

## 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), but with the modified time constants described by Moore *et al.* (2018). It was developed from C code for the same model, and Matlab code written for ANSI S3.4-2007, based on Moore *et al.* (1997) and Glasberg and Moore (2006) and ISO 532-2 (2017), based on Moore and Glasberg (2007). The source code is provided free for any research purposes. For commercial use please contact Brian C. J. Moore.

The code requires Matlab to be installed. If you use a recent version of Matlab (after about 2014), you may need to provide wrap-up functions for two functions that are no longer supported, wavwrite and wavread (see below).

The code may be used with wav files (one or two channels). If a one-channel file is used, the program assumes diotic presentation. To calculate the loudness of a monaural signal, a second channel filled with zeros must be added.

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 tv2018.m.

The function tv2018( 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. The signal s and sampling rate Fs are specified, the filename is only used as a name for the 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. This allows calibration of absolute level.

**filenameFilter:** The filename of the filter that specifies the transfer function through the outer and middle ear.  
Use 'ff\_32000.mat' for free-field presentation, 'df\_32000.mat' for diffuse-field presentation or 'ed\_32000.mat' for middle-ear only (when the signal is picked up at the eardrum, or headphones with a "flat" frequency response at the eardrum are used).

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 is the short-term loudness, and the third is long-term loudness, all in sone. In addition, 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 loudness level in phon. Finally, the program creates a Matlab figure with a black line representing instantaneous loudness, a blue line representing short-term loudness and a red line representing long-term loudness.

### IV. EXAMPLES

Type

```
tv2018( '1k100ms.wav', 50, 'ff_32000.mat' )
```

to calculate loudness for the example wav file. The signal is a 100-ms segment of a 1000-Hz tone with a level 10 dB below the full-scale level. If a full-scale sinusoid has a level of 50 dB SPL (as specified by the "50" in the example above), the signal in the example wav file would have a level of 40 dB SPL and the outputs show the loudness of a 1-kHz pure tone with a duration of 100 ms and a level of 40 dB SPL. To calculate the loudness of a 1-kHz pure tone with a duration of 100 ms and a level of X dB SPL, specify the full-scale level as X+10.

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

```
[a, b, c] = tv2018( 'name', 70, 'df_32000.mat', s, Fs )
```

### V. RECENT VERSIONS OF MATLAB

If you have a version of Matlab that does not provide wavread or wavwrite, 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 Fast Fourier Transform (FFT), convert sone to phon or Hz to Cam (the units of the  $ERB_N$ -number scale), calculate the equivalent rectangular bandwidth of the auditory filter, calculate binaural inhibition, and implement automatic gain circuits, among other things.

## REFERENCES

- Glasberg, B. R., and Moore, B. C. J. (2006). "Prediction of absolute thresholds and equal-loudness contours using a modified loudness model," *J. Acoust. Soc. Am.* **120**, 585-588.
- ISO 532-2 (2017). *Acoustics - 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 Baer, T. (1997). "A model for the prediction of thresholds, loudness and partial loudness," *J. Audio Eng. Soc.* **45**, 224-240.
- Moore, B. C. J., Glasberg, B. R., Varathanathan, A., and Schlittenlacher, J. (2016). "A loudness model for time-varying sounds incorporating binaural inhibition," *Trends Hear.* **20**, 1-16.
- Moore, B. C. J., Jarvis, M., Harries, L., and Schlittenlacher, J. (2018). "Testing and refining a loudness model for time-varying sounds incorporating binaural inhibition," *J. Acoust. Soc. Am.* **143**, 1504-1513.