

Proc. of the 7th Int. Conference on Digital Audio Effects (DAFX-04), Naples, Italy, October 5-8, 2004

IMPLEMENTING LOUDNESS MODELS IN MATLAB

J. Timoney, T. Lysaght

Dept. of Computer Science
NUI Maynooth, Maynooth, Co.
Kildare, Ireland
jtimoney, tlysaght@cs.may.ie

Marc Schoenwiesner

AG Neurobiologie Institut
Leipzig, Germany
marcs@rz.uni-leipzig.de

L. MacManus

Dept. of Elec. Engineering
DIT, Dublin, Ireland
lorcan.macmanus@dit.ie

ABSTRACT

In the field of psychoacoustic analysis the goal is to construct a transformation that will map a time waveform into a domain that best captures the response of a human perceiving sound. A key element of such transformations is the mapping between the sound intensity in decibels and its actual perceived loudness. A number of different loudness models exist to achieve this mapping. This paper examines implementation strategies for some of the more well-known models in the Matlab software environment.

1. INTRODUCTION

The primary tool in the field of audio for the time-frequency analysis of sound is the Spectrogram. It is popular because it is computationally fast and its output is well understood. However, since the 1990's much work has been carried out on the development of better tools for sound analysis that make more efforts to take psychoacoustic properties into consideration. This has been driven by the availability of the technology to fully implement the results of psychoacoustic research that had been published over the previous decades, combined with the desire for significant advances in the coding of speech and audio signals. The MP3 standard is a good example of this. Thus nowadays, many algorithms designed for speech and audio processing will make reference to psychoacoustic transformations. An important limitation of the spectrogram in this regard is the manner in which the signal intensity is displayed, generally in Decibels SPL. While this provides a measure of objective sound intensity, it does not properly capture the subjective impression a sound creates on the listener in terms of its loudness. To achieve this the sensitivity of the ear to the various sound levels of the frequency components contained in the sound must be accounted for. This is the kind of information contained in equal loudness curves for the human ear [1]. These curves show that the ear is less sensitive to low frequency sounds, having a maximum sensitivity in the region of 3-4kHz. Employing these curves to modify the dB SPL intensity display of the sound transforms the intensity to the Phon scale, where different frequency components having the same Phon value will have the same loudness but will have different dB SPL

intensities. One disadvantage with the Phon scale is that it is not directly proportional to perceived loudness, and thus a doubling of loudness value in Phons does not mean a doubling of the sound loudness [2]. To this end, the Sone scale was introduced to provide a linear scale of loudness. The Sone scale can be related to the phon scale by the equation [3]

$$L(i) = \left(\frac{D(i)}{40} \right)^{2.642}, \quad \text{if } D(i) < 40 \quad (1)$$

$$L(i) = 2^{0.1(D(i)-40)}, \quad \text{if } D(i) \geq 40$$

where $L(i)$ is the perceived loudness of the critical band i , and $D(i)$ is the spread critical spectrum in terms of phons in band i . The conversion from a time domain signal to a representation that describes its loudness in terms of Sone is outlined in Figure 1

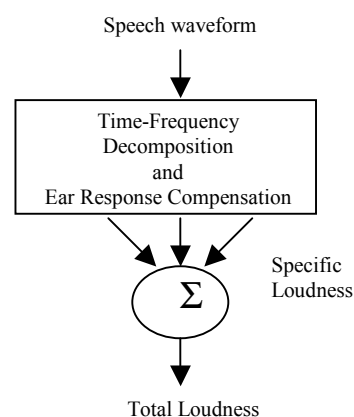


Figure 1: Block Diagram of loudness modeling procedure [4]

There are various approaches to implementing the different stages of the Loudness model in Figure 1. The basic procedure is to first transform the signal into the time-frequency domain. The frequency analysis points specified will have a relation to the critical band resolution of the ear. Time and Frequency masking may be accounted for and compensation carried out for components below

the threshold of audibility. This stage is followed by a conversion from the intensity levels of each time-frequency slice to specific loudness levels for each frequency band. These are then summed to give the overall loudness for each time slice.

In this paper, three implementation strategies are examined:

1. A direct implementation based on a time-frequency decomposition, a mapping from dB SPL to Phon followed by a direct implementation of equation (1);
2. Three implementations of Zwicker's model;
3. The Moore and Glasberg Loudness model;

The sources for some of the implementations discussed are speech quality measurement strategies. Specifically, the time-frequency decompositions and loudness conversion from the EMBSD [3], PSQM [10] and PEAQ [12] measures are investigated.

2. MODEL IMPLEMENTATIONS

2.1 Calibration

In all implementations the first, and possibly most crucial, stage is the calibration of the input signal. In Matlab sounds read in from wav files are normalized to have amplitude levels lying between 1 and -1. However, this will neither reflect the true recording or playback levels of the sound. The amplitude of sound can be scaled to give the sound a desired value of dB SPL. When using dB SPL to set a sound level, a value for the reference level must be chosen. For air, the reference level is usually chosen as 20 micropascals [5]. If the actual dB SPL used when recording the sound is unknown, in the case of speech, if it is at a normal level, it is reasonable to assume a conversational level of between 65 and 70 dB SPL. Thus, to scale the signal vector y to a level of 70dB in Matlab [6]:

```
SPLmeas=70;
Pref = 20e-6;
y_refscaled= (y./Pref);
RMS=sqrt(mean(y_refscaled.^2));
SPLmat=20*log10(RMS); % dB SPL in matlab
c=10^((SPLmeas-SPLmat)/20);
ycal=c*(y_refscaled);
```

If the HUTear toolbox is installed, it is also possible to use the function [7]:

```
ret=Pascalize(y,70);
```

2.2 Direct Implementation of Loudness Representation

The algorithm for the direct implementation is taken from the EMBSD speech quality measure. The signal is separated into frames, each one windowed with a Hanning function and the power spectral density obtained. Each power spectrum is partitioned into critical bands of width one bark, with an upper frequency limit of 3.4 kHz. In [3], Schroeder's spreading function model is applied to include the effects of frequency masking across the critical bands. The loudness level of each critical band in units of *phon* is obtained using a set of equal-loudness contours taken from the literature and dB intensity values that lie in between the published contours are interpolated to get the correct loudness level [3]. These loudness levels are then converted to sone using equation (1).

```
NFFT=1024;NOVERLAP=0;
Bf=1:18;
[Yxx,f] = psd(ycal,NFFT,fs,NFFT,0);
Yxx_scale=(2.*Yxx)./NFFT;
[B_XX,bark]=bk_frq02(Bf,f,Yxx_scale);
C_XX=spread_new(Bf,B_XX);
P_XX=dbtophon(C_XX);
S_XX=phtosn(P_XX);
N_mbsd(1)=sum(S_XX);
```

The Matlab programs are as given in [3]. However, it was found that it was necessary to make adjustments to the program dbtophon.m. First of all, in the program code a file named equal.mat is called as it holds the transcribed equal loudness contours. However, the C program version in the thesis also contains the contour values in an array, which can be copied for use with Matlab. Furthermore, the lines below, which were found to cause errors on occasion:

```
j = 1;
while T(i) >= eqlcon(j,i)
j = j + 1;
if j == 16, fprintf(1, 'ERROR\n'), end
end
if j == 1, P_XX(i) = phons(1), end
```

can be replaced with:

```
[I]=find(T(i)<=equalcon(:,i));
if min(I)==1, P_XX(i)=phons(1), end
```

2.3 Implementations based on Zwicker's model

Possibly the most well-known and popular model of loudness is the one proposed by Zwicker. It has formed part of an international standard [8], and has been adopted for use in a number of ITU standards on speech and audio quality. However, differences exist in the implementations.

2.3.1 Implementation of the DIN 45631/ISO532B Loudness Model

This Matlab program was a direct conversion from the basic program provided in [6]. This implementation uses a filterbank of one-third-octave filters for the spectral decomposition of the signal. However, a drawback is that this yields only a rough approximation to the shape of the auditory filters and the location of their centre frequencies. The equation for the specific loudness N' in Sone/Bark of a the dB SPL sound level L_G in a one-third-octave band is given by [9]:

$$N' = 0.0064 \cdot 10^{0.025 L_{ETQ}} \cdot \left[\left(11 + \frac{1}{4} 10^{0.1(L_G - a_0 - L_{ETQ})} \right) - 1 \right] \quad (2)$$

The transmission of freefield sound to our hearing system through the head and the outer ear is described as attenuation a_0 . The excitation threshold in quiet is L_{ETQ} . Values for these two parameters are given in [8]. To run the implementation given in [6], the sequence of Matlab commands is:

```
[Yxx,f]=PowSpec(Pref.*ycal,fs,df);
[YdB,err]=Convert2dB(Yxx,1);
```

In this implementation, the power spectral density of the signal Y_{xx} is found without the factor P_{ref} taken into account. Furthermore, this quantity is only included after it has been converted to dB, i.e. the following line is required in `Convert2dB.m`:

```
YdB=10*log10((cal^2)*Yxx/(Pref^2));
```

The other input arguments to `PowSpec` are f_s and dF , which are the sampling frequency and the frequency resolution respectively. The third-octave-band filters are generated using the code below and the filter design is performed using a program obtained from the Mathworks called `Oct2dsgn.m`. The output of `PowSpec` is given by f . Scaling of the filter responses is carried out to ensure no energy gain is introduced into the signal. The filtered bands of the signal power spectrum is given as Lt :

```
[H, err]=GenerateFilters_16000(f);
H=0.94833723551160*H;
for ink=1:24
Lt(ink)=10*log10(sum((10.^(YdB/10))
.*(abs(H(ink,:).^2))));
end
```

The final function call returns the total loudness N and specific loudness vector N_s . The input MS defines the sound field and by default is set to a free field, i.e., $MS = 'f'$.

```
[N, Ns, err]=DIN45631_16000(Lt, MS);
```

2.3.2 Zwicker's Loudness model as used in PSQM

The PSQM algorithm was adopted by the ITU for speech quality assessment [10], but it has recently been replaced by the PESQ algorithm [11]. The equation used for the implementation of Zwicker's model is different to (2). The specific loudness $LX(f)_n$ is given by:

$$LX(f)_n = S_l \left(\frac{P_0(f)}{0.5} \right)^\gamma \cdot \left[\left(0.5 + 0.5 \cdot \frac{PPX(f)_n}{P_0(f)} \right)^\gamma - 1 \right] \quad (3)$$

where S_l is a scaling factor, $P_0(f)$ is the absolute hearing threshold at frequency f , $PPX(f)_n$ is the Pitch Power Density at frequency f for frame n and γ is a constant. The Pitch power densities are the power spectrum of a single frame warped to the bark scale with a resolution of 0.321 Bark, with scaling relative to the bandwidth in Hertz. According to the PSQM document [10], $S_l = 1/240.5$ and $\gamma = 0.001$:

```
[Yxx, f]=psd(ycal, NFFT, fs, NFFT, NOVERLAP);
deltaz=0.312;
for i=2:57
deltaf=Ffreqs(i)-Ffreqs(i-1);
scal=(deltaf./deltaz);
indice=(Ffreqs(i-1) <= f & f < Ffreqs(i));
index=find(indice>0);
index_first=index(1);
index_last=index(end);
PPX(i-1)=(scal./(index_last-
index_first+1)).*sum(Yxx(Ffreqs(i-1) <= f &
f < Ffreqs(i)));
end
Lx=S1.*(P0).^gamma.*((1-
0.5+0.5.*PPX./P0).^gamma-1);
```

```
Lx(find(Lx<0))=0;
N_pseq=sum(Lx); %total loudness
```

2.3.3 Zwicker's Loudness model as used in PEAQ

This is the most sophisticated of the psycho-acoustic decompositions [12]. The power spectrum of each frame is weighted by the frequency response of the outer and middle ear derived from a model. The power spectral energies are then grouped into Critical bands, spaced at 0.25 Bark. An offset is then added to the Critical band energies to compensate for internal noise generated in the ear. A triangular (in dB) spreading function is used to implement spreading in the frequency domain. The spread excitation pattern is given by \tilde{E}_{SR} . Unlike the PSQM algorithm the values for the excitation threshold in PEAQ were computed using a model description [12], with c a constant that is set to 1.07664:

$$LX(f)_n = c \left(\frac{E_t(f)}{s(f)E_0} \right)^{0.23} \cdot \left[\left(1 - s(f) + \frac{s(f)\tilde{E}_{SR}(f)_n}{E_t(f)} \right)^{0.23} - 1 \right] \quad (4)$$

where, in terms of dB, the threshold index is given by:

$$s_{dB}(f) = -2 - 2.05 \tan^{-1} \left(\frac{f}{4000} \right) - 0.75 \tan^{-1} \left(\left(\frac{f}{1600} \right)^2 \right) \quad (5)$$

and the excitation threshold is:

$$E_{tdB}(f) = 3.64(f/1000)^{-0.8} \quad (6)$$

A complete implementation of this function is given in [13]. The variables $X2$ are the power spectrum of the frame, E_b is the bark warped spectrum, E is the spectrum following the application of spreading, and Lx and N_{tot} are the specific loudness and total loudness respectively. The functions named in the code below are the same as described in [13] but with the additional input parameters of signal length len and sampling frequency Fs .

```
X2=PQDFTFrame(ycal, len);
Eb=PQgroupCB(X2, 'Basic', len, Fs);
E=PQspreadCB(Eb, 'Basic', Fs);
[Ntot]=PQLoud(E, 'Basic', 'FFT', Fs);
N_tot(1)=Ntot;
```

2.4 Loudness Model of Moore and Glasberg

The model of Moore and Glasberg [14] is different to that of Zwicker in that the auditory frequency scale used is the equivalent rectangular bandwidth (ERB) and the equation for the specific loudness in a filter band is:

$$N' = C \left(E_{sig}^\alpha - E_{thQ}^\alpha \right) \quad (7)$$

where E_{sig} is the excitation pattern within a particular frequency band, E_{thQ} is the excitation at the hearing threshold, and C and α are constants.

A Matlab implementation was proposed by [15]. It relied on the HUTEar toolbox [7] to calculate the excitation patterns. Assuming a gammatone filterbank with 128 filters, the suggested model input parameters were:

```
model.fs=Fs;
[f, b, CentFreq]=Make_cgtbank(128, Fs, 200, 6);
save gt128_test f b CentFreq;
model.cochlea.fb.file='gt128_test.mat';
model.cochlea.asymmcomp=1;
```

```
model.haircell.rcf.r='half';
model.haircell.rcf.c=0.7;
model.haircell.rcf.f='1kHz';
model.neural.function='mean';
```

For each frame the excitation pattern of the signal was generated using the AudMod function from the HUTear toolbox [7]:

```
Esig=AudMod(ycal,model);
```

Similarly, to generate the excitation pattern at the hearing threshold:

```
[CrctLinPwr, frqNpts, CrctdB] =
    OutMidCrct2('MAF',128,Fs);
MAF = interp1
    (frqNpts,CrctdB,CentFreq,'linear','extrap');
for i=1:length(CentFreq)
    tones(i,:)=pascalize(sin(2.*pi.*
        (0:frame_len-1).*CentFreqs(i)./Fs),MAF(i));
end;
```

```
Ethq=AudMod(sum(tones),model);
```

Once the excitation patterns are known, the specific and total loudness can be calculated. To find the total loudness from the specific loudness, scaling is applied based on the bandwidth of the ERB filters before summing:

```
N=C.*(Esig.^alpha-Ethq.^alpha);
N(find(N<0))=0;
EarQ = 1/0.107939;
minBW = 24.7;
order = 1;
b = 1.019;
ERBwidth = ((CentFreq/EarQ).^order +
    minBW^order).^(1/order);
totalLoudness=sum((N.*ERBwidth)');
```

3. OUTPUT CALIBRATION AND TESTING

In the cases of the MBSD loudness model, the PSQM loudness model and the Moore and Glasberg loudness model, calibration was found to be necessary. The Matlab function `lsqcurvefit.m` from the optimization toolbox was used. One of its requirements is a function name within its input arguments, and using the MBSD loudness model as an example, it was written in the form:

```
function [l]= mbsd_cal(coef,S_XX)
l=diag(sqrt(coef.*S_XX)*sqrt(coef.*S_XX)');
```

where `coef` is the calibration parameter, `S_XX` is excitation used to compute the specific loudness and `l` is the total loudness. Sinewaves of frequency 1000Hz, 512-points in length, sampled at 16kHz and calibrated to be {40,50,60,70,80} dB SPL were used. These should have a total loudness of {1,2,4,8,16} respectively. In the case of the MBSD loudness model `S_XX` needs to be scaled by a factor of 0.2567. For the PSQM model, $S_i = 7.63 \times 10^{-4}$ and $\gamma = 0.2941$. For the Moore and Glasberg model $C = 0.0002$ and $\alpha = 0.8885$.

The total loudness in Sone produced by each model for these sinewaves is given in Table 1. It can be seen from the table

that none of the measures produce the exact figure for total loudness but that all are approximately close to the expected value.

Model	40db	50dB	60dB	70dB	80dB
MBSD	0.701	1.709	3.835	7.97	16.102
DIN45631	0.8030	1.974	4.513	9.43	18.76
PSQM	0.975	1.985	4.001	8.007	15.98
PEAQ	1.312	2.551	4.701	8.363	14.486
MooreGlas	0.9364	2.06	3.978	7.23	13.1

Table 1: Input Sinusoid SPL Values and Models outputs in Sones

3. CONCLUSIONS

This paper has presented Matlab implementations of a number of loudness models. Furthermore, where necessary the issue of model calibration was addressed. Finally, results were presented to demonstrate the model output for a sinewaves of various dB SPL levels.

4. REFERENCES

- [1] Gelfand, S.A., *Hearing: An Introduction to Psychological and Physiological Acoustics*, Marcel Dekker, 1998.
- [2] <http://hyperphysics.phy-astr.gsu.edu/hbase/sound/phon.html#c2>
- [3] Wonho, Y., Enhanced modified bark spectral distortion (EMBSD): An objective speech quality measure based on audible distortion and cognition model, Ph.D. thesis, Temple University, Ft. Washington, USA, 1999. http://www.temple.edu/speech_lab/Wonhos_Dissertation.PDF
- [4] Appell, J., et al., 'Review of loudness models for normal and hearing-impaired listeners based on the model proposed by Zwicker', *Audiologische Akustik*, 40, No.(2), 2002.
- [5] http://www.nd.edu/~atassi/Teaching/ame553/Notes/Sound_power.pdf
- [6] <http://widget.ecn.purdue.edu/~hastinga/Research.htm>
- [7] <http://www.acoustics.hut.fi/software/HUTear/>
- [8] Zwicker, E., Fastl, H., and Dallmayr, C., 'BASIC program for calculating the loudness of sounds from their 1/3 oct band spectra according to ISO 532 B', *Acustica* 55, 1984, pp. 63-67.
- [9] Quast, H., 'Absolute Perceived Loudness of Speech', *Proceedings of the 7th Joint Symposium on Neural Computation, USC*, 2000.
- [10] ITU Recommendation, *P.861 Objective Measurement of Telephone Band (300-3400Hz) Speech Codecs (PSQM)*
- [11] ITU Recommendation, *P.862 Perceptual Evaluation of Speech Quality (PESQ), the New ITU Standard for End-to-end Speech Quality Assessment*,
- [12] Thiede, T., Perceptual Audio Quality Assessment using a Non-Linear Filter Bank, Ph.D. thesis, Technische Universität Berlin, Berlin, Germany, 1999.
- [13] Kabal, P., 'An Examination and Interpretation of ITU-R BS.1387: Perceptual Evaluation of Audio Quality', *TSP Lab Technical Report*, Dept. Electrical & Computer Engineering, McGill University, May 2002. <http://www.tsp.ece.mcgill.ca/MMSP/Documents/Reports/index.html#KabalR2002>
- [14] Moore, B., Glasberg, B., Baer, T., 'A model for the prediction of thresholds, loudness and partial loudness', *J. Audio Eng. Soc.* 45, 1997, pp. 224-240.
- [15] <http://www.auditory.org/postings/2002/565.html>