Retrospective use of signal safeguarding powered by giant-FFT method

Hideki Kawahara[1], Ken-Ichi Sakakibara[2], and Kohei Yatabe[3]
[1]Emeritus Professor, Wakayama University, Japan
[2]Associate Professor, Health Science University of Hokkaido, Japan
[3]Associate Professor, Tokyo University of Agriculture and Technology, Japan

start at: 15 June 2025 14:18 JST
(updated: 16 June 2025, June 17, 2025)

Contents

1 Introduction

We introduced “signal sefeguarding” [1] to make (virtually) all sounds suitable for acoustic impulse response measurements. This research memo introduces a highly valuable procedure that applies “signal safeguarding” to previously acquired acoustic measurement data retrospectively.

The following MATLAB-like codes show the general idea. `xOrg` is a vector variable consisting of the test signal. `yOrg` is a vector variable consisting of the out of a system to the test signal input. `xOrg` is a vector variable consisting of the safeguarded test signal.

```
iRespOrg = ifft(fft(yOrg)./fft(xOrg));
iRespSg = ifft(fft(yOrg)./fft(xSg));
```

The variable `iRespOrg` consists of the estimated impulse response of the target system using the standard procedure. The variable `iRespSg` consists of the estimated impulse response of the target system by replacing the original test signal with the safeguarded test signal. **Note that the numerator is identical.**

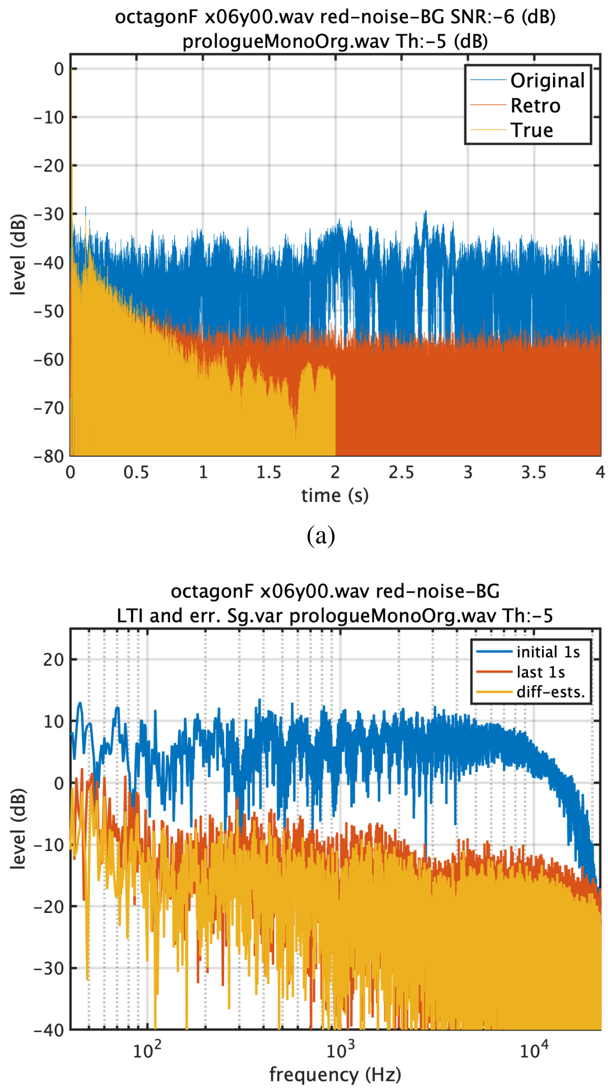
Figure 1 illustrates this retrospective application. The signal is a monaural version of a song “Prologure” in the RWC music database [2, 3]. The song’s length is 4 minutes and 58 seconds. In this simulation, we added a red noise to the system output to make observation SNR to −6 dB.

The upper plot (a) shows the decay of the impulse response estimates and the truth. Using the original test signal `xOrg` introduces a significant error (blue line). Replacing the original test signal with the safeguarded signal reduced the error by about 20 dB. Within the initial 0.5 s, this safeguarded result is close to the true (the target system’s original) impulse response.

The lower plot (b) shows the frequency response of the estimated impulse response and error. The initial 1 s of the estimated impulse response represents the system’s impulse response (blue line). The last 1 s of the estimated impulse response represents the interference due to observation noise (red line). The yellow line shows the error between the true value and the estimated impulse response.

Note that this illustration is not the theoretical best result. We used heuristics to design the parameters used in signal safeguarding. The optimized design of signal safeguarding is for future research.

This research memo is supplemental material for writing journal paper(s). This memo compiles revised formulations of scattered information on the method, fixes, and extensions of ’signal safeguarding’ from our references [4, 5]. The proposal of “giant-FFT sampling rate conversion” [6] motivated this reformulation and led to the idea of “retrospective application of signal safeguarding.”



(a)

(b)

Figure 1: (a) The decay of the impulse response estimates and the truth. (b) The frequency response of the estimated impulse response and the error estimates.

2 Giant-FFT-based signal safeguarding

“Giant-FFT” is FFT. This naming is a reminder of the huge progress (more than billion times powerful [7]) in computational power in the last half-century. The highly efficient Fourier transform [8] also mede the following idea practical.

Let’s start from a brief review of the discrete Fourier transform and signal safeguarding with vector notation.

2.1 Discrete Fourier Transform: DFT

Let $\mathbf{F}$ represents the DFT matrix consisting of (p, q)-th element $F_{p,q} = (1/\sqrt{L})\mathrm{e}^{\gamma(p-1)(q-1)}$ ($\gamma = -2\pi\sqrt{-1}/L$). ($\mathbb{i}$) Let DFT of a L dimensional vector $\mathbf{x}$ consisting of disctrete time signal represents a L dimensional vector $\mathbb{X}$. Then, the following equations define DFT and the inverse DFT.

$$\mathbb{X} = \mathbf{F}\mathbf{x}, \quad \mathbf{x} = \mathbf{F}^{\mathrm{H}}\mathbb{X}, \tag{1}$$

where the symbol H represents the conjugate transpose.

2.2 Impulse response

The output $\mathbf{y}$ of a linear time invariant system (let its impulse response in the discrete time domain $\mathbf{h}$ and $\mathbb{H}$ in the frequency domain) for the input $\mathbf{x}$ is below:

$$\mathbf{y} = \mathbf{F}^{\mathrm{H}}\mathbb{Y}, \ \textit{i.e.} \ \ \mathbf{y} = \mathbf{F}^{\mathrm{H}}\left(\mathbb{H} \odot \mathbb{X}\right), \tag{2}$$

where $\odot$ represents the Hadamard product.

Then, the inverse DFT of the element-wise division of the output $\mathbf{Y}$ by the input $\mathbf{X}$ yields the impulse response $\mathbf{h}$.

$$\mathbf{h} = \mathbf{F}^{\mathrm{H}}\left(\mathbb{Y} \oslash \mathbb{X}\right), \tag{3}$$

where $\oslash$ represents the Hadamard division.

2.3 Signal safeguarding

Acoustic measurements in the real world inevitably consist of the observation noise $\mathbf{r}$ and $\mathbb{R}$ (in the time and frequency). The estimated impulse response $\mathbf{h}_{\mathrm{est}}$ using the observed singal $\mathbf{s} = \mathbf{y}+\mathbf{r}$ ($\mathbb{S} = \mathbb{Y} + \mathbb{R}$ in the frequency domain.) is below:

$$\mathbf{h}_{\mathrm{est}} = \mathbf{F}^{\mathrm{H}}(\mathbb{S} \oslash \mathbb{X}) = \mathbf{F}^{\mathrm{H}}[(\mathbb{Y} \oslash \mathbb{X}) + (\mathbb{R} \oslash \mathbb{X})], \tag{4}$$

where the second argument in (4) represents the error due to observation noise. It shows that small absolute valued elements of $\mathbb{X}$ result in huge error. This huge error is why arbitrary signals are generally irrelevant for acoustic measurements.

Signal safeguarding sets lower limits $\mathbb{T} = T \cdot \mathbf{1}$ in the frequency domain and makes all elements have larger absolute values than T. This definition of $\mathbb{T}$ is the original proposal in the reference [1] to simplify and shorten the descriptions of the method due to limitted space in the letter article. We made $\mathbb{T}$ consists of frequency dependent threshould value $T[m]$ in the implementation of the interactive and real-time tool for acoustic condition measurement [4, 5].

Then, define a binary (0, 1) valued vector function $K_{\mathbb{T}}(\mathbb{X})$. The m-th elelemt of $K_{\mathbb{T}}(\mathbb{X})$ as follows.

$$(K_{\mathbb{T}}(\mathbb{X}))_m = \begin{cases} 1 & , \ T[m] > |X[m]| \\ 0 & , \ \text{otherwise} \end{cases} . \tag{5}$$

Then, the following equations define the procedure and provides an interpretation.

$$\begin{aligned} \mathbf{x}_{\mathrm{SG}} &= \mathbf{F}^{\mathrm{H}}[\mathbb{X} + K_{\mathbb{T}}(\mathbb{X}) \odot (\mathbb{T} - |\mathbb{X}|) \odot (\mathbb{X} \oslash |\mathbb{X}|)] & (6) \\ &= \mathbf{x} + \mathbf{F}^{\mathrm{H}}[K_{\mathbb{T}}(\mathbb{X}) \odot (\mathbb{T} - |\mathbb{X}|) \odot (\mathbb{X} \oslash |\mathbb{X}|)] & (7) \end{aligned}$$

where $|\mathbb{X}|$ represents a vector consisting of absolute value of each element of $\mathbb{X}$. The equation (7) provides an interpretation that the signal safequarding procedure adds (virtually stable and slight) noise to the original signal. This interpretation is **essential for retrospective application of signal safeguarding.**

2.4 Giant-FFT method for signal resampling

This section introduces giant-FFT SRC [6]. The following explanation is my interpretation.

Let $\mathbf{x}_{\mathrm{O}}$ represent the original discrete time signal. Let L represent the length of the input signal. Let f_{sO} and f_{sC} represent the sampling frequency of the original and the converted signals.

First step is to add trailing zeros to the orignal signal. This operation is to make the length of the converted signal $L_{f\mathrm{sC}}$ to an integer multiplel of a unit length L_{unit} defined by f_{sO} and f_{sC}.

$$L_{f\mathrm{sC}} = L_{\mathrm{unit}} \left\lceil \frac{L_{f\mathrm{sO}}}{L_{\mathrm{unit}}} \right\rceil, \quad \text{where} \quad L_{\mathrm{unit}} = \frac{f_{\mathrm{sO}} f_{\mathrm{sC}}}{\gcd(f_{\mathrm{sO}}, f_{\mathrm{sC}})^2}, \tag{8}$$

where $\lceil a \rceil$ provieds the minimum integer that is greater than or equal to a number a.

Let $\mathbf{x}_{\mathrm{Oext}}$ represent the extended orignal signal by trailing zero padding. Let $\mathbb{X}_{\mathrm{Oext}}$ represent the DFT of $\mathbf{x}_{\mathrm{Oext}}$. Then, f_{sO} and f_{sC} are in the extrapolated sequence of the FFT-bin's corresponding frequency of $\mathbb{X}_{\mathrm{Oext}}$.

The next step is to fill the FFT-bins of the converted signal. Let $f_{\mathrm{cO}}[m]$ represents the corresponding frequency of the m-th element of $\mathbb{X}_{\mathrm{Oext}}$. When $f_{\mathrm{sO}} > f_{\mathrm{sC}}$ (downsampling), copy the element of $\mathbb{X}_{\mathrm{Oext}}$ for all m that holds the condition ($f_{\mathrm{cO}}[m] \leq f_{\mathrm{sC}}/2$). Then, fill the rest of the elements by mirror image of complex conjugate of the copied elements. This procedure truncates the $\mathbb{X}_{\mathrm{Oext}}$ and yields the converted signal's DFT $\mathbb{X}_{\mathrm{C}}$.

When $f_{\mathrm{sO}} < f_{\mathrm{sC}}$ (upsampling), copy the element of $\mathbb{X}_{\mathrm{Oext}}$ for all m that holds the condition ($f_{\mathrm{cO}}[m] \leq f_{\mathrm{sC}}/2$). Then, fill the rest of the elements by mirror image of complex conjugate of the copied elements. This procedure adds filling zeros in $\mathbb{X}_{\mathrm{C}}$, the converted signal's DFT.

The inverse Fourier transformation of $\mathbb{X}_{\mathrm{C}}$ provides the sampling rete-converted signal $\mathbf{x}_{\mathrm{C}}$.

$$\mathbf{x}_{\mathrm{C}} = \mathbf{F}^{\mathrm{H}} \mathbb{X}_{\mathrm{C}}. \tag{9}$$

Figure 1 of the reference [6] shows this procedure. Figure 9 and Fig. 10 illustrate effectiveness of the giant-FFT SRC.

In downsampling, because it is a truncation, it is a good practice to smoothly attenuate high-frequency end tward zero. This note is in the reference [6] and we implemented in our tools.

2.5 MATLAB implementation: `samplingRateConvByDFTwin`

We implemented a giant-FFT SRC using MATLAB. By typing "help samplingRateConvByDFTwin" displays how to use this function. The following script shows an example how to use it. The original test signal is a periodic pulse train with an interval 1001 samples (the fundamental frequency (f_{o} instead of f_0 [9]) is 95.9041 Hz.).

```
fsIn = 96000;
fsOut = 44100;
x = zeros(fsIn*10,1);
x(1:1001:end) = 1;
tic
ym = resample(x,fsOut,fsIn);
toc
tic
[y,output] = samplingRateConvByDFTwin(x,fsIn,fsOut,2000);
toc
```

This script outputs the elapsed time of each resampler. They are 0.009803 s and 0.020825 s, respectively. The real-time ratios are 1021 and 480. We used MATLAB R2024b on MacBook Pro M1 Max with 64 GB memory.

The following script displays the results and save it as a PNG format image file.

```
figure;
fxx = (0:length(x)-1)'/length(x)*fsIn;
fxOut = (0:length(y)-1)'/length(y)*fsOut;
figure;
set(gcf,"Position",[680   602   560   276]);
w = nuttallwin12(length(ym));
maxP = max(20*log10(abs(fft(ym.*w))));
semilogx(fxOut,20*log10(abs(fft(ym.*w)))-maxP,"LineWidth",2);
grid on;
hold on;
semilogx(fxOut,20*log10(abs(fft(y.*w)))-maxP,"LineWidth",2);
set(gca,"FontSize",14,"LineWidth",2)
axis([10 fsOut -280 10])
xlabel("frequency (Hz)")
ylabel("level (dB)")
legend("MATLAB resample","giant-FFT SRC","Orientation", ...
    "horizontal","Location","south")
pause(1)
print -dpng -r200 srSample96k441k95FrctionalHz.png
```

Figure 2 compares the results of resampling function of MATLAB builtin `resample` and the giant-FFT SRC. The function `nuttallwin12` is a six-term cosine series window [10] designed for the reduced-aliasing glottal source model. It has very low sidelobe level and fast sidelobe decay (−114.24 dB and 54 dB/octave, respectively). The aliasing is only in the MATLAB `resample` result.

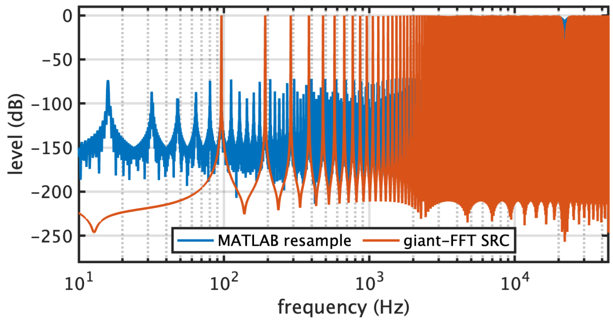


Figure 2: Sampling rate conversion test by MATLAB resample and giant-FFT SRC.

2.6 MATLAB implementation: `signalSafeguardwithGiantFFTSRC`

We implemented a signal safeguarding function with giant-FFT SRC. By typing "help signalSafeguardwithGiantFFTSRC" displays how to use this function. It is in the comment part of the m-function as follows.

```
function [y, output] ...
    = signalSafeguardwithGiantFFTSRC(xOriginal,fsIn,fsOut,thLevel,varargin)
% signal safeguarding with giant FFTs sampling-rate conversion
%  This sampling rate conversion does not introduce aliasing effects.
%
% Pattern 1
%  y = signalSafeguardwithGiantFFTSRC(xOriginal,fsIn,fsOut,thLevel)
%  Argument
%      xOriginal : input data vector with sampling frequency fsIn
%      fsIn      : sampling frequency of input signal (Hz)
%      fsOut     : sampling frequency of output signal (Hz)
%      thLevel   : threshould level from average spectrum (dB)
%  Output
%      y    : converted output with sampling frquency fsOut
%
% Pattern 2
%  [y, output] = signalSafeguardwithGiantFFTSRC(xOriginal,fsIn,fsOut,thLevel)
% Output
%      output  : structure with detailed debug data and shows spectrum figure
%
% Pattern 3
%  [y, output] = signalSafeguardwithGiantFFTSRC(xOriginal,fsIn,fsOut,thLevel,fHigh)
% Artument (additional)
%      fHigh  : high frequency end of safeguarding (Hz)
%               default Nyquist frequency - 3000
% Output
%      output  : structure with detailed debug data and shows spectrum figure
```

This implementation is redundant. Resampling with signal safeguarding is necessary only for evaluation by simulation using several different sampling frequencies.

The following script shows how to use it for acoustic measurement simulation.

3 MATLAB sample script for retrospective application of signal safeguarding

We added a sample MATLAB script for testing retrospective application of signal safeguarding.

Please try. The script is `retroSGSample.m`.

References

[1] H. Kawahara and K. Yatabe, "Safeguarding test signals for acoustic measurement using arbitrary sounds: Measuring impulse response by playing music," *Acoustical Science and Technology*, vol. 43, no. 3, pp. 209–212, 2022. [Online]. Available: https://dx.doi.org/10.1250/ast.43.209

[2] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "Rwc music database: Popular, classical and jazz music databases." in *Proceedings of the 3rd International Conference on Music Information Retrieval (ISMIR 2002)*, 2002, pp. 287–288.

[3] M. Goto, "RWC music database: (English version)," 2002, https://staff.aist.go.jp/m.goto/RWC-MDB/, (retrieved 15 June 2025).

[4] H. Kawahara, K. Yatabe, K.-I. Sakakibara, M. Mizumachi, and T. Kitamura, "Simultaneous measurement of multiple acoustic attributes using structured periodic test signals including music and other sound materials," in *Proc. APSIPA ASC*. IEEE, 31 Oct. 2023, pp. 173–180. [Online]. Available: \https://dx.doi.org/10.1109/APSIPAASC58517.2023.10317411

[5] H. Kawahara, K.-I. Sakakibara, M. Mizumachi, and K. Yatabe, "Proposal of protocols for speech materials acquisition and presentation assisted by tools based on structured test signals," in *Proc. the 27th Oriental COCOSDA*, 2024, pp. 1–6. [Online]. Available: \https://doi.org/10.1109/O-COCOSDA64382.2024.10800149

[6] V. Välimäki and S. Bilbao, "Giant FFTs for sample-rate conversion," *Journal of the Audio Engineering Society*, vol. 71, no. 3, pp. 88–99, 10 Mar. 2023. [Online]. Available: http://dx.doi.org/10.17743/jaes.2022.0061

[7] C. E. Leiserson, N. C. Thompson, J. S. Emer, B. C. Kuszmaul, B. W. Lampson, and et al., "There's plenty of room at the top: What will drive computer performance after moore's law?" *Science*, vol. 368, no. 6495, p. eaam9744, 2020. [Online]. Available: \https://dx.doi.org/10.1126/science.aam9744

[8] M. Frigo and S. Johnson, "FFTW: an adaptive software architecture for the FFT," in *Proc. ICASSP '98*, vol. 3, 1998, pp. 1381–1384. [Online]. Available: \http://dx.doi.org/10.1109/ICASSP.1998.681704

[9] I. R. Titze, R. J. Baken, K. W. Bozeman, S. Granqvist, N. Henrich, C. T. Herbst, D. M. Howard, E. J. Hunter, D. Kaelin, R. D. Kent, J. Kreiman, M. Kob, A. Löfqvist, S. McCoy, D. G. Miller, H. Noé, R. C. Scherer, J. R. Smith, B. H. Story, J. G. Švec, S. Ternström, and J. Wolfe, "Toward a consensus on symbolic notation of harmonics, resonances, and formants in vocalization," *J. Acoust. Soc. Am.*, vol. 137, no. 5, pp. 3005–3007, 2015. [Online]. Available: https://doi.org/10.1121/1.4919349

[10] H. Kawahara, K.-I. Sakakibara, M. Morise, H. Banno, T. Toda, and T. Irino, "A new cosine series antialiasing function and its application to aliasing-free glottal source models for speech and singing synthesis," in *Proc. Interspeech 2017*, Stocholm, Aug. 2017, pp. 1358–1362. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2017-15

[11] T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 5220–5224. [Online]. Available: http://dx.doi.org/10.1109/ICASSP.2017.7953152

[12] Open Speech and Language Resources, "SLR28 Room impulse response and noise database," 2017, https://www.openslr.org/28/, (retrieved 08 June 2025).

[13] isophonics, "QMUL Room impulse response data set," 2008, http://isophonics.net/content/room-impulse-response-data-set, (retrieved 08 June 2025).

[14] R. Stewart and M. Sandler, "Database of omnidirectional and B-format room impulse responses," in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2010, pp. 165–168. [Online]. Available: http://dx.doi.org/10.1109/ICASSP.2010.5496083

A Room impulse response database

We used several database for evaluating our methods. The following article [11] introduces a site [12] that has a collection of availabe room impulse response (RIR) databases. We selected QMU RIR database [13, 14] in this memo.

A.1 Queen Mary University Room Impulre Response database

QMU RIR database consists of RIR of three rooms, Great Hall, Octagon, and Classroom. In this note, we used RIR of the Octagon. The following is an excerpt in the database description [13].

> The Octagon is a Victorian building completed in 1888 and originally designed to be a library. It is currently used as a conference venue, but the walls are still lined with books with a wooden floor and plaster ceiling. As the name suggests, the room has eight walls each 7.5 m in length and a domed ceiling reaching 21 m over the floor, with an approximate volume of 9500 cubic m.

The measurement used Genelec 8250 power monitor loudspeaker for the sound source. The recording used two microhpones. Omindirectional recordings used an ominidiretional condenser microhpone DPA 4006. Acoustic field recordings used Soundfield SPS422B for recording in a B-format. The sweep tone based procesure derived these RIR data.

The following is the download files. They are zip files.

- Documentation (photo of room, diagram of layout) and sample IR (1.8 MB)
- Omnidirectional (60.3 MB)
- W of B-format (64.3 MB)
- X of B-format (64.5 MB)
- Y of B-format (63.4 MB)
- Z of B-format (62.9 MB)

Figure 3 shows the photo and recording setting of the source and the acquired locations. Locations are separated 5 m each. In total 169 recordings are made and saved using WAV format in 96000 Hz and 24 bit sampling. The diagram shows their file name and corrsponding locations.

A.1.1 RIR characteristics

We selected RIR files of the Octagon. The center locations of the front row and the back (far end) row. Their file names are `x06y00.wav` and `x06y12.wav`. Please refer Fig. 3 to find its actual positions in the room.

Figure 4 shows impulse response examples. In the figure, (a) and (c) show the decay of the (absolute) value of the impulse response using logarithmic (dB) vertical axis. In the figure, (b), and (d) show the frequency response representation of the impulse responses. The recording locations are center in the front row (a) and (b) and the center in the last (far end) row (c) and (d).

The decay rate (a) and (c) are virtually identical meaning the reverberation time does not change inside the same room. The higher frequency response decays more steeply in the last row. It is the result of low direct sound level.

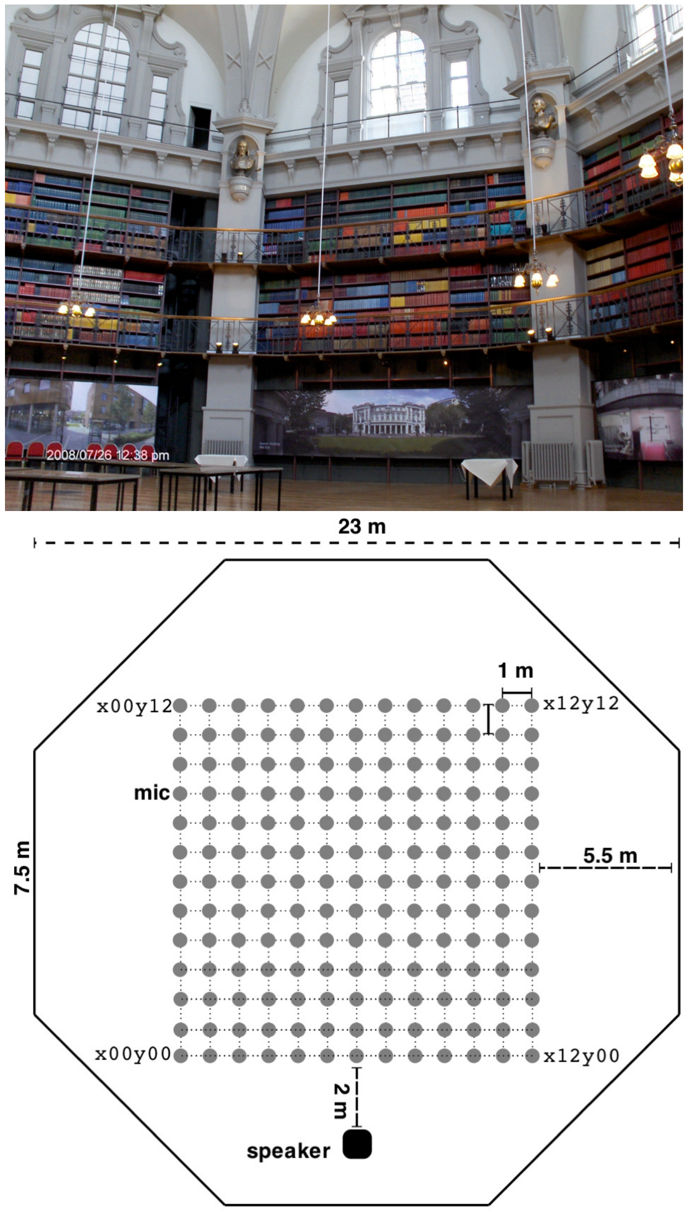
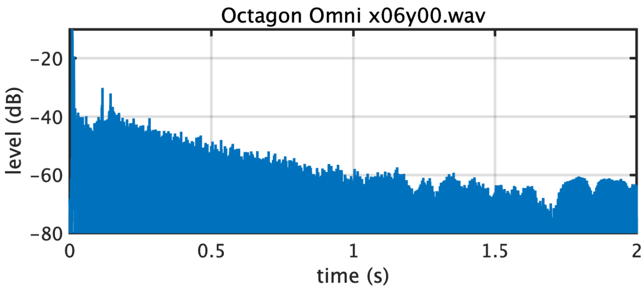
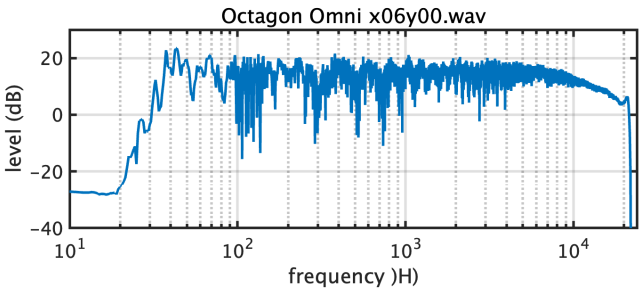


Figure 3: Octagon room (old libraty hall) and source and receiver positions. The photo and diagram are downloaded from QUM IRI site [13].

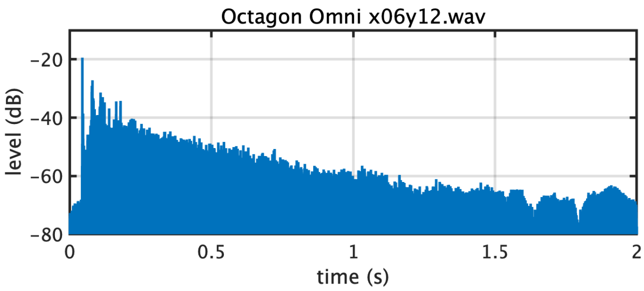
The QMU RIR database provides RIR retriev functions. We wrote a MATLAB script to use the functions to get desired RIR file. We used the giant-FFT SRC for converting RIR data from 96000 Hz (database) to 44100 Hz (simulations and other recording materials).



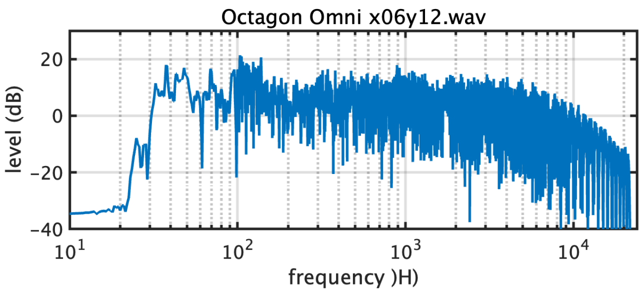
(a)



(b)



(c)



(d)

Figure 4: Power decay and frequency responses of Octagon

```
roomList = {'greathall','octagon','classroom'};
roomID = 2;
roomName = roomList{roomID};
rowList = {'F','B'};% Front and Back
rowID = 2;
rowName = rowList{rowID};
simNoiseList = {'white','pink','red'};
noiseId = 3;
simNoise = simNoiseList{noiseId};

l_data = length(xSg);
xOrg = outPrm.x;
xSgConst = outPrm.xSgConst;
xWhite = outPrm.whiteDetNoise;

baseDir =  ['/Volumes/HD-CD-A/RIRdatabase/QMLirDB/' roomName 'Omni/Omni/'];
switch rowName
    case rowList{1}
        switch roomName
            case roomList{1}
                fileRIR = 'x06y00.wav';
            case roomList{2}
                fileRIR = 'x06y00.wav';
            case roomList{3}
                fileRIR = '30x00y.wav';
        end
    case rowList{2}
        switch roomName
            case roomList{1}
                fileRIR = 'x06y12.wav';
            case roomList{2}
                fileRIR = 'x06y12.wav';
            case roomList{3}
                fileRIR = '30x45y.wav';
        end
end
[hIn,fsRIR] = audioread([baseDir fileRIR]);
h = samplingRateConvByDFTwin(hIn,fsRIR,fsOut);
hExt = [h;zeros(l_data-length(h),1)];
hExtF = fft(hExt);
```

Figure 5: A MATLAB interface script to select a RIR file from QMU RIR database. `testScrptForEA2025July.m`