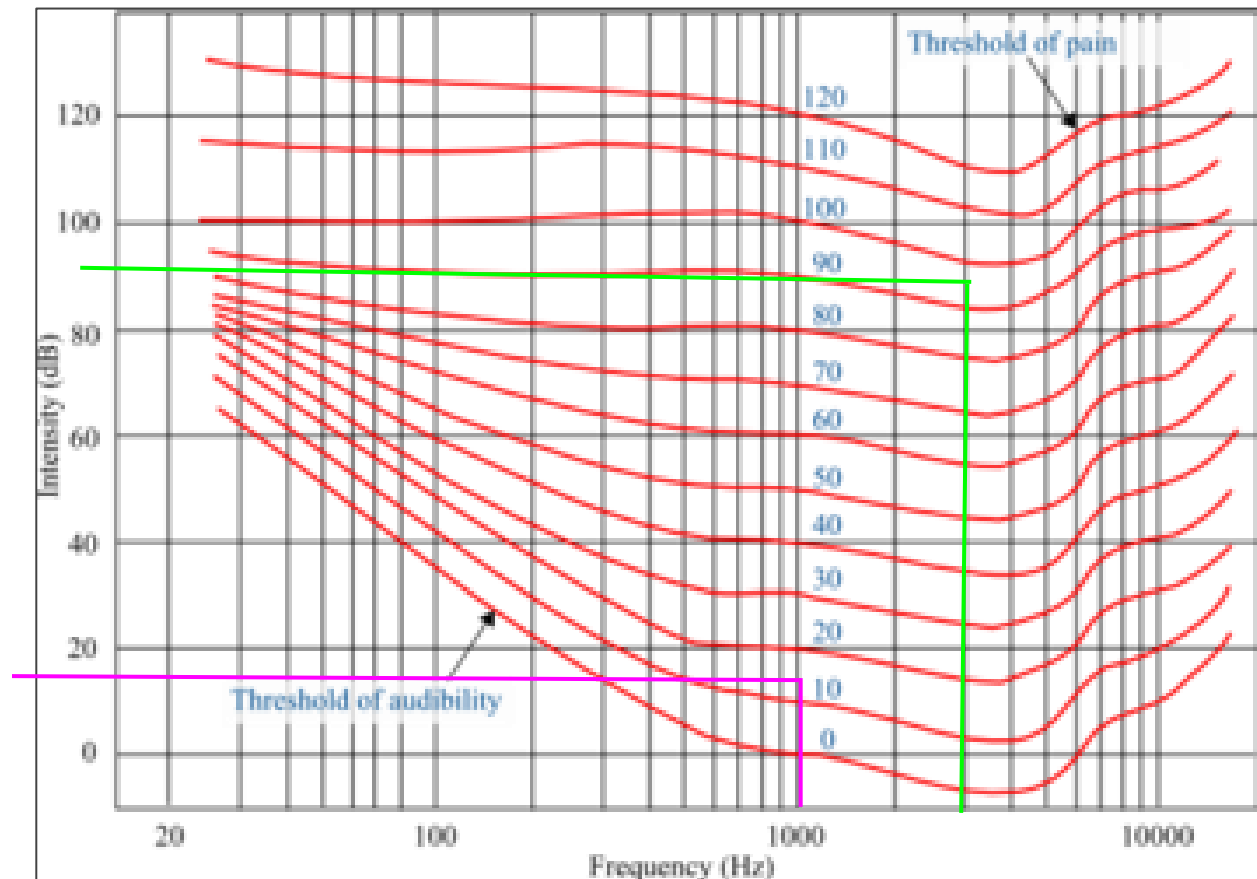
En nuestra segunda situación hemos generado el archivo “enmascaramiento2.wav” teóricamente para que el tono enmascare al ruido la SPL del tono debe ser superior entre 28-21 dB. En este caso **NO** se cumple por lo que el tono **NO** enmascara al ruido. Si lo reproducimos podemos percibir un tono + ruido.



Como vemos, supera este nivel de ruido, pero no lo supera lo suficiente como para poder enmascarado por nuestro tono principal ya que no supera la distancia mínima de los 21 dB requeridos, con lo cual al reproducirlo tendremos esa “sensación” de ruido de fondo .



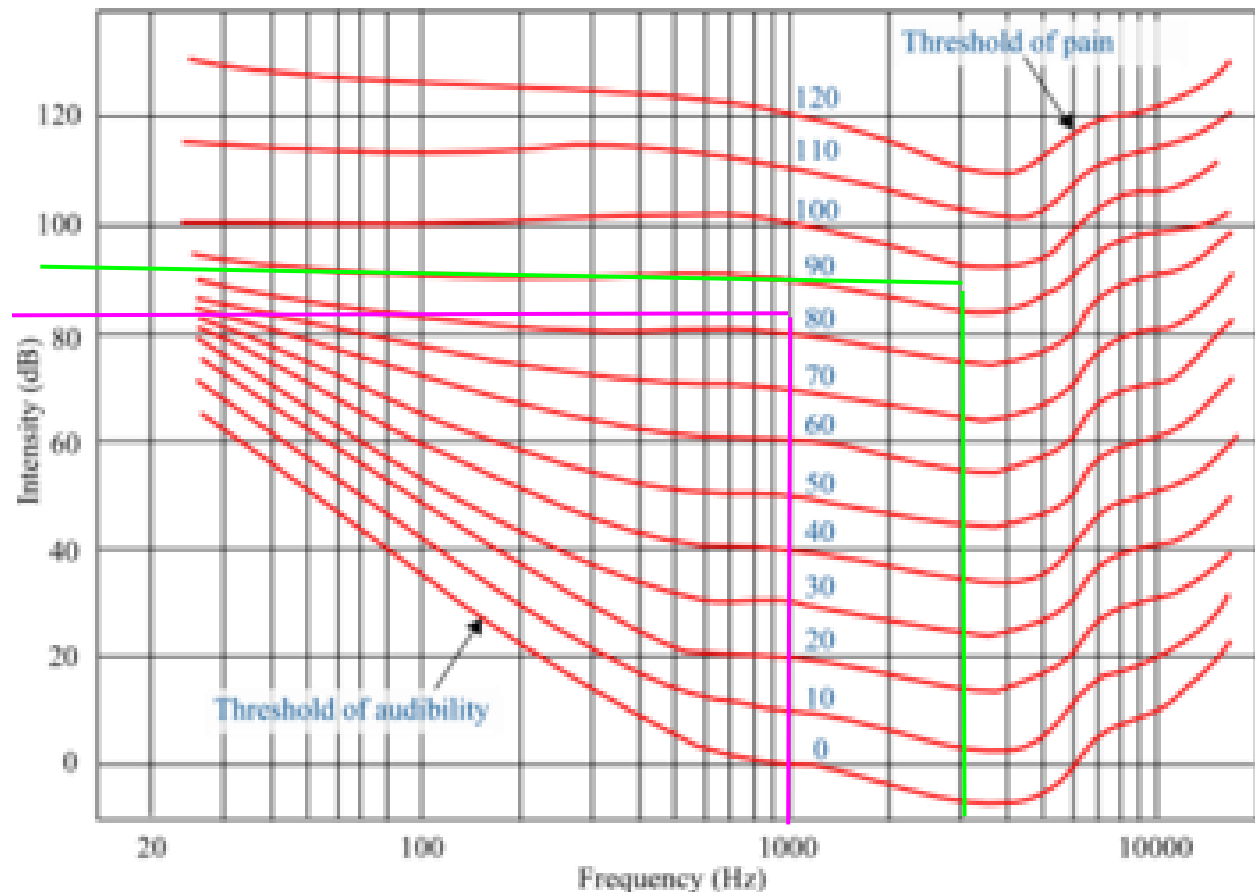
En nuestra tercera situación hemos generado el archivo “enmascaramiento3.wav” teóricamente para que el tono enmascare al ruido la SPL del tono debe ser superior entre 28-21 dB. En este caso se cumple por lo que el tono enmascara al ruido. Si lo reproducimos podemos percibir un tono.



Como vemos, supera este nivel de ruido, muy por encima de los 28 dB requeridos, con lo cual se queda enmascarado, veamos la ultima situacion de todas:



En nuestra cuarta situación hemos generado el archivo “enmascaramiento4.wav” teóricamente para que el tono enmascare al ruido la SPL del tono debe ser superior entre 28-21 dB. En este caso **NO** se cumple por lo que el tono **NO** enmascara al ruido. Si lo reproducimos podemos percibir un tono + ruido.



Como vemos, supera este nivel de ruido, pero no lo supera lo suficiente como para poder enmascarado por nuestro tono principal ya que no supera la distancia mínima de los 21 dB requeridos, con lo cual al reproducirlo tendremos esa “sensación” de ruido de fondo .



Determine experimentalmente el nivel de enmascaramiento TMN para un tono de frecuencia 1000 Hz y 94 dB de SPL y una señal ruidosa de frecuencia central 1000 Hz y ancho de banda de 100Hz.

```
fs=44100;  
ftono_u=1000;SPLtono_u=94;fruido_u=1000;SPLruido_u=SPLtono_u-35;  
x_final_u=lea2(ftono_u,SPLtono_u,fruido_u,SPLruido_u);  
%sound(x_final_u(:,3),fs);  
audiowrite("TMN_35dB.wav",x_final_u,fs);
```

Se empieza a enmascarar (significativamente) partir de que el tono se encuentre 35 dB por encima del ruido, aunque realizando una prueba subjetiva, el ruido no resulta enmascarado totalmente hasta que no se encuentra superado por 50dB el nivel del tono por encima del ruido, se adjunta también al fichero como "TMN_50dB.wav", como demostración al lector.

Considere a continuación el enmascaramiento de una señal ruidosa a un tono (NMT). Para ello considere una señal ruidosa con frecuencia cental de 1000 Hz; 100 Hz de ancho de banda y 84 dB de SPL. Genere un tono puro de 1000 Hz e intensidad variable. Encontrar experimentalmente el valor de la SPL a partir de la cual el tono es enmascarado por el ruido. Proceda de modo análogo al caso anterior y proporcione un fichero de audio wav correspondiente nombrandole "NTM_XXdB.wav", siendo XX la diferencia de nivel del ruido respecto al tono para enmascararle.

Recordar, que en las lecciones en el aula se suministro el siguiente umbral:





Vamos a realizar el enmascaramiento de un sonido con ruido:

Debemos de tener en cuenta que:

Con lo cual nuestro nivel de ruido resultante que debe ser superior al tono, estará en torno a estos valores .

```
fs=44100;
ftono_u=1000;SPLtono_u=80-6;fruido_u=1000;SPLruido_u=80;
x_final_u=lea2(ftono_u,SPLtono_u,fruido_u,SPLruido_u);
sound(x_final_u(:,3),fs);
audiowrite("NTM_6dB.wav",x_final_u,fs);
```

Como se puede observar, hemos conseguido enmascarar correctamente a nuestro tono con un nivel de ruido de 6 dB superior al tono dado

Responder a la siguiente cuestión: ¿Qué tipo de señal, tono o ruido, tiene mayor efecto enmascarante?

Sin duda, la capacidad del ruido para enmascarar, contiene un mayor efecto, que el uso de un tono y es que aqui tenemos que tener en cuenta la relación que hay con el concepto de banda critica y es el siguiente:

- Cuando tenemos un ruido de banda estrecha provoca un enmascaramiento mayor que un tono puro de misma intensidad y frecuencia.
- A medida que nuestro nivel de ruido crece, con ello, aumenta el rango de frecuencias que se ven aumentadas por el mismo ejerciendo en ellas un efecto enmascarador.
- Siempre será más fácil enmascarar un tono por un ruido si la frecuencia central del tono se encuentra superior a la frecuencia central del ruido.



II.5 Simulación de batidos auditivos

Se trata de crear una función llamada 'lea4.m' que sea capaz de crear un batido auditivo. La rutina tendrá como variables de entrada : f_c frecuencia central de batido. Δf desviación de la frecuencia central. La amplitud máxima del Batido se iguala a la unidad Cada vez que se llame a la rutina 'lea4.m' deberá dibujar la forma de onda del batido (oscilaciones de amplitud) en un intervalo de tiempo de 2.5 sg. La salida de la rutina será la forma de onda del batido (señal total y)

$$y_{total} = 2 \cos(2\pi \Delta f t) \cdot \cos(2\pi f_c t)$$

```
function [y_total] = lea4(fc,If,A)
%definimos parametros y constantes =>
fs=44100;ts=1/fs;
t=0:ts:1;
%definimos nuestra señal =>
y_total=2*A*cos(2*pi*If*t).*cos(2*pi*fc*t);
%representamos =>
plot(t,y_total);title('Batido If (Hz)');
xlabel('Tiempo(s)');ylabel('Amplitud (V)');
hold on;
end
```



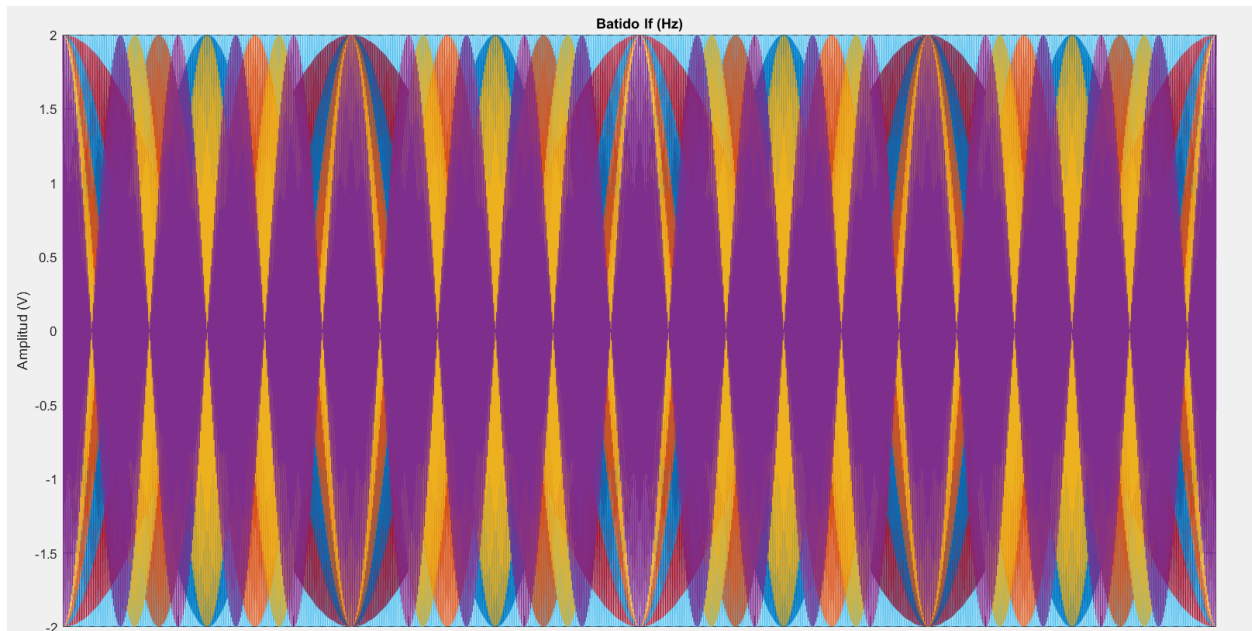
Se creará ahora un programa 'lea5.m' que irá simulando un batido auditivo con frecuencia central de 1Khz para diferentes desviaciones de frecuencia. Se usará la rutina 'lea4.m' y en pasos de 2 Hz, comenzando por $\Delta f = -10$ y finalizando por $\Delta f = +10$ se creará un fichero .wav donde se perciba el batido para los diferentes valores de Δf . Comentar, el valor de Δf a partir del cual se puede percibir el batido.

Esto hace que la frecuencia de un tono con frecuencia central 1Khz, la frecuencia del segundo tono se encuentre entre 990 Hz y 1010Hz en pasos de 2Hz.

```
%parametros iniciales =>
fc=1000;fs=44100;
A=1;
If=10;
aux=[];
%ejecucion del barrido=>
figure;
] for If=-10:2:10
    y_total=lea4(fc,If,A);
    aux=[aux y_total];%guardado de cada barrido
-end
hold off;
sound(aux,fs);
audiowrite("batido 1k If 10.wav",aux,fs);
```



Representacion en el tiempo:



Se ha dejado tambien como prueba al lector un barrido en pasos de 4 en 4 Hz desde 950Hz hasta 1050Hz, para oirlo en la memoria se llama "batido_1k_If_50.wav", la figura que ocurre es :

