UNIVERSITÀ
DEGLI STUDI DI TRIESTE

dipartimento
di ingegneria
e architettura

Psicoacustica

A.Carini – Elettronica per l'audio e l'acustica

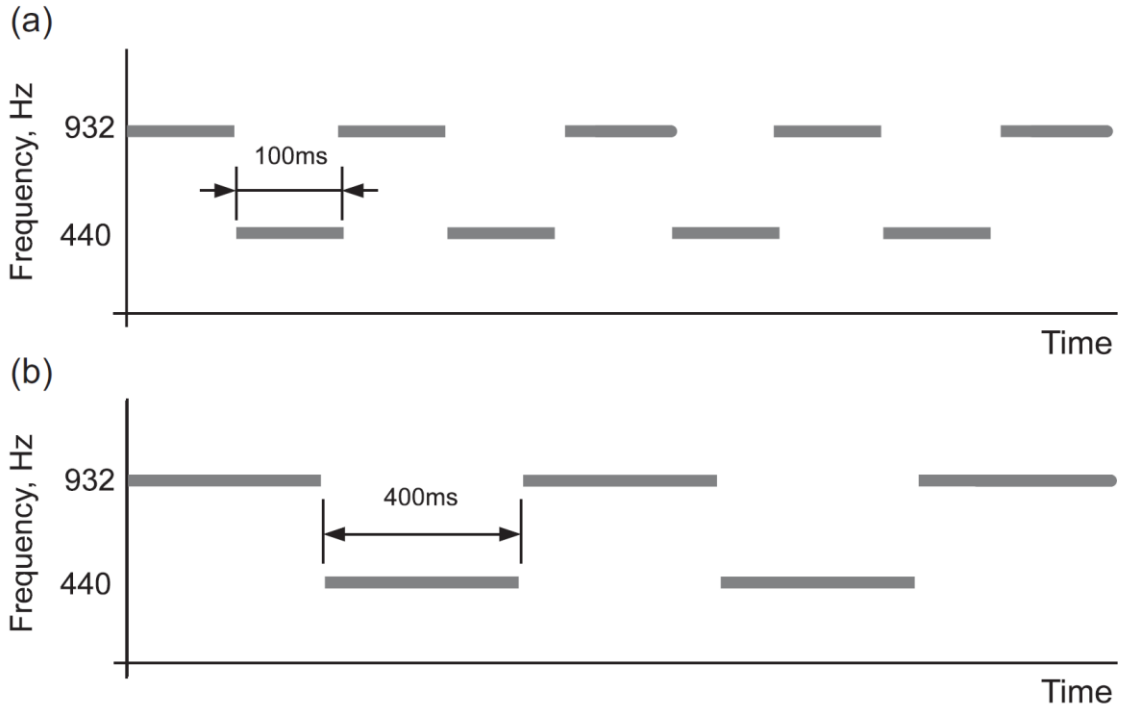
Analisi della scena sonora

- L’analisi della scena sonora descrive il processo uditivo che viene usato per trattare e interpretare combinazioni di suoni complessi.
- Quando sulla stessa scena ci sono più suoni, l’orecchio riesce a sintonizzarsi sul singolo suono che vuole ascoltare estraendolo dalla scena.
- Ci sono diversi meccanismi che intervengono nell’analisi di una scena sonora:
  - La prossimità,
  - La chiusura,
  - Il destino comune,
  - La buona continuazione.

La prossimità

- Suoni in prossimità sono quelli vicini in termini di ampiezza, pitch, durata, timbro.
- Il cervello ha la tendenza a classificare suoni in prossimità come appartenenti alla stessa sorgente.

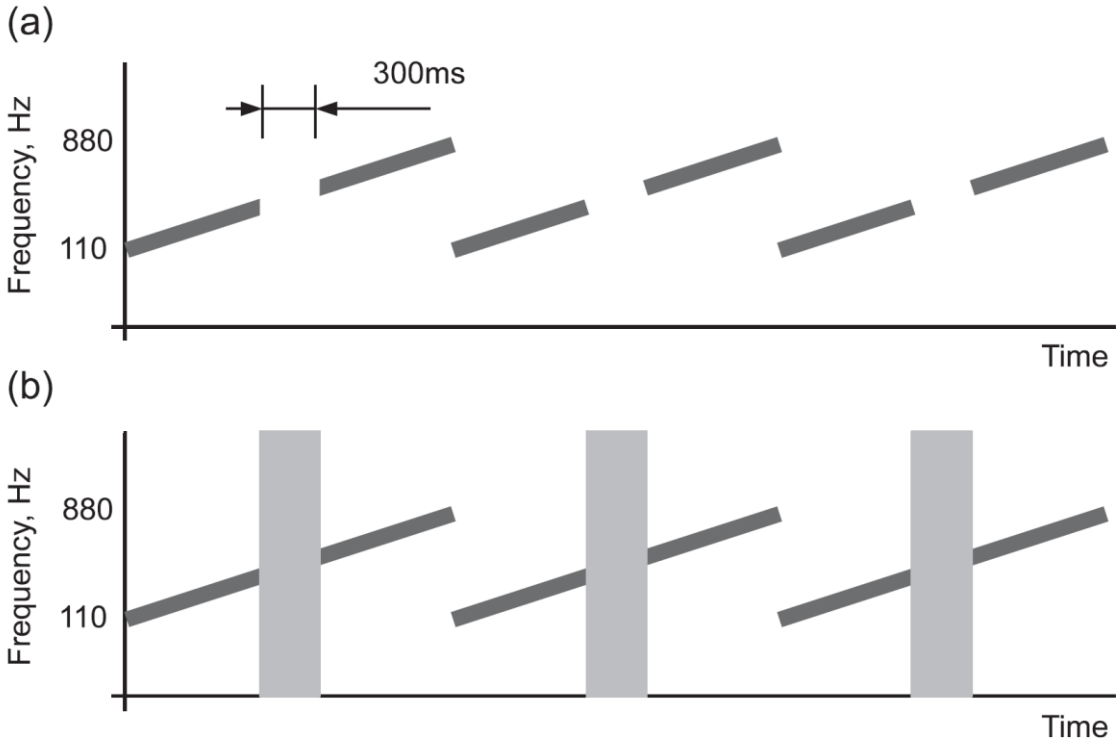
La prossimità - esempio



Chiusura

- Il principio di chiusura consente al cervello di riempire eventuali «buchi» per soddisfare le sue aspettative rispetto alla continuità o all'origine del suono.
- Il fenomeno è anche noto con il nome di induzione uditoria: il cervello non deduce la continuazione del suono mancante ma induce/crea un rimpiazzo per coprire i buchi.

Chiusura - esempio



Destino comune

- Quando gruppi di toni o rumori partono, si arrestano, fluttuano insieme, vengono normalmente interpretati come parte di un suono combinato avente una sorgente comune.

Destino comune - esempio

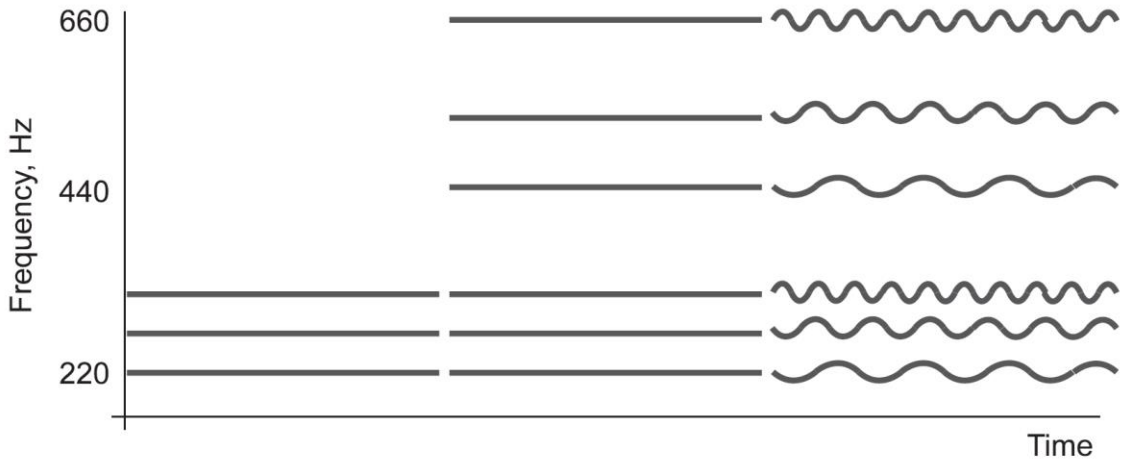
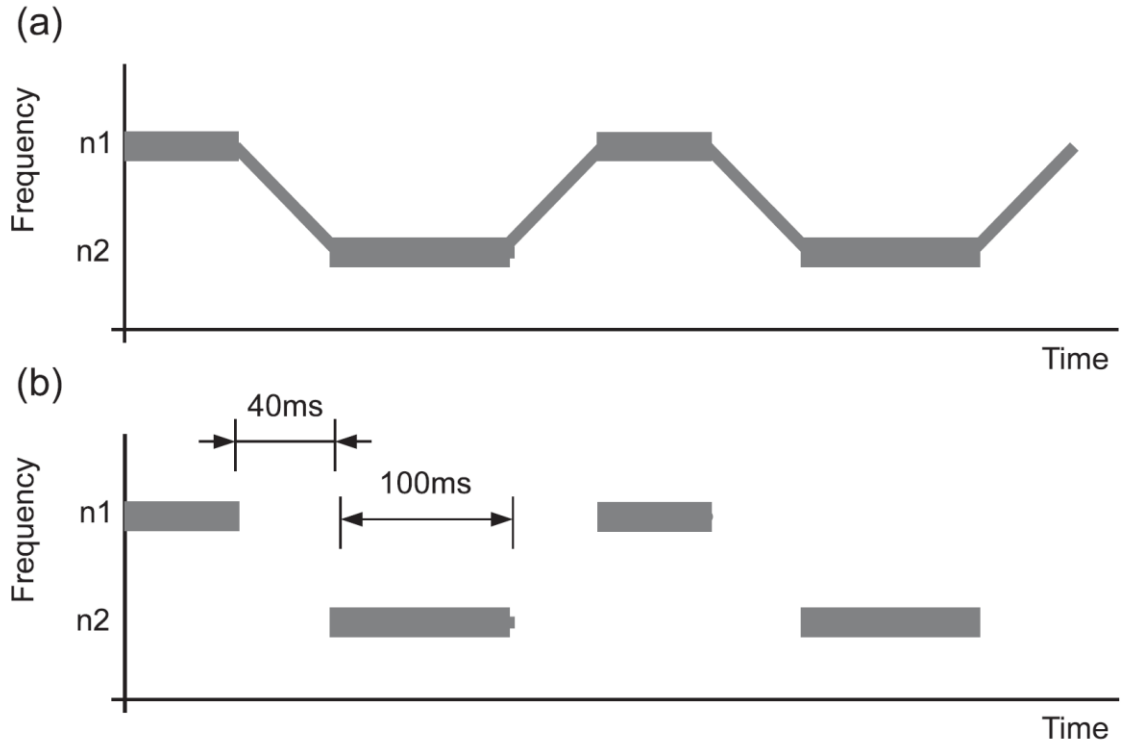


Figure 5.3 Illustration of common fate by comparing three chords. The first generates a pleasing note from three related sinewaves of different frequency. The second adds a first harmonic to each of the three fundamentals. The third modulates each of the three tones plus their first harmonic with unrelated modulating frequencies.

La buona continuazione

- In genere in natura i suoni non iniziano, si fermano o cambiano di frequenza istantaneamente. Ci sarà normalmente un «attacco» dell'ampiezza all'inizio, un decadimento alla fine. La frequenza istantanea slitterà da una frequenza all'altra più o meno lentamente.
- In presenza di suoni complessi, il cervello tende a classificare suoni connessi da queste variazioni graduali come provenienti dalla stessa sorgente. Al contrario, suoni che non sono legati vengono ritenuti provenienti da sorgenti diverse.
- Il fenomeno è detto della «buona continuazione» ma più propriamente dovrebbe essere chiamato della connettività dei suoni.

La buona continuazione - esempio



Modellazione psicoacustica

- La registrazione di un'onda sonora «fisica» contiene elementi che sono molto rilevanti per un ascoltatore e elementi che non lo sono affatto.
- Esempio: il mascheramento post-stimolo

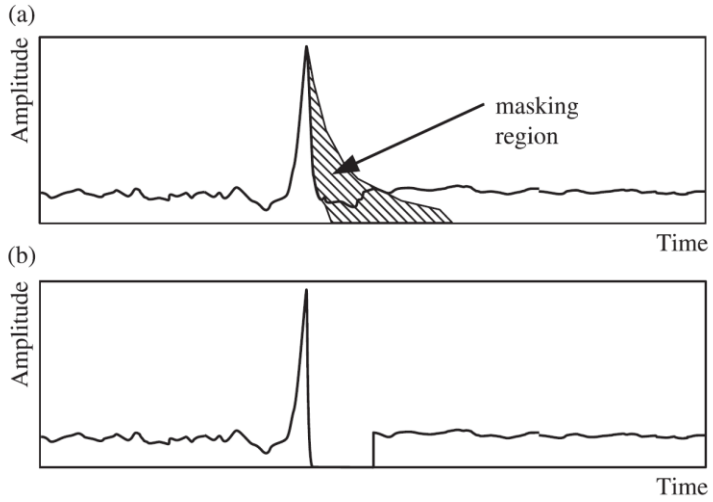


Figure 5.5 Illustration of post-stimulatory masking in the time domain detected (a) and inaudible information stripped out (b).

Applicazioni della psicoacustica

- Compressione di segnali audio ad alta fedeltà.
- Compressione della voce.
- Steganografia audio (data hiding o watermarking).
- Cancellazione attiva del rumore.
- Miglioramento dell'intellegibilità della voce.
- Sistemi di riconoscimento della voce.
- Machine hearing.
- Sintesi del suono e della voce.

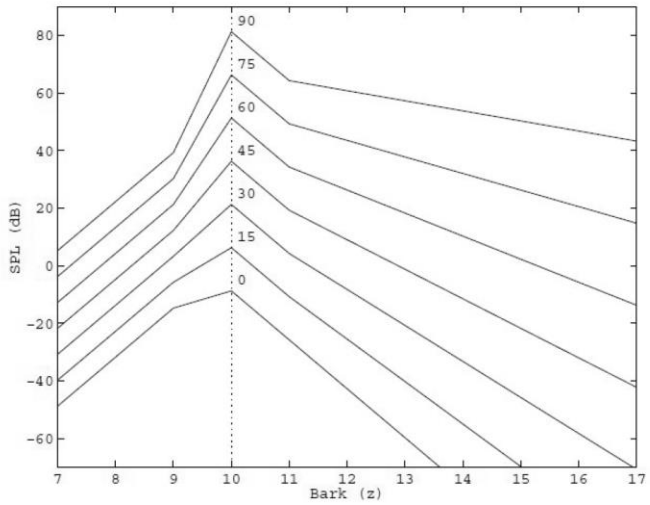
- La maggior parte di questi sistemi impiega:
  - Un modello che predice il mascheramento,
  - Una pre-enfasi basata su equal-loudness (curve isofoniche),
  - Metodi basati sulla discriminazione delle frequenze (scala MEL o Bark).

Applicazioni della psicoacustica

- Il mascheramento di frequenze simultanee è molto utilizzato in quanto facilmente modellabile.
- I modelli di mascheramento considerano l'effetto che hanno toni o rumori passabanda e, per quanto opinabile, applicano i risultati a suoni complessi.
- Si assume che i suoni complessi possano essere scomposti in un insieme di toni o rumori passabanda, ognuno dei quali produce un effetto di mascheramento, con il mascheramento complessivo dato dalla somma dei contributi separati.
- Il mascheramento reale dipende dall'esatta ampiezza del suono che giunge all'orecchio. Questa informazione in genere non è disponibile. Si fa riferimento alle ampiezze relative dei suoni e si assume un livello approssimato d'ascolto.

Modello a bande critiche

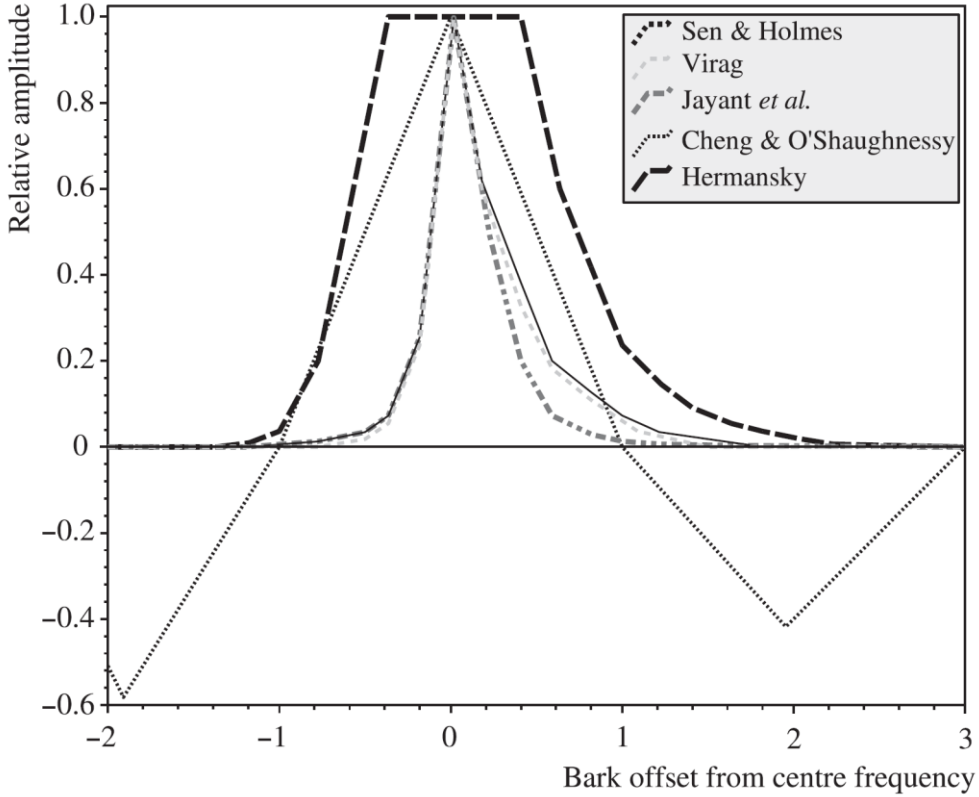
- Il suono in una banda critica influenza pure le bande critiche vicine. Dato un forte suono a una certa frequenza la sua influenza tende a diminuire con la distanza in frequenza. Ciascun suono influenza gli altri suoni in relazione alla loro ampiezza e frequenza.



Prototype spreading functions at $z=10$ as a function of masker level

(mascheramento con toni nell'MPEG)

Critical band spreading functions



Frequenza centrale di
1 kHz e 70 dB SPL

“It is a rather crude approximation
of what is known about the shape of auditory filter»
Hermansky

Critical band spreading functions

Listing 5.1 spread_hz.m

```
function band=spread_hz(hz_array,hz_c)
  %hz_array is an array of Hz frequencies
  %hz_c is the current centre frequency in Hz
barkc=f2bark(hz_c);
band=zeros(size(hz_array));

for hi=1:length(hz_array)
    barki=f2bark(hz_array(hi));
    barkd=barki-barkc;
    if barkd >= -2.5 & barkd <=-0.5
        band(hi)=10^(1*(barkd+0.5));
    elseif barkd > -0.5 & barkd <0.5
        band(hi)=1;
    elseif barkd >= 0.5 & barkd <=1.3
        band(hi)=10^-(2.5*(barkd-0.5));
    end
end
```

barkd=barkc-barki;

Critical band spreading functions

```
F=[0:512]*4000/512;
spread=spread_hz(F, 1200);
plot(F,spread,'o-')
```

Mel-based spreading functions

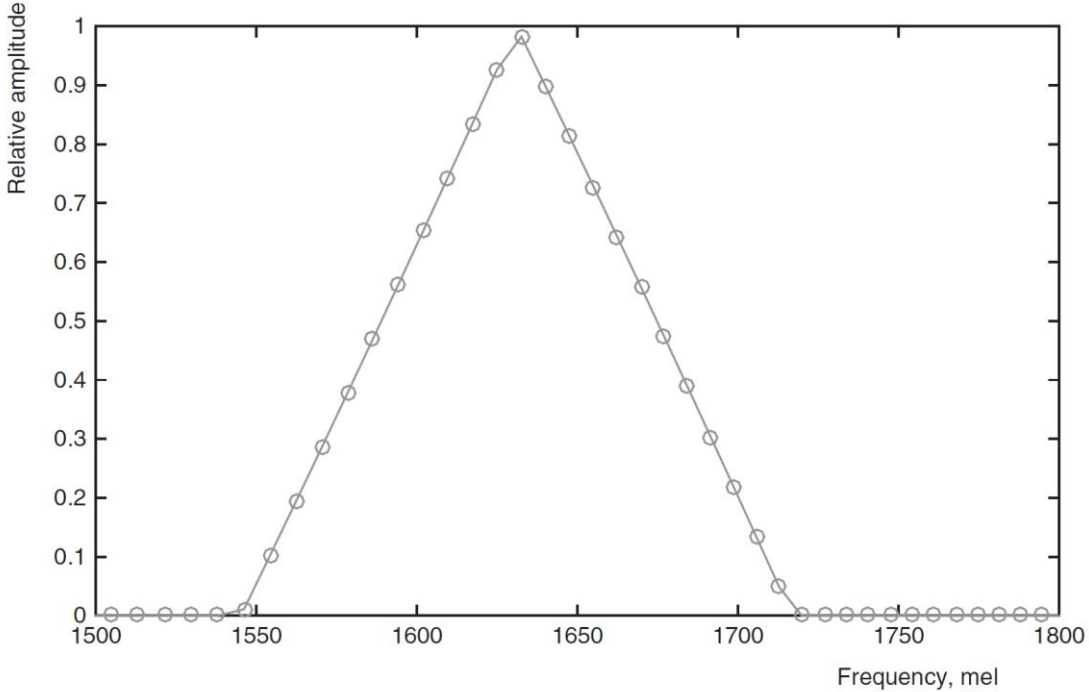


Figure 5.7 The triangular mel-based spreading function.

Mel-based spreading functions

Listing 5.2 spread_mel.m

```
function band=spread_mel(hz_points,hz_c,hz_size,hz_max)
%hz_array is an array spaced in Hz
%hz_c is the current index
band=zeros(1, hz_size);
hz1=hz_points(max(1,hz_c-1));                    %start
hz2=hz_points(hz_c);                             %middle
hz3=hz_points(min(length(hz_points),hz_c+1)); %end
%-----
for hi=1:hz_size
    hz=hi*hz_max/hz_size;
    if hz > hz3
        band(hi)=0;
    elseif hz>=hz2
        band(hi)=(hz3-hz)/(hz3-hz2);
    elseif hz>=hz1
        band(hi)=(hz-hz1)/(hz2-hz1);
    else
        band(hi)=0;
    end
end
```

Mel-based spreading functions

```
mmax=f2mel(4000);
melarray=[0:mmax/255:mmax]; %256 elements
hzarray=mel2f(melarray);
[idx idx]=min(abs(hzarray-1200));
spread=spread_mel(hzarray,idx,100,4000);
plot(f2mel([1:100]*4000/100),spread,'o-')
```

```
mmax=f2mel(4000);
melarray=[0:mmax/24:mmax];
hzarray=mel2f(melarray);
[idx idx]=min(abs(hzarray-1200));
spread=spread_mel(hzarray,idx,256,4000);
plot(f2mel([1:256]*4000/256),spread,'o-')
```

Critical band filter-banks

- Le spreading functions viste sono usate per modellare l'effetto di una singola banda critica. Normalmente sono usate per consentire la rappresentazione di suoni in modo percettivamente rilevante, mediante un vettore corrispondente all'eccitazione delle singole bande critiche.
- L'eccitazione viene calcolata a partire dallo spettro del segnale (dalla DFT) calcolando il contributo a ciascuna banda critica.
- Si lavora o nel dominio Bark (ad es. con funzione di Hermansky) o nel dominio Mel con funzioni triangolari.

Critical band filter-banks

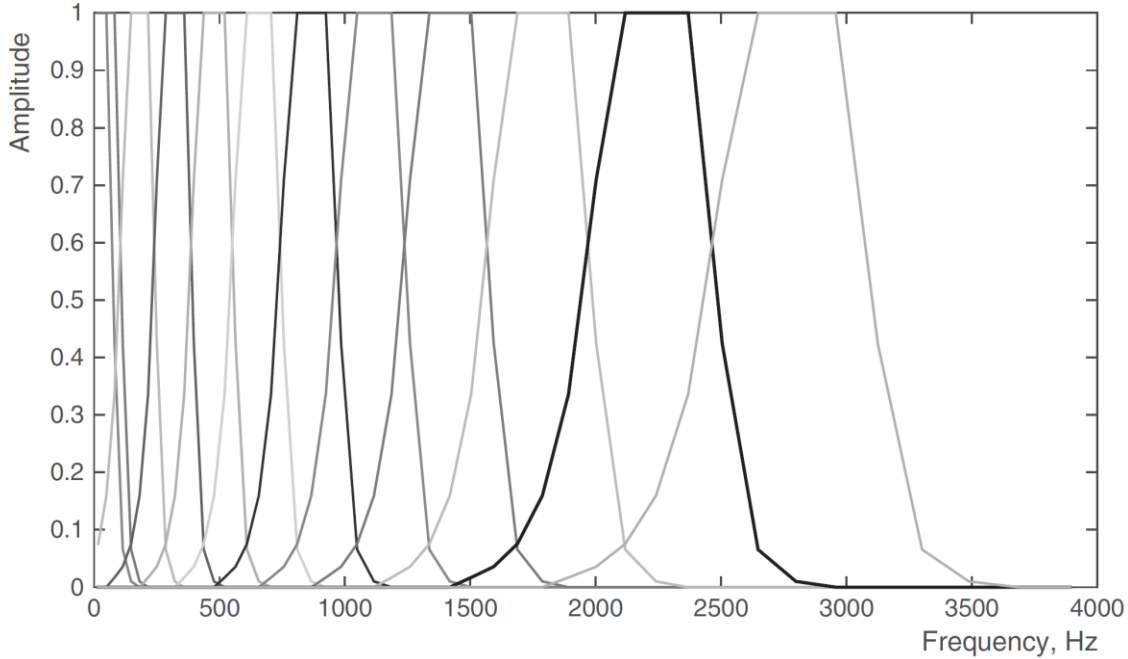


Figure 5.8 A plot of 12 critical-band spreading functions, based on Equation (5.1), over a 48-element array.

Critical band filter-banks

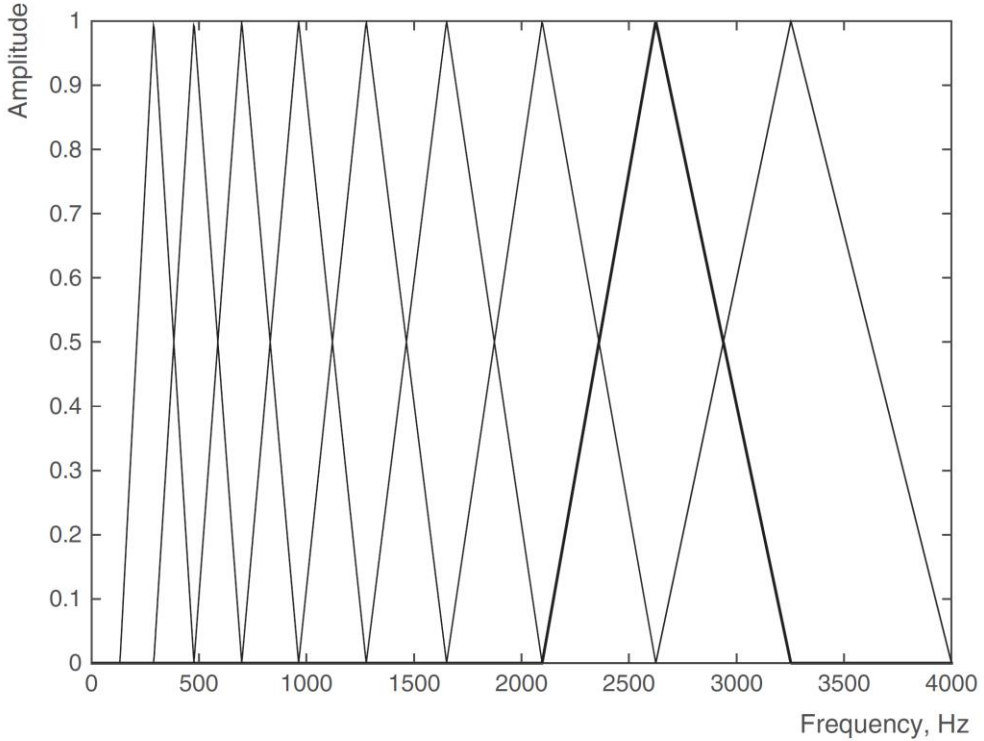


Figure 5.9 A plot of nine mel frequency critical-band spreading functions.

Critical band filter-banks

- Consideriamo l'implementazione di un modello psicoacustico nell'ipotesi che si voglia esaminare un segnale audio per determinare la soglia di mascheramento dovuta ai suoni presenti nel segnale stesso.
- Tre passi:
  - Analisi spettrale
  - Warping a bande critiche e convoluzione con funzione di spreading.
  - Conversione / pre-enfasi equal-loudness (es. pesatura con legge A).

Spectral analysis

```
S=fft(seg.*hamming(1,256));
S=S(1:128);
```

Equal-loudness pre-enphasis

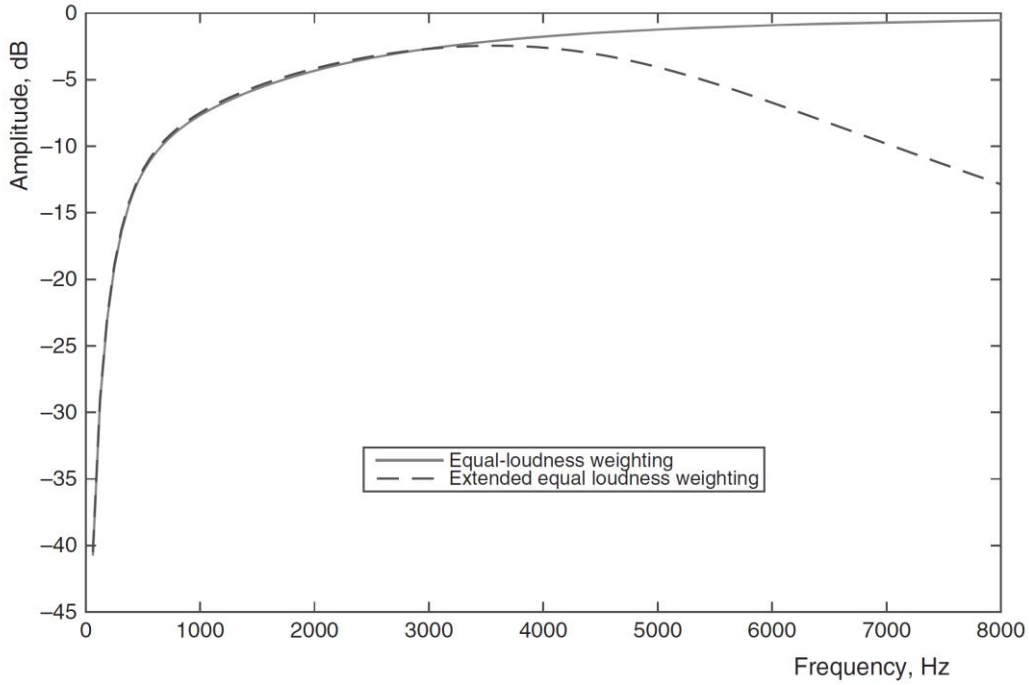


Figure 5.10 The shape of the equal-loudness emphasis curves of Equations (5.5) and (5.4).

Equal-loudness pre-enphasis

$$E(\omega) = \frac{\omega^4(\omega^2 + 56.8 \times 10^6)}{(\omega^2 + 6.3 \times 10^6)^2(\omega^2 + 0.38 \times 10^9)}.$$ <4 kHz

$$E(\omega) = \frac{\omega^4(\omega^2 + 56.8 \times 10^6)10^{27}}{(\omega^2 + 6.3 \times 10^6)^2(\omega^2 + 0.38 \times 10^9)(\omega^6 + 9.58 \times 10^{26})}.$$ >4 kHz

Listing 5.3 equal_loudness_preemph.m

```
function eql=equal_loudness_preemph(hz,varargin)
%hz is frequency in Hz
%if second argument given, use extended form
w=hz*2*pi; %convert to abs angular freq.
w2=w.^2;   %squared version
if ~isempty(varargin)
     eql=1e27*((568e5+w2).*(w2.^2))./(((63e5+w2).^2)
         .*(38e7+w2).*((w.^6)+958e24));
else
    eql=((568e5+w2).*(w2.^2))./(((63e5+w2).^2).*(38e7+w
        .^2));
end
```

Hermansky-style model

Listing 5.4 percept_model.m

```
function [x, xf]=percept_model(seg, N, Fs)
% Map audio frame seg to N-point 0:(Fs/2)Hz percept
    model. N typically 40 at Fs = 8 kHz
b_low=0;                %Bark span lower limit
b_top=f2bark(Fs/2);     %Bark span upper limit
bdiv=(b_top-b_low)/N; %Bark resolution
%Define an array of centre frequencies
xb=b_low+bdiv/2:bdiv:b_top-bdiv/2;
xf=bark2f(xb);  %Convert to Hz
S=abs(fft(seg));
S=S(1:length(S)/2);
F=[1:length(S)]*(Fs/2)/length(S);
x=zeros(1,N);
for xi=1:N
  bark=xb(xi);
  hz=bark2f(bark);
  %compute spreading function
  spr=spread_hz(F,hz);
  %compute summed influence
x_sum=sum(spr.*S')*equal_loudness_preemph(hz);
  %intensity loudness power law
  x(xi)=(x_sum)^0.33;
end
```

Alla normalizzazione con legge A si fa in genere seguire un enfasi secondo legge di potenza

$$\mathcal{P}(\omega) = \{E(\omega)\Theta(\omega)\}^{0.33}.$$

MFCC model

Listing 5.6 mfcc_model.m

```
function cc=mfcc_model(seg, N, M, Fs)
% Do FFT of audio frame seg, map to M MFCCs
% from 0 Hz to Fs/2 Hz, using N filterbanks
% typical values N=26,M=12,Fs=8000,seg~20ms
m_low=0;              %mel span lower limit
m_top=f2mel(Fs/2);  %mel span upper limit
mdiv=(m_top-m_low)/(N-1); %mel resolution
%Define an array of centre frequencies
xm=m_low:mdiv:m_top;
%Convert this to Hz frequencies
xf=mel2f(xm);
%Quantise to the FFT resolution
xq = floor((length(seg)/2 + 1)*xf/(Fs/2));
%Take the FFT of the speech...
S=fft(seg);
S=abs(2*(S.*S)/length(S));
S=S(1:length(S)/2);
F=[1:length(S)]*(Fs/2)/length(S);
%Compute the mel filterbanks.m
x1=zeros(1,N);
for xi=1:N
    band=spread_mel(xf,xi,length(S),Fs/2);
    x1(xi)=sum(band.*S');
end
x=log(x1);
%Convert to MFCC using loop (could use matrix)
cc=zeros(1,M);
for xc=1:M
  cc(xc)=sqrt(2/N)*sum(x.*cos(pi*xc*([1:N]-0.5)/N));
end
```

Mascheramento di rumori

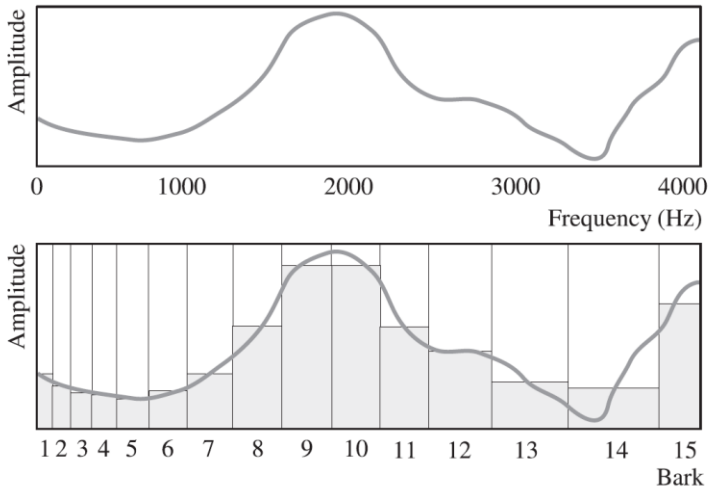


Figure 5.13 In-band masking level derived and plotted (bottom) from an example spectrum (top) for 15 critical bands of constant Bark width.

Vedere:

- Ian Vince McLoughlin, “Speech and Audio Processing”- Cambridge Univesity Press (2016)
  - Cap. 5