MANCHESTER 1824
The University of Manchester

Mobile Systems

Lab Support Lecture 3

Workshop 3:

(Psycho-acoustics)

COMP28512
Steve Furber & Barry Cheetham

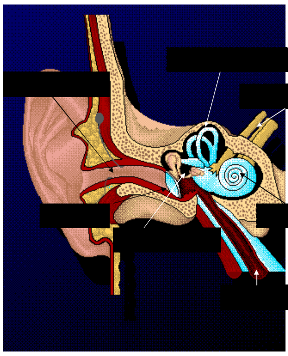
MANCHESTER 1824
The University of Manchester

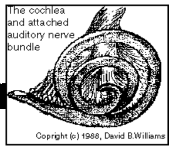
Psycho-acoustics

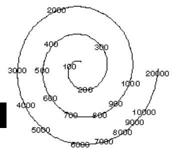
- Principles of human perception of sound
  - MP3 compression algorithm uses model of human hearing to remove data (perceptual coding algorithm)
  - frequency range is about 20 Hz to 20 kHz,
  - most sensitive at 2 to 4 kHz.

MANCHESTER 1824
The University of Manchester

Structure of the Ear





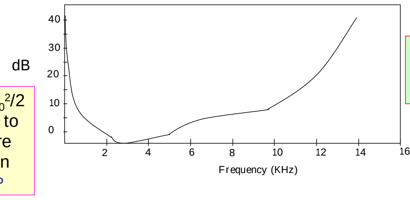


MANCHESTER 1824
The University of Manchester

Expt 1: Threshold of hearing

- Put a person in a quiet room & play a 2 kHz sine-wave.
- Raise level of 2 kHz tone until just barely audible.
- Note amplitude A_0 of sine-wave. Its power = $A_0^2/2$.
- In theory, sound pressure variation will be ≈ 20μPascals.
- Repeat expt with new frequency to find new amplitude A.
- Plot $10\log_{10}((A^2/2) / (A_0^2/2))$ for many frequencies.

rel to $A_0^2/2$
≈ equiv to
pressure
variation
of 20 μP



Masking contour 'in quiet'

MANCHESTER 1824
The University of Manchester

Masking contour "in quiet"

- Power of sine-wave $(A^2/2) \propto$ pressure variation (loudness of sound)
- Sine-waves of power below threshold will not be heard.
- Can save bits by not transmitting them.
- Ear most sensitive in frequency range 2-4 kHz
- Two tones of equal power & different frequencies will not sound equally loud
- Sensitivity of ear decreases at low & high frequencies.
- Performing Expt 1 would be possible, but tedious.
- Fortunately a formula exists.
- Taking reference as 20 μPascals we get dB_SPL
- SPL stands for "Sound pressure level"

MANCHESTER 1824
The University of Manchester

Formula for threshold in quiet

$$Tq\_dB(f) = 3.64 \times f^{-0.8} - 6.5 \times \exp(-0.6 \times (f - 3.3)^2) + f^4/1000$$

Gives hearing threshold in dB_SPL at frequency f **kHz**.

Assumes volume control has been callibrated as follows:

Callibration of volume control

- Choose A_0 to be a small value, say 16.
- Generate a 2 kHz sine-wave of amplitude A_0
- Then set volume control so you can **just** hear it.
- Amplitude A_0 corresponds to 20 μP pressure variation.
- We have now set the volume control so that sound level is 0 dB_SPL when sine-wave amplitude is A_0.
- If we change the volume control, or A_0, must re-callibrate

- With 16-bit A-D converter, largest amplitude $\approx 2^{15}$.
- This is 2^{11} times larger than A_0.

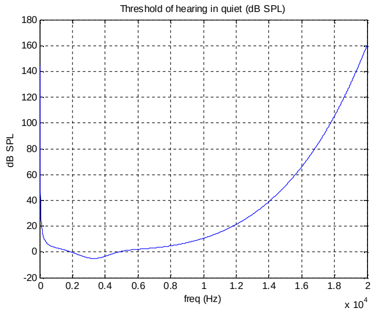
$\therefore \approx$ 66 dB louder than threshold; i.e $\approx$ 66 dB_SPL.

$\therefore$ To generate sine-wave at 60 dB_SPL, set A $\approx 2^{14}$ = 16384 (Actually it is 60.2 dB_SPL)

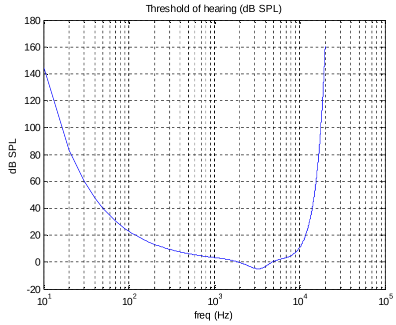
Check

- Have set volume control so that a 2 kHz sine-wave $A_0 \sin(2\pi ft)$ with f =2000 can **just** be heard:
- Power of this sine-wave is $A_0^2/2$.
- We chose A_0 = 16.
- What sine-wave amplitude A must we use to generate a sound level of 60 dB_SPL?
- $10 \times \log10((A^2/2) / (A_0^2/2)) = 60$
- $(A/16)^2 = 10^6$

Linear frequency scale



Log-frequency scale



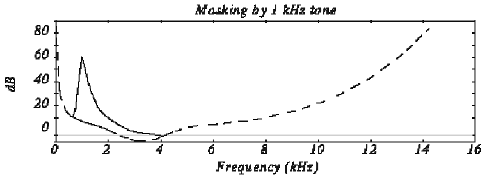
dB SPL

- Implies a reference of 20 μPascals RMS in air.
- Quietest sound a person can detect at about 1 kHz*.
- Sound of a mosquito flying about 3 meters away.
- Threshold varies with frequency.
- Is 1 kHz frequency of a typical mosquito, I wonder ?
- Sound level of a sine-wave in dB(SPL) is: 10 log10 (Power of sine-wave / Po)
- Po = power of 1 kHz* sine-wave at threshold of hearing.

* Graph seems to make this 2 kHz

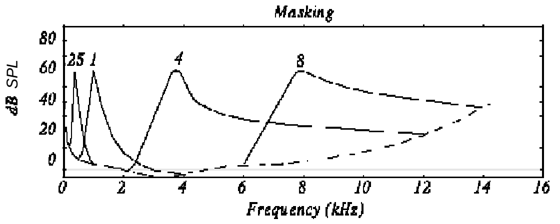
Expt 2: Frequency Masking

- Play 1 kHz masking tone at amplitude A_1 = 16000 corresponding to 60 dB_SPL
- Play test tone at 1.1 kHz with amplitude A
- Start with A=0 & increase A until just distinguishable.
- Note value of threshold A.
- Express power $A^2/2$ of test tone in dB_SPL
- Repeat expt over a range of test tone frequencies
- Plot threshold of hearing for test tone:



Effect of varying Masker

- Could repeat previous experiment for various frequencies of masking tones : 250Hz, 1 kHz, 4 kHz, etc.
- This would be exceedingly tedious,
- But it has been done, & results look like this:



Formula for freq-masking

- Assume a masker at *Fmask* Hz of power *Pmask* dB_SPL
- Generate a test tone at *f* Hz of power *Ptest* dB_SPL.
- Do we hear the test tone?
- Let fd = *f* – *Fmask* & convert *fd* to Barks (called *zd*):

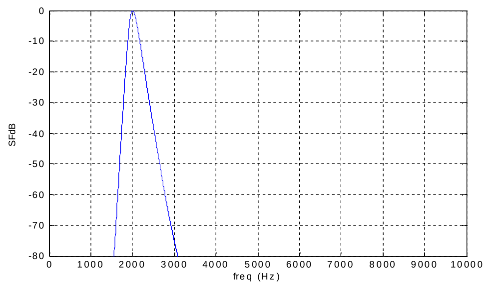
  $zd = 13 \times atan(0.00076 \times fd) + 3.5 \times atan((fd/7500)^2)$
- Then calculate spreading function:

  $SF(f) = Pmask + 15.81 + 7.5 \times (zd+0.474) - 17.5 \times sqrt(1+(zd+0.474)^2)$
- Units are dB_SPL
- At freq *f*, a sinusoid of power < SF(*f*) will not be heard.

Spreading functn with *Fmask* = 2 kHz

Linear frequency scale

Add power of masker in dB_SPL

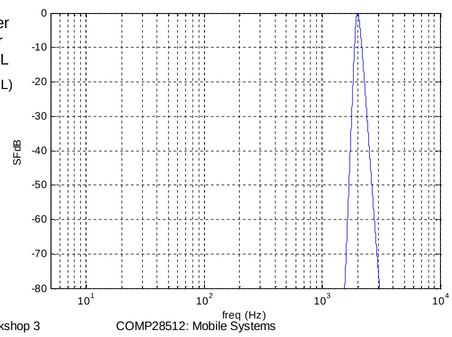


Spreading functn with *Fmask* = 2 kHz

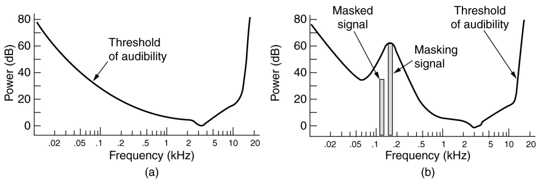
Log frequency scale

Add power of masker in dB_SPL

(60 dB_SPL)



Frequency-Masking

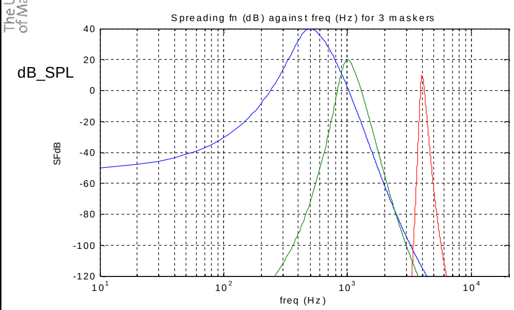


(a) Threshold of audibility 'in quiet' as function of frequency.
(b) Threshold of audibility with 'freq-masking' effect.

3 maskers with different amplitudes

```
F1=500; F2=1000; F3 = 4000;  % Freqs of maskers in Hz
A1dB_SPL = 40;  % Power of F1 masker in dB rel to Pref
A2dB_SPL = 20; A3dB_SPL = 10;
for n = 1:2000
   f(n)=n*10;   % Freq to plot in Hz
   f1d(n) = f(n) - F1;  f2d(n) = f(n) - F2;  f3d(n) = f(n) - F3;
   z1d(n) = 13*atan(0.00076*f1d(n)) + 3.5 * atan((f1d(n)/7500)^2); % Barks
   SFdB1(n) = 15.81 +7.5*(z1d(n)+0.474) - 17.5 * sqrt(1+(z1d(n)+0.474)^2 );
   % Repeat for f2d(n) and f3d(n)
   SFdB1(n) = SFdB1(n) + A1dB_SPL;  % Add DBs
   SFdB2(n) = SFdB2(n) + A2dB_SPL;
   SFdB3(n) = SFdB3(n) + A3dB_SPL;
end;
semilogx(f,SFdB1,f,SFdB2, f, SFdB3); grid on;
xlabel('freq (Hz)'); ylabel('SFdB');
```

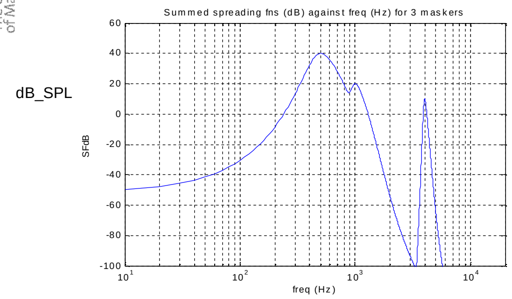
3 indep spreading functns



Combined effect of 3 maskers

```
%Add together their powers (in Watts, not dB)
for n=1:2000
   SF1(n) = 10^(SFdB1(n)/10);
   SF2(n) = 10^(SFdB2(n)/10);
   SF3(n) = 10^(SFdB3(n)/10);
   SF(n) = SF1(n) + SF2(n) + SF3(n);
   SFdB(n) = 10*log10(SF(n));
end;
```

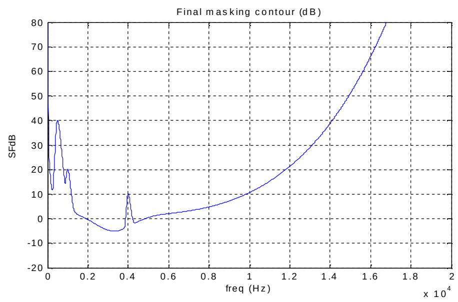

Combined effect of 3 maskers



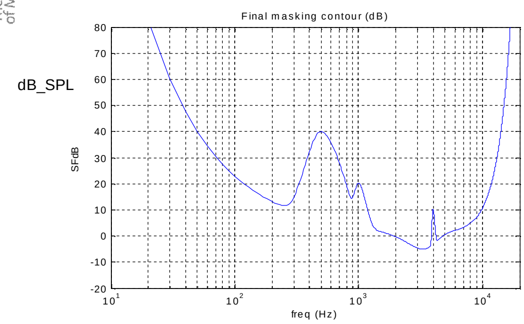
Combine with threshold of quiet

```
for n = 1:2000
   SF(n) = 10^(SFdB(n)/10); % from previous graph
   Tq(n) = 10^(Tq_dB(n)/10);  % Threshold of quiet
   SF(n) = SF(n) + Tq(n);
   SFdB(n) = 10*log10(SF(n));
 end;
```


Final masking contour (lin scale)



Final masking contour (log scale)



End of frequency masking

- Have achieved goal of deriving a masking contour when there are several maskers of different amplitudes.
- Any signal component below this masking contour will not be heard.
- The masking contour will be continually changing.
- There is another effect to consider now:

3 Mar'15 Workshop 3 COMP28512: Mobile Systems 25

Temporal masking

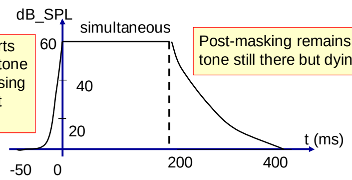
- A loud sound will render inaudible a quieter sound occurring shortly before or shortly after it.
- Duration of the effect depends on amplitude difference.
- Effect of frequency masking will continue for a while after a masking tone has finished.
- And, to a lesser extent, even before it starts.
- So the frequency masking contour for a given frame should be calculated taking account previous frames
- And perhaps one frame ahead as well

3 Mar'15 Workshop 3 COMP28512: Mobile Systems 26

Illustration of temporal masking

- Assume a 60 dB_SPL masking tone at 2 kHz.
- Assume it starts at 0 ms & stops 200 ms later.
- Spreading functn is as shown earlier while masking tone is present. Does not suddenly disappear after 200 ms.
- Masking continues as though tone still there but dying away.
- Pre-masking effect taken into account in similar way,

Pre-masking starts as tho' masking tone is there & increasing like this, before it actually starts

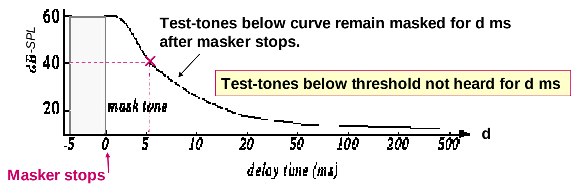


Post-masking remains as tho' masking tone still there but dying away like this.

3 Mar'15 Workshop 3 COMP28512: Mobile Systems 27

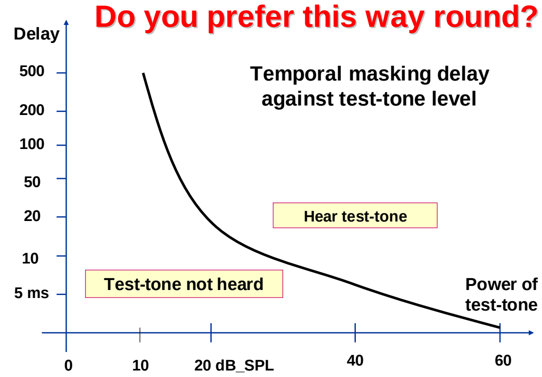
Expt 3: Temporal Masking

- Play 1 kHz masking tone of amplitude 16000 (60 dB_SPL)
- Add1.1 kHz test tone of power 40 dB_SPL
- Test tone should not be heard (it's masked)
- Stop masking tone, then stop test-tone *d ms* later.
- If *d >100 ms*, you should hear test-tone when masker stops.
- Reduce d until test-tone cannot be heard.
- Repeat with different levels of test-tone & plot against d.



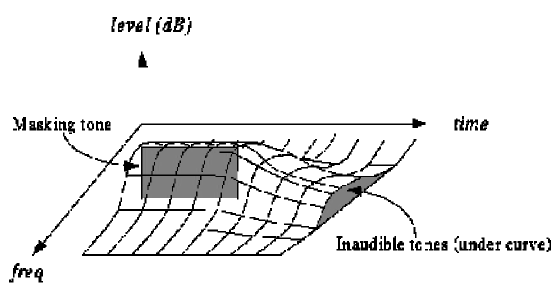
3 Mar'15 Workshop 3 COMP28512: Mobile Systems 28

Do you prefer this way round?



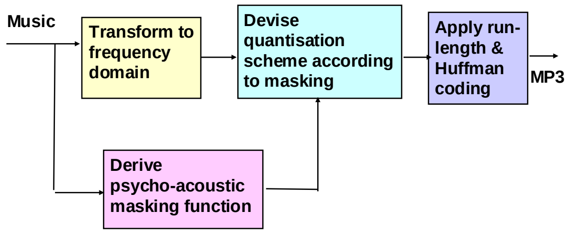
3 Mar'15 Workshop 3 COMP28512: Mobile Systems 29

Combined effect

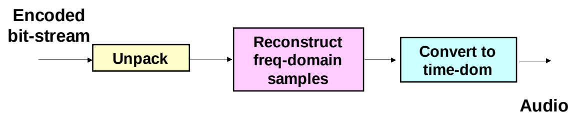


3 Mar'15 Workshop 3 COMP28512: Mobile Systems 30

MP3 encoder



MP3 decoder



Time to Frequency Transform

Transforming time/level input signals to frequency/power

FFT – fast & easy, but needs amplitude & phase.

DCT – no complex numbers

Wavelets use non-sine/cosine functions:
for better performance on data with sharp discontinuities.

MP3 algorithm overview

1. Take suitably sized (overlapping) blocks of audio.
2. Spectrally analyse each block.
3. Determine masking contour using psychoacoustic model.
4. If amplitude of a spectral sample is below masking contour, don't encode it – it will not be heard..
5. Otherwise, decide how many bits are needed to represent it according to how many dBs it is above masking contour.
   - Need 1 bit for up to 6 dB, 2 bits for up to 12 dB, & so on
   - Makes noise introduced by quantization fall below masking contour.
6. Produces samples with varying numbr of bits & lots of zeros.
7. Employ 'variable length coding' & 'run-length coding'.
8. Reconstruct DCT samples & undo overlapping at receiver.

Spectrum with 32 frequency bands

- Assume levels of first 16 of the 32 freq bands are:

Band	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Level dB_{SPL}	0	8	12	10	6	2	10	60	35	20	15	2	3	5	3	1

- Level of sound in 8th band is 60 dBspl
- Assume spreading formula gives masking of up to 12 dBspl in 7th band & up to 15 dBspl in 9th band.
- Level in 7th band is 10 dBspl (< 12) so call it 0.
- Level in 9th band is 35 dBspl (> 15 dBspl).
  - It is 20 dB above masking threshold, so have to send it.
  - Only power above the masking level needs to be sent.
  - We get 6 dB per bit, so only need 4 bits for band 9.

Example

- Assume we get the following quantised DCT values:

10100 0 0 0 0 0 01101 111 0 0 0 0 0101 0 0 111

S1 S2 S3 S4 S3

- Represent each non-zero word by a symbol: S1, S2, etc...
- Represent each symbol by a special code (see later)
- Then introduce run-length coding by allowing up to (say) 15 zeros to follow each symbol:

(S1 0101) (S2 0000) (S3 0100) (S4 0010) (S3 0000)

- Finally use Huffman coding to represent the symbols.

(Hey, what happens if we get a run of more than 15 zeros?)

Huffman coding

- Make a look-up table for symbols:

Symbol	Value	Frequency	Huffman code
S0	0	17	longer code
S1	10100	200	
S2	01101	375	
S3	111	679	shorter code
S4	xxx	142	
S5	xxx	13	longer code
S6	xxx	255	
etc.	..	...	..

- Assign shorter code-words to frequently occurring symbols (like S3).
- Longer codes to rare symbols (like S5).
- Self terminating codes used. More on this later.

MPEG layers

Layer 1
- DCT-type filter with one frame
- equal frequency spread per band
- psychoacoustic model only uses frequency masking

Layer 2
- use three frames in filter
  - before, current, next, a total of 1152 samples
- models a bit of temporal masking

Layer 3 (mp3)
- better critical band filter is used
  - non-equal frequencies
- psychoacoustic model includes temporal masking effects
- takes into account stereo redundancy
- uses Huffman coder

MP3 Compression Rates

Format	Bit rate	Compression ratio	File size per min	Notes
WAV	uncompressed	1:1	10 MB	For CD quality, wav file sampled at 44.1kHz / 16 bits
MP3	160 kbps	9:1	1.5 MB	Low compression, superior sound quality
MP3	128 kbps	11:1	1 MB	Standard MP3 bit-rate
MP3	96 kbps	15:1	700K	Lower quality MP3
MP3	64 kbps	22:1	400K	≈ same quality as FM radio

Summary

- mp3 compresses audio signals
  - using psycho-acoustic techniques
  - closely tuned to human hearing
- Applies DCT to overlapping frames
- Remove inaudible frequency & temporal detail
  - better than 10:1 compression at CD quality
- Masking contours for threshold of hearing
- Equations & graphs studied.
- Need run-length coding & Huffman coding (later).