Technical reference: FVN (Frequency domain Velvet Noise) tools

Hideki Kawahara

3:59pm, September 15, 2019
(Start date: September 6)

Contents

List of Figures

List of Tables

1 Introduction

Frequency domain variants of velvet noise (FVN)[1, 2] forms a new family of test signals for impulse response and nonlinearity measurements (for acoustic system measurements). This technical document introduces tools based on FVN[3] and constituent MATLAB functions. Please refer to our preprint[4] for APSIPA ASC 2019.

2 `allInOneFVNmeasurement`:MATLAB function for one stop acoustic measurement

This function is a wrapper of all necessary steps for FVN-based acoustic system measurements; such as calibration, measurements, analysis, and report.

2.1 Description

This function measures attributes of acoustic systems. The attributes are impulse response, background noise, and components generated by deviations of the target systems from time-invariant linear systems. This function also provides necessary procedures for calibration of the recording subsystem, analysis of measured attributes, and report generation of the results.

2.2 Measurement procedure

Type the following command in the MATLAB command window. It displays the following help text of this function.

```
allInOneFVNmeasurement
```

2.2.1 System requirement

Measurement using this function requires a machine which has MATLAB installed and connected to a microphone and a loudspeaker. The AD converter and the DA converter need to share the same clock source for conversion. Calibration requires the microphone and the microphone of a sound level meter have to set close together.

2.2.2 Calibration: with default condition

Type the following command in the command window.

```
allInOneFVNmeasurement('calibration')
```

It displays the following information, and a noisy sound starts from the loudspeaker.

```
Test signal plays 13 seconds.
Adjust the test signal level to 80 dB at the microphone
```

Adjust the output level control of the DA converter of the audio interface to make the sound pressure level (measured using A-weighting) 80 dB.

One of the following messages is displayed depending on the result.

Sensitivity of the input is proper:

```
OK! MaxValue: XXXX
```

"XXXX" represents the maximum absolute value of the input signal.

Sensitivity is too low:

```
Increase the mic sensitivity. MaxValue: XXXX
```

Sensitivity is too high:

```
Reduce the mic sensitivity. MaxValue: XXXX
```

You can try the calibration again by typing:

```
allInOneFVNmeasurement('calibration')
```

The default signal duration is rather short. For the first session, please set the sound pressure level. Then, in the next session, please adjust the input level.

2.2.3 Measurement: default setting

Type the following command in the command window.

```
allInOneFVNmeasurement('XXXXXXXX')
```

You can write any text other than "`calibration`" in the parentheses.

Then, the following message appears and starts noisy test sound from the loudspeaker.

```
Measurement starts. Be quiet for 16 seconds. Please.
```

After several seconds from the end of the test sound, a plot shown in Fig. 1 appears.

```
                      fvnFile: 'fvnMin200ms'
      averagedThirdOctaveResponse: [1 × 16384 double]
  averagedThirdOctaveNonlinearComp: [1 × 16384 double]
     averagedThirdOctaveBackground: [1 × 16384 double]
         averagedResponseWaveform: [15876 × 1 double]
       individualResponseWaveform: [15876 × 4 double]
       averagedBackgroundWaveform: [15876 × 1 double]
     individualBackgroundWaveform: [15876 × 4 double]
                         memoText: 'test'
                    number_of_FVNs: 4
                    responseLength: 0.3000
                soundPressureLevel: 80
                       signalRange: [294135 310010]
```

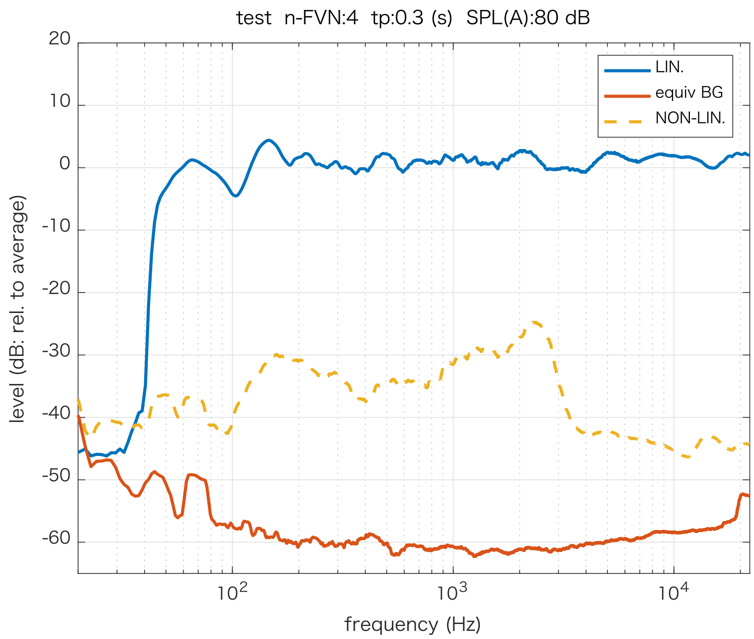

Figure 1: Plot example of the results. (blue line) The amplitude-frequency response of the system. In other words, the response of the linear component. (red line) Effects of the background noise. (dashed yellow line) Response due to nonlinearity. These are the RMS (root mean square) values with one-third octave rectangular smoothing.

```
   backgroundRange: [792464 808339]
 samplingFrequency: 44100
   modeOfOperation: 'measurement'
      creationDate: '10-Jul-2019 23:46:27'
    levelStructure: [1 × 1 struct]
       elapsedTime: 37.1656
  recordedWaveform: [1510400 × 1 double]
```

At the end of the measurement, this function outputs the following files:

```
-rw-r--r--    1 kawahara  staff     961551  7 10 23:46 fvn44k20190710T234626.eps
-rw-r--r--    1 kawahara  staff    4531244  7 10 23:46 fvn44k20190710T234626.wav
-rw-r--r--    1 kawahara  staff   93931780  7 10 23:46 fvn44k20190710T234626.mat
```

The file names are unique by using the time stamp. The file with the extension “`eps`” saves the figure. The file with the extension “`wav`” saves the recorded signal of this measurement. The file with the extension “`mat`” saves the conditions and the results of the measurement. This file does not consist of the recorded signal.

Table 1: Typical throughput time for argument setting.

response length	#FVN=2	#FVN=3	#FVN=4	#FVN=5	#FVN=6
0.3 (default)	18	24	34	55	98
0.2	10	14	21	35	63
0.1	5	7	10	17	31

2.2.4 Analysis and reporting

Default condition generates a file with extension "`eps`" consisting of the default figure output. We provide another tool to generate detailed analysis results by reading the output file with extension "`mat`" and extension "`wav`."

2.3 Measurements and calibration in other (non-default) consitions

The function "`allInOneFVNmeasurement`" has the following generic calling sequence.

```
output = allInOneFVNmeasurement(memoText, number_of_FVNs, response_length, spl, exmode)
```

Arguments other than the first one are optional. User can set non-default conditions by setting values to these optional arguments.

2.3.1 Simplified measurement: measurement only linear response

Measuring only the linear component reduces the total duration of the measurement. Setting the value of the argument "`number_of_FVNs`" invokes this simplified measurement. The argument "`response_length`," which sets the duration of the target response also determines the total duration of the measurement. For rooms with short (about 0.1 s) reverberation time and the distance between the loudspeaker and the microphone is about one to two meter, setting the value to 0.1 s provides the similar results to the default condition. Table 1 shows the typical measurement time for different settings.

Figure 2 shows examples of simplified measurement. The measured amplitude responses are close to each other. The effect of the background noise depends on the number of FVN sequences. In these plots, the test signal consisting of three FVN sequences provides the smallest effects because the number of independent measurements is the maximum.

2.3.2 Measurement of linear and non-linear components : Level dependency

Effects of nonlinearity depend on the sound pressure level. Calibration of the sound acquisition system is essential for making this measurement reproducible. Setting the sound pressure level at the microphone using the argument "`spl`" is for this calibration.

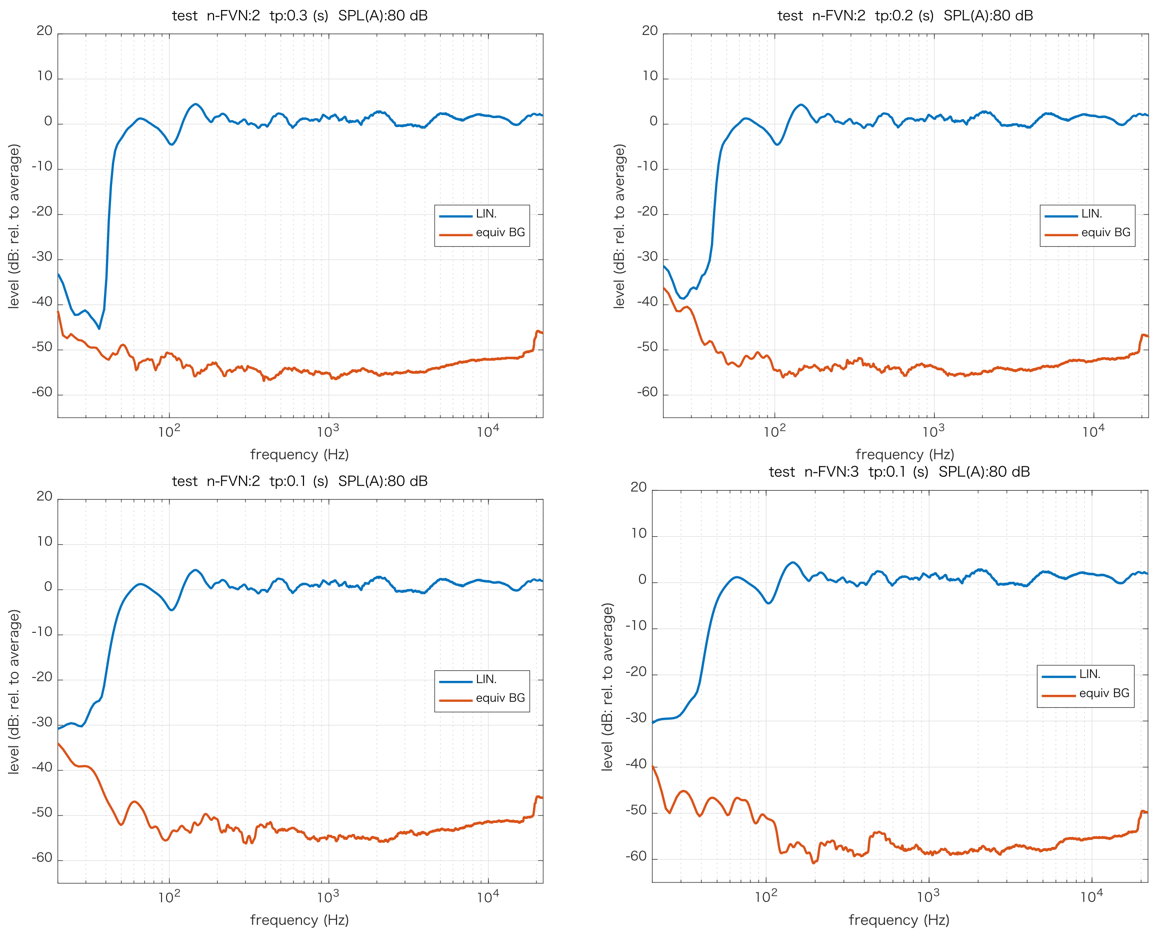

Figure 2: Plot examples of simplified measurement. Note that only the linear and background components are displayed.

The following example uses 70 dB and 85 dB for calibrating the recording system. The distance from the loudspeaker and the microphone is 50 cm. Typing the following command and set the sound pressure level at the microphone to 70 dB by adjusting the test signal output level using a sound level meter. Then, adjust the input sensitivity.

```
allInOneFVNmeasurement('calibration',4, 0.3, 70)
```

The next step is the measurement of the system. Type the following command.

```
allInOneFVNmeasurement('test',4, 0.3, 70)
```

Replace the “`test`” in the argument with a note to describe this experiment.

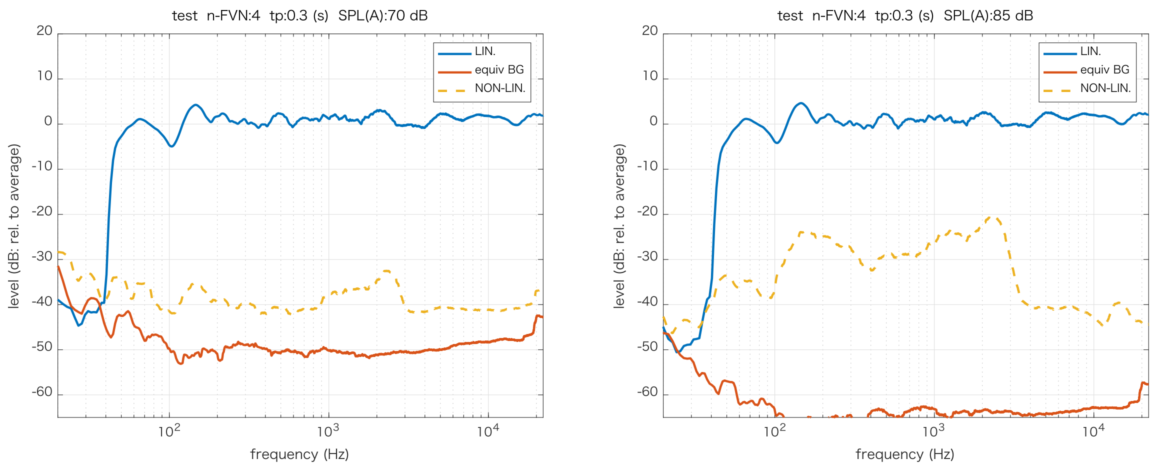

Figure 3: Level dependency of the non-linear component.

Repeat a similar step for the calibration level 85 dB. Note that the sound pressure level in this function uses A-weighting.

Figure 3 shows the results. The level of nonlinear component for 85 dB is more than 10 dB higher than 70 dB condition. (FYI, the author's preferred sound pressure level for listening to music is from 65 dB to 70 dB at around 1 m from the loudspeakers.)

2.3.3 Other arguments

The argument "`exmode`" enables to extend the measurement scheme. Currently (13/Sept./2019) an available extension is "`measurement2ch`". It is for two-channel measurement. It records two input signals. For example, by connecting the monitor signal to the right channel, it is possible to measure propagation delay from the signal output to the loudspeaker to the captured sound by the microphone.

2.4 Help texts

2.4.1 Calling sequence:

```
% Acoustic impulse response and nonlinear component measurement
% How to call
% output = allInOneFVNmeasurement(memoText)
% output = allInOneFVNmeasurement(memoText, number_of_FVNs)
% output = allInOneFVNmeasurement(memoText, number_of_FVNs, response_length)
% output = allInOneFVNmeasurement(memoText, number_of_FVNs, response_length, spl)
% output = allInOneFVNmeasurement(memoText, number_of_FVNs, response_length, spl, exmode)
```

2.4.2 Description of arguments

```
% Argument
%  memoText        : text string
%  number_of_FVNs  : number of FVN sequences for nonlinearity measurement
%                    must be equal or larger than 4.
%                    The default value is 4.
%  response_length : observation length of the impulse response (s)
%                    The default value is 0.3 (s)
%  spl             : sound pressure level at microphone in A-weighting (dB)
%                    The default value is 80 dB
%  exmode          : execution mode selector
%                    'measurement', 'measurement2ch', 'calibration', 'diagnosis'
%                    default 'measurement'
```

2.4.3 Description of output

```
% Output
%   output         : structure with the following fields
%                    Field name explains itself
%        averagedThirdOctaveResponse: 1-d vector double
%    averagedThirdOctaveNonlinearComp: 1-d vector double
%      averagedThirdOctaveBackground: 1-d vector double
%           averagedResponseWaveform: 1-d vector double
%         individualResponseWaveform: matrix
%        averagedBackgroundWavelform: 1-d vector double
%       individualBackgroundWaveform: matrix
%                           memoText: text string
%                     number_of_FVNs: integer
%                     responseLength: double (s)
%                  soundPressureLevel: double (dB)
%                        signalRange: indecies of start and end
%                    backgroundRange: indecies of start and end
%                   samplingFrequency: double (Hz)
%                       creationDate: text string
%                     levelStructure: structure with calibration info.
%                        elapsedTime: double (s)
%                    recordedWaveform: 1-d vector double
```

3 `oneThirdSpecDisplayForFVN`: Detailed report generation

This function generates the recording condition report from the saved files by the one-stop acoustic measurement function using FVN, "`allInOneFVNmeasurement`." The report consists of the sound pressure levels of the recorded signal, the background noise, and the power spectrum.

3.1 Description

It plots the sound pressure level trajectories for "fast" and "slow" time constants defined in the standard of the sound level meter. It also generates plots of the power spectrum and the smoothed power spectrum using one-third octave rectangular smoother.

3.2 Procedure

The following commands read the recorded file by the function "`allInOneFVNmeasurement`" and pass the data, sampling frequency, and the calibrated sound pressure level to this reporting function.

```
>> [x,fs] = audioread('fvn44k20190710T234626.wav');
>> oneThirdSpecDisplayForFVN(x, fs, 80)
```

where ">>" is the system prompt of MATLAB.

The function produces the plots shown in Fig. 4 and output the following results.[1]

```
               sigWavePower: 2.3887e+08
                  time_axis: [1510400 × 1 double]
                  fastSPLdB: [1510400 × 1 double]
                  slowSPLdB: [1510400 × 1 double]
  rawLevelAnalysisStructure: [1 × 1 struct]
        signalPowerSpectrum: [1048576 × 1 double]
    backgroundPowerSpectrum: [1048576 × 1 double]
     frequencyAxisPowerSpec: [1048576 × 1 double]
                       fftl: 1048576
          samplingFrequency: 44100
           signalOneThirdDB: [524288 × 1 double]
       backgroundOneThirdDB: [524288 × 1 double]
      frequencyAxisOneThird: [524288 × 1 double]
          calibrationFactor: 113.3825
```

[1]The sounds are recorded in midnight in a room of a quiet suburban house. The microphone (DPA-4066) has a low noise level (in the equivalent sound pressure level of A-weighting 26 dB). The spectral shape of the background noise in Fig. 4 indicates that the background noise in higher frequency region (higher than 1 kHz) is due to the microphone.

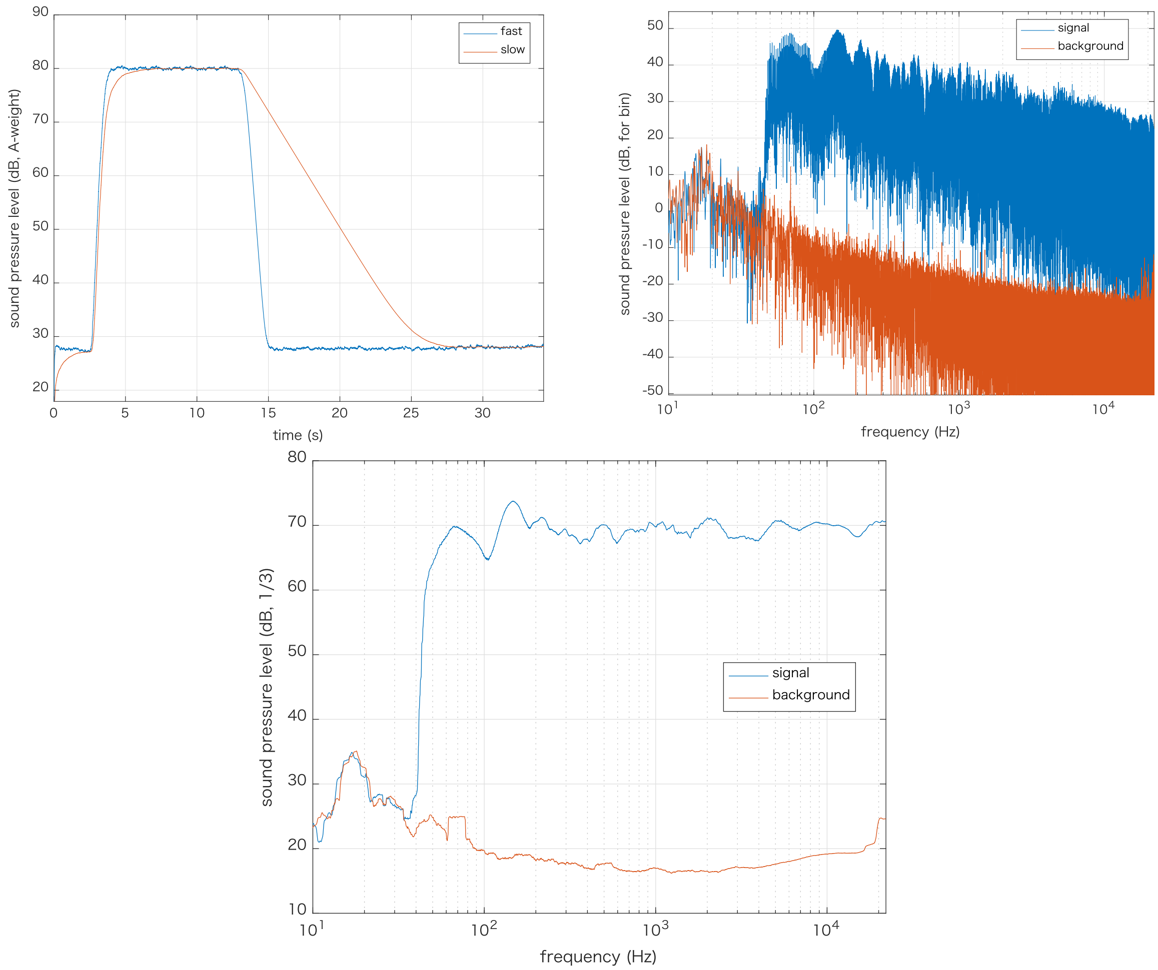

Figure 4: (top left) Trajectories of the sound pressure level in A-weighting. (top right) Power spectra of the test signal and the background noise. The vertical axis represents the sound pressure level at each FFT-bin. (bottom) Smoothed power spectra using the rectangular smoother having one-third octave width of the center frequency. The horizontal axis represents the center frequency.

The output of the function `oneThirdSpecDisplayForFVN` consists of enough information to plot figures shown in Fig. 4. The calibration information in the field "`calibrationFactor`" (in this example, $c_{f\mathrm{dB}} = 113.3825$) enables to convert the recorded digital value $x(t)$ to the value $p(t)$ represented using a physical unit (for example Pa).

$$p(t) = 20 \times 10^{-6} 10^{\frac{c_{f\mathrm{dB}}}{20}} x(t), \tag{1}$$

where 20×10^{-6} is the coefficient to define the reference sound pressure level 0 dB (20μPa).

The function `oneThirdSpecDisplayForFVN` does not save the displayed figure. Save the displayed figure using "`print`" command if necessary.

3.2.1 Calling convention

The following is a generic calling sequence of the function "`oneThirdSpecDisplayForFVN`".

```
output = oneThirdSpecDisplayForFVN(x, fs, spl, displayOn, msgText)
```

The argument "`x`" is a one-dimensional column vector representing recorded sound data. The argument "`fs`" represents the sampling frequency in Hz. The argument "`spl`" represents the sound pressure level used for calibration. These are the necessary arguments and cannot be omitted.

The argument "`displayOn`" is a switch for displaying the figures (1: display on, 0: display off). The default value is "1". You can set any text to the argument "`msgText`."

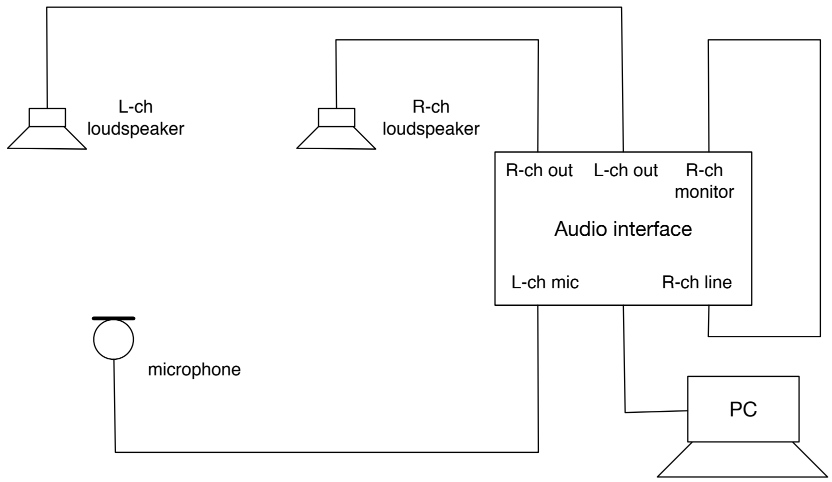

Figure 5: Schematic diagram for using this real-time and interactive acoustic measurement tool. The target is the acoustic paths from two loudspeakers to the microphone.

4 `standingWvViewer`:Real-time and interactive acoustic measurement tool

This application provides real-time and interactive acoustic measurement based on FVN sequences.

4.1 System setup

Figure 5 shows a schematic diagram of the measurement system. This setting measures acoustic attributes the paths from the two loudspeakers to the microphone. The attributes are the amplitude transfer function, the impulse response, the amplitude transfer function in the low-frequency range, reverberation, and the sound pressure level.

We use two FVN sequences which are orthogonal. The test sequence for the left channel allocates the unit FVN using the same polarity. The test sequence for the right channel allocates the unit FVN using alternating polarity. The microphone captures the mixed signal from the left and the right loudspeakers. The microphone is connected to the left input channel (channel-1). The right input channel is connected to the right channel of the monitor output of the audio interface. The sampling frequency is 44100 Hz. The DA and the AD converter shares the same sampling clock.

4.2 Measurement procedure

Connect the target loudspeakers and the microphone to the measuring system by adopting Fig.5. Then, type the following command.

```
standingWvViewer
```

The command invoked the GUI of the tool. The upper image of Fig. 6 shows the initial state of the GUI. Within several seconds, the test signal, which sounds like noise, starts. For about nine seconds from invoking the tool, the tool starts shows the measurement results. Calibration is necessary for reproducible measurements.

First, adjust the output level of the audio interface to set the sound pressure level at the microphone to one of the following values, 70, 75, and 80 dB. Then, click the appropriate radio button in the panel titled "SPL cal.".

4.2.1 Calibration of the microphone sensiivity

This stage uses only the rightmost bar graph showing the input level to set the appropriate input level. Keep the peak level (red line) not to exceed -6 dB. This adjustment is a practical guideline. Make the length of the green bar as long as possible while keeping peak level lower than -6 dB.

Place the sound level meter (IEC class-1 or 2) close to the microphone for monitoring the sound pressure level. Adjust the output level of the audio interface to make the sound pressure level 70, 75, or 80 dB (using C-weighting). While this adjustment, re-adjust the sensitivity of the microphone input to keep the condition mentioned in the previous paragraph. When the conditions are ready, click one of the relevant radio buttons in the "SPL cal." panel. This click completes the calibration procedure and starts proper measurement. Figure 7 shows the snapshot of the calibrated GUI. The calibration level, in this case, is 80 dB.

4.2.2 Microphone positioning and errors in measurement

The GUI updates displayed information in real-time, in other words, time-variable. However, the procedure for measuring impulse response using FVN sequences assumes that the measured system is linear and time-invariant. Violating this assumption introduces errors in the measurement results. This implementation also assumes that the reverberation time is shorter than 200 ms and the repetition interval is 300 ms. The length of the unit FVN used in the test signal is about ten times longer than this repetition interval. The total length involved in signal compression requires the microphone to stay at a position

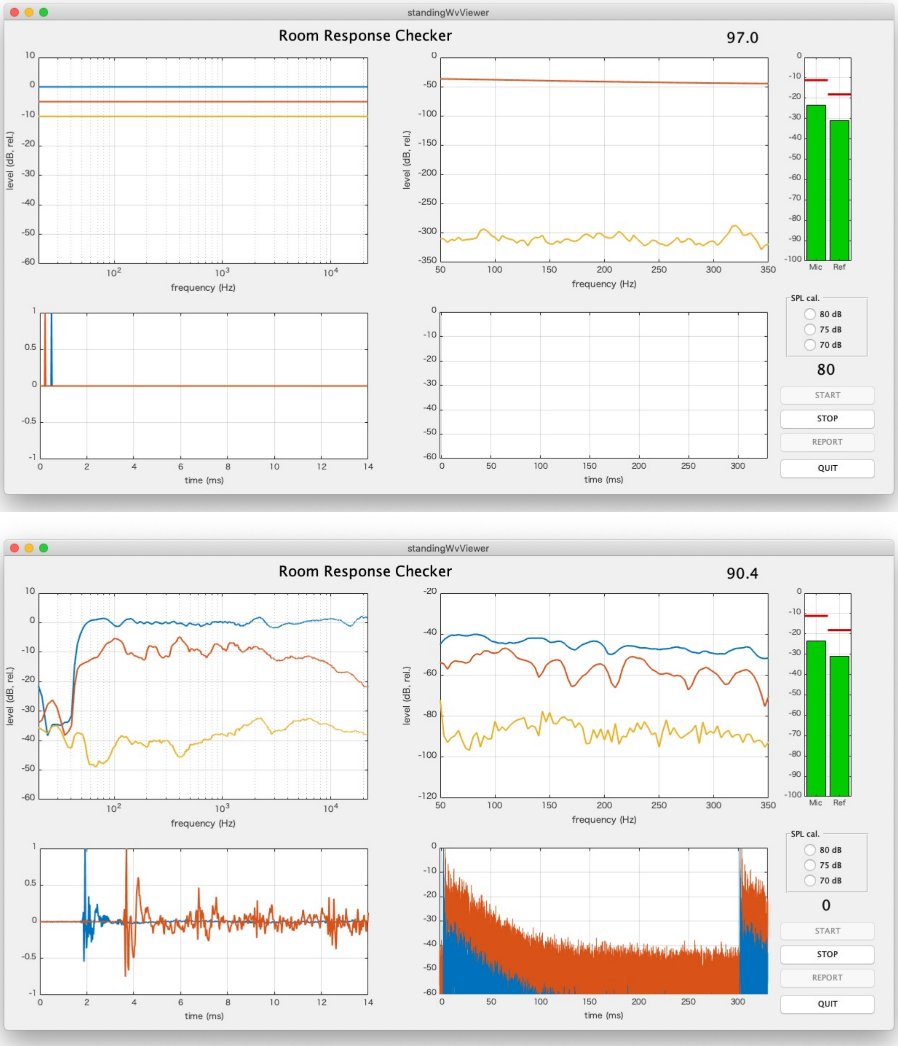

Figure 6: Snapshots of the initial state of the GUI of the acoustic measurement tool. The rightmost bar graph show the input level of the audio interface. The maximum input level is normalized to 0 dB. After nine seconds, the GUI starts to show the measured results as shown in the bottom image. However, they are not calibrated.

about four to five seconds for the measurement to yield reliable results. The GUI provides information about errors in the measurement results.

The test sequences made from FVN allocation has frequency components only at integer multiples of $1/(2t_r)$ where t_r represents the repetition interval. The residual by removing these harmonic components consists of the background noise and components generated by non-linear responses. Yellow lines in the transfer function figures show the RMS value

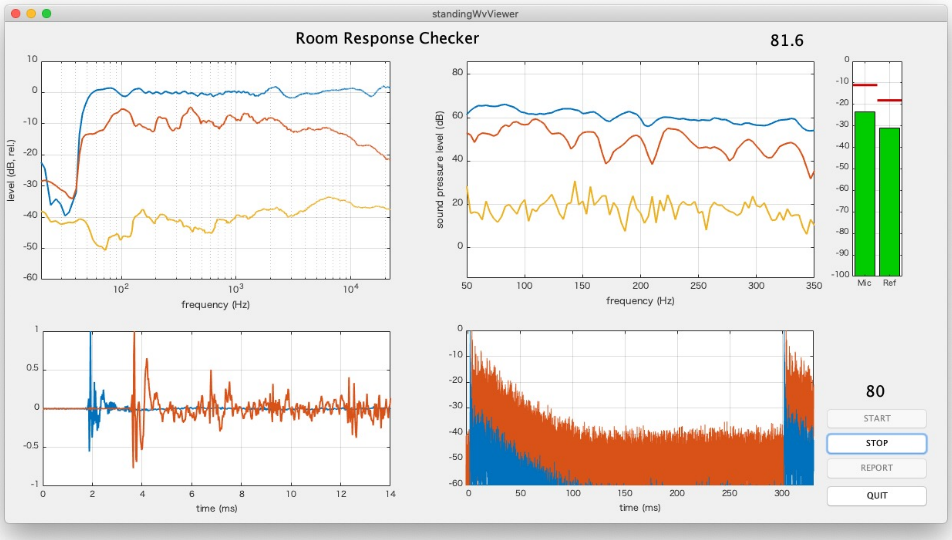

Figure 7: Snapshot of the calibrated GUI. Top left panel shows the smoothed amplitude transfer function using one-third octave smoother. Bottom left panel shows the impulse responses. The top right graph shows the amplitude transfer function without smoothing in low-frequency range (50 Hz to 350 Hz). The rightmost bar graph shows the input level relative to the maximum input level (MSB is 0 dB).

of these components.

4.3 Details of the displayed information

The following sections describe the information shown in Fig. 7. The GUI updates the displayed information in real-time. MATLAB codes of the function `stadingWaveViewer` starting from the line 255 implement updating the information as an internal function `update_display`.

4.3.1 Smoothed amplitude transfer function with one-third octaave : Top left

The top left panel of Fig.7 shows the smoothed amplitude-frequency response. The rectangular smoother having one-third octave width smooth the power spectra calculated from the impulse responses and the residual signal. The blue line shows the response of the left channel and the red line shows that of the right channel. The yellow line shows errors.

The RMS average level of the highest response normalized all responses. This normalization makes the average level of the highest response to 0 dB. The averaging uses every 1/24 octave points from 100 Hz to 10 kHz. Figure 7 selects the left channel as the highest response and sets its average level to 0 dB.

4.3.2 Impulse response from 0 to 14 ms: Bottom left

The bottom left panel of Fig. 7 shows the first part of the impulse responses. The maximum absolute value of each channel normalized each impulse response. The blue line represents the left channel, and the red line represents the right channel. In this setting, the left channel has a short (26 cm) propagation path. This short path makes the direct response dominant, and the responses due to reflections are too small to be visible.

On the other hand, the response of the right channel, which has longer (85 cm) propagation path, shows many indirect responses where the direct response dies out. The origin of the time axis is the location of the impulse. Note that the tested loudspeaker (IK Multimedia iLoud Micro Monitor) uses about 1 ms for its internal digital signal processing.

4.3.3 Whole impulse response: dB amplitude: Bottom right

The bottom right panel shows the whole impulse response (and the first part of the next cycle of the repetition). The blue line represents the left channel, and the red line represents the right channel. The vertical axis uses dB to represents the absolute value of the response because responses decay exponentially, and consequentially, they have a wide dynamic range. The maximum value of each channel is normalized to 0 dB.

4.3.4 Display for monitoring the room acoustics (standing wave): Top right

The top right panel displays the raw power spectra at harmonic frequencies. The left channel has components at $nf_{\mathrm{o}} = 1/t_r$. The right channel has components at $(n - 1/2)f_{\mathrm{o}}$. The vertical axis represents the calibrated sound pressure level, where 0 dB = 20μPa.

The blue line represents the left channel, and the red line represents the right channel. The yellow line represents the error. The left channel in which the direct response dominates shows a virtually flat and dip-less shape. On the other hand, the right channel where indirect sounds are non-negligible shows many dips reflecting interferences due to the standing wave of the room.

4.4 GUI operation elements: buttons

This tool is running while the counter indicated at the top right is positive. Control buttons which are irrelevant in this mode are disabled and dimmed. The GUI has the following buttons:

START Re-starts this tool when it is not running.

STOP Stop real-time measurement and update

REPORT Save the GUI image and the results when stoped. The file names are unique by using the time stamp.

QUIT Terminates the tool.

4.5 Contents of the saved data

The "`REPORT`" button saves the structure variable "`measuredData`" with the following fields.

```
measuredData =

  struct with fields:

          recordedWave: [308701 × 2 double]
                 x1Org: [79380 × 1 double]
                 x2Org: [79380 × 1 double]
                 x3Org: [79380 × 1 double]
                    x1: [79380 × 1 double]
                    x2: [79380 × 1 double]
                    x3: [79380 × 1 double]
                 x_ref: [79380 × 1 double]
    reference_location: 27780
              zero_idx: 88
              calLevel: 80
             levelBias: 111.1546
     samplingFrequency: 44100
```

The following describes each field:

`recordedWave` recorded waveform

`x1Org, x2Org, x3Org` Compressed responses using corresponding unit FVNs. "`x1Org, x2Org, x3Org`" represents the left, right, and the residual signal respectively.

`x1, x2, x3, x_ref` Whitened compressed responses using the inverse filter of the LPC-shaper for approximating the pink-noise spectrum. They are left, right, residual and the reference signal of the loop-back monitor.

`reference_location` Location of the impulse position of the loop-back monitor.

`zero_idx` Bias for display: sample numbers for 2 ms duration.

`calLevel` The sound pressure level used for calibration (dB using A-weighting)

`levelBias` Calibration constant which converts the digital recorded data to the value measured using physical unit Pa.

`samplingFrequency` Sampling frequency (Hz)

References

[1] 河原英紀, 津崎実, 坂野秀樹, 森勢将雅, 松井淑恵, and 入野俊夫, "velvet noise とその変種の聴覚心理・生理研究への応用可能性について," **電子情報通信学会技術研究報告**, vol. IEICE-117, no. 470(HIP), pp. 99–104, 2017.

[2] H. Kawahara, K.-I. Sakakibara, M. Morise, H. Banno, T. Toda, and T. Irino, "Frequency domain variants of velvet noise and their application to speech processing and synthesis," in *Proc. Interspeech 2018*, Hyderabad, India, 2018, pp. 2027–2031.

[3] 河原英紀, 榊原健一, and 水町光徳, "周波数領域 velvet noise を用いた音響計測手順の拡張について," **電子情報通信学会技術研究報告（応用音響）**, vol. IEICE-119, no. 115, pp. 77–82, 2019.

[4] H. Kawahara, K.-I. Sakakibara, M. Mizumachi, H. Banno, M. Morise, and T. Irino, "Frequency domain variant of velvet noise and its application to acoustic measurements," *arXiv preprint arXiv:1909.04301*, 9 2019.