Loudspeaker Room Auralisation (LoRA) Toolbox

v0.4 - Quick guide – 26/03/10

Installation

- Copy the folder LoRAToolbox on your Hard drive (ex: C:\xyz\loRAtoolbox)
- Set this path in Matlab. Type `addpath('C:\xyz\LoRAToolbox')`
- The toolbox is installed

Quick guide

ODEON source files

The LoRA toolbox computes multichannel room impulse response (mRIR) from the early reflections and the energy curve exported from ODEON. Specific parameters must be set before computing a response in ODEON; a typically set of options can be found in appendix A. Make sure you are using a point source with an omnidirectional directivity pattern. The resulting amplitude of the mRIR depends on the sound power set up in ODEON in the characteristic of the point source.

For a specific job in a specific room, 2 ASCII text files need to be saved in ODEON in order to compute the mRIR. This is done by selecting the desired Job in the 'Joblist' section of ODEON; run it if not computed yet (Alt+R) then select 'View single point response' (Alt+P) to show the job response. Select the tab 'Decay curves' and press Ctrl+A to export the data in ASCII format. Save the text file in the appropriate[1] folder and make sure that the end of the file name is "(...)EnergyCurves.Txt" (Default). Then, select the tab 'Reflectogram' and type Ctrl+A. Save the ASCII text file in the same folder and make sure the end of the file is "(...)EarlyReflections.Txt", the beginning being the same as the one before.

mRIR computation

The toolbox uses a structure LoRA in most of its functions. The function `LoRA_startup` initializes the structure (see appendix) and must be run before using the function of the toolbox.

There are two ways of computing mRIR in the LoRA toolbox:

- Using a **simple function** for each room and each job:

The function `LoRAmRIR` computes mRIR for each room and job for the Ambisonic order requested giving into argument the name of the path name and the file name of the room and job.

```
exampleLoRA.m

% Initialisation
LoRA_startup('fs',44100,...
             'LoudSetName','LoudspeakersPos3D')
path = 'C:\user\add-on\LoRAToolBox\Example\';
room = 'Elmia RoundRobin2 detailed.Job02.00001';

% Compute each part of the response (mRIR)
[mIRearly,ylate,Param] = LoRAmRIR(path,room,LoRA.renderDS,LoRA.renderER);
% Add the direct sound the early reflections and the late reflections
ymRIR = AddDSERlate(mIRearly,ylate);
```

[1] The appropriate folder would be the one specify in `LoRA.PathReadODEON` for batch processing.

The file `loudspeakersPos3D.m` provides the azimuth and the elevation of the loudspeakers (here corresponding to the array from the CAHR SpaceLab). The function `AddDSERlate` is used to sum the direct sound, the early reflections and the late reflections responses.

- Using **batch processing** (automated computation of a large set of mRIR):

This method automatically computes a set of mRIR and/or convolved with the desired anechoic sound files.

```
examplebatch.m

% Initialisation
LoRA_startup('fs',44100,'Session','Demo3D',...
             'LoudSetName','LoudspeakersPos3D',...
             'renderDS',4,...
             'renderER',4,...
             'PathStoreIR','C:\user\DemoLoRA\mRIRs\',...
             'PathStoreConv','C:\user\DemoLoRA\ConvFiles\',...
             'PathReadODEON','C:\user\DemoLoRA\Odeon_data\',...
             'PathSoundSamples','C:\user\DemoLoRA\AnecSound\')

% Computes mRIR for the room files in PathReadODEON and then convolve them
% with the anechoic sound sample in PathSoundSample.
batchLoRAProc
```

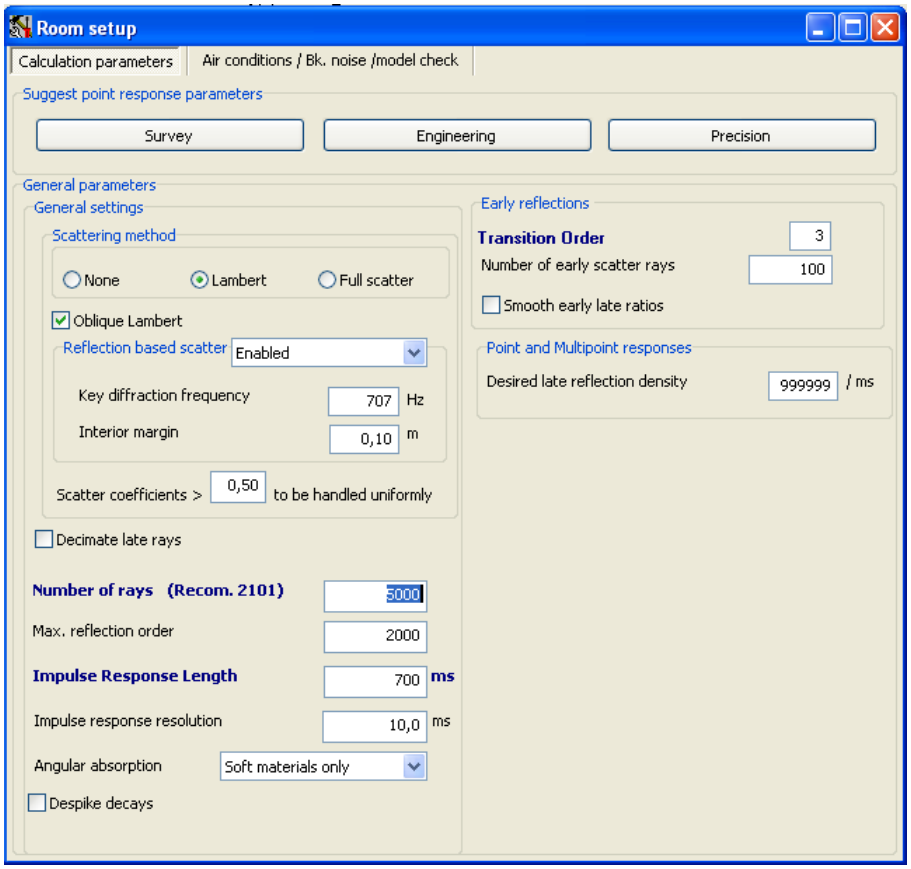
The function `batchLoRAProc` asks for which room and job to compute the mRIR. Then, the anechoic sound files should be chosen to convolve. In the convolution process the result is calibrate for the system (if calibration info available) and store the multichannel sound file in the path defined by `PathStoreConv` ready to be played.

Type ‘help `batchLoRAProc’` for more information.

Appendix A

Options to be used in Odeon for job computations

The parameter “Number of Rays”, Max reflection order and “Impulse response Length” can be modified according to the room characteristics



LoRA structure

The LoRA structure is initialized by `LoRA_startup` and store information such as path names, frequency sampling, filter bank, calibration information…

For more information see the header of `LoRA_startup`

```
%% Build LoRA structure
LoRA     = struct(                              ...
    'session',           session,               ... Name of the session
    'PathStoreIR',       PathStoreIR ,          ... Path where Impulse responses are stored
    'PathStoreConv',     PathStoreConv,         ... Path where Convolved sound are stored
    'PathReadODEON',     PathReadODEON,         ... Path where room data text files are stored
    'PathSoundSamples', PathSoundSamples,       ... Path where anechoic sound files are stored
    'PathLoRAlog',       pathlog,               ... Path where logs are stored
    'PathCal',           pathcal,               ... Calibration path
    'fs',                fs,                    ... Sampling frequency for response computation
    'LoudSetName',       LoudSetName,           ... Name of the Loudspeaker array
    'LoudR',             LoudR,                 ... Loudspeaker array radius (for NFC filters)
    'pos',               Loudset,               ... Loudspeakers position (azimuth,elevation)
    'nL',                nL,                    ... Number of loudspeakers
    'isnfc',             isnfc,                 ... Boolean, is NFC filters used (1) or not (0)
    'isAmb2D',           isset2D,               ... Horizontal-only (1) or full 3D processing (0)
    'DecMat',            DecMat,                ... Decoding matrix
    'h',                 h,                     ... impulse response of the octave filter bank
    'H',                 H,                     ... freq response of the octave filter bank
    'hlen',              len,                   ... length of the filters
    'renderDS',          renderDS,              ... ambisonic order for the direct sound
    'renderER',          renderER,              ... ambisonic order for the early reflections
    'rendering',         rendering,             ... rendering
    'hm',                EQ.hm,                 ... Loudspeaker equalization filter
    'CalSPLFact',        EQ.CalSPLFact,         ... Factor to derive SPL levels
    'CalEqdB',           EQ.CalEqdB,            ... Mean of the SPL level of each loudspeakers (input
ampref RMS)
    'CalTimeSync',       EQ.CalTimeSync,        ... Time synchronisation accross channel
    'ampref',            0.08,                  ... reference amplitude for calibration
    'history',           history                ... history: rooms and sound selection
    );
```