Matlab code for Calculation of the Loudness of Time-Varying Sounds

Brian C.J. Moore, Brian R. Glasberg and Josef Schlittenlacher
bcjm@cam.ac.uk, bg12@cam.ac.uk, js2251@cam.ac.uk

Department of Experimental Psychology, University of Cambridge,
Downing Street, Cambridge CB2 3EB, England

I. INTRODUCTION

The Matlab code provided calculates loudness according to the model described by Moore et al. (2016). It was developed from C code for the same model, and Matlab code written for ANSI S3.4-2007, the model of Moore and Glasberg (2007) and the proposed ISO 532-2 standard. The source code is provided for free for any research purposes.

The code requires Matlab to be installed. If you use a newer version, you must provide wrap-up functions for the replaced wavwrite and wavread (see below).

The code may be used for wav-files (one or two channels). If it is a one-channel signal, the program assumes diotic presentation. To calculate the loudness of a monaural signal, a second channel filled with zeros must be added.
Furthermore, it is also possible to use Matlab variables as input. These must be column vectors as obtained by the function wavread().

II. RUNNING THE PROGRAM

Open Matlab and switch to the directory that contains tv2016.m.

The function tv2016(filenameSound, dBMax, filenameFilter, s, Fs) contains five parameters. To calculate the loudness from a wav-file, only the first three must be passed, to calculate the loudness from a Matlab variable, all five must be passed.

filenameSound:	The filename of the wav-file for which the loudness is calculated. Pass any dummy name if s and Fs are specified. In that case it is only used as a name for output files
dBMax:	The root-mean-square sound pressure level of a full-scale sinusoid, i.e. a sinusoid whose peak amplitude is 1 in Matlab.
filenameFilter:	The filename of the filter that specifies the transfer function through the outer and middle ear. Use either ‘ff_32000.mat’ for free-field presentation, ‘df_32000.mat’ for diffuse-field presentation or ‘ed_32000.mat’ when the frequency response is flat at the eardrum.
s:	A two-column variable containing the signal
Fs:	The sampling rate of signal s

III. OUTPUTS OF THE PROGRAM

The function returns three variables, each of them being vectors starting at t = 0 ms and having a step size of 1 ms. The first vector is the instantaneous loudness, the second one short-term loudness, the third one long-term loudness, all in sone.

Furthermore, the program creates a text file in the subdirectory out, having the same filename as specified in filenameSound and the extension '.txt'. It contains seven columns, specifying the time in ms, instantaneous loudness, short-term loudness and long-term loudness in both sone and phon.
Furthermore, the program creates a Matlab figure with a black line representing instantaneous loudness, a blue line short-term loudness and a red line long-term loudness.

IV. EXAMPLES

Type

```
    tv2016( '1k100ms.wav', 50, 'ff_32000.mat' )
```

to calculate the loudness for the example wav file. If a full sinusoid has 50 dB SPL, the signal in the example wav-file would have 40 dB SPL and the outputs show the loudness for a 1-kHz pure tone with a duration of 100 ms and a level of 40 dB SPL.

To calculate the loudness for your own signals stored as Matlab variables and further processing in Matlab, use

```
    [a, b, c] = tv2016( 'any', 70, 'df_32000.mat', s, Fs )
```

V. NEW VERSIONS OF MATLAB

If you have a newer version of Matlab that does not provide wavread any longer, write the following functions and store them in the working directory or wherever Matlab will find them:

```
function [s Fs] = wavread( filename );
[s Fs] = audioread( filename );
```

and

```
function wavwrite(y,Fs,nbits,wavefile)
if nargin == 3
    wavefile = nbits;
    nbits = 16;
end
audiowrite(wavefile,y,Fs,'BitsPerSample',nbits);
```

VI. SUBROUTINES

You will find many useful subroutines in the main directory and subdirectory 'functions'. They may be used to calculate excitation patterns, perform a FFT, convert sone to phon or Hz to Cam (the ERB-number), calculate the equivalent rectangular bandwidth, binaural inhibition, automatic gain circuits, and so on.
We will not provide further details for them in this manual, however, most of them are straightforward to use.

REFERENCES

ANSI (**2007**). *ANSI S3.4-2007. Procedure for the computation of loudness of steady sounds* (American National Standards Institute, New York).

ISO **(2016)**. ISO/DIS 532-2. *Methods for calculating loudness — Part 2: Moore-Glasberg method* (International Organization for standardization, Geneva)

Moore, B. C. J., and Glasberg, B. R. (**2007**). "Modeling binaural loudness," J. Acoust. Soc. Am. **121**, 1604-1612.

Moore, B. C. J., Glasberg, B. R., and Schlittenlacher, J. **(2016)** "A Loudness Model for Time-Varying Sounds Incorporating Binaural Inhibition," submitted