Audio Engineering Society

Convention Express Paper 133

Presented at the 155th Convention
2023 October 25–27, New York, USA

Implementation of Simultaneous Deconvolution on a Real-time Smartphone App

Ashish Rawat, Sunil Bharitkar, Allan Devantier, Matthew McDuffee, and Ritesh Banka

Samsung Research America, DMS Audio, Valencia CA, USA

Correspondence should be addressed to Ashish Rawat (ashish.rawat@samsung.com)

ABSTRACT

In-room speaker system equalization was traditionally implemented by exciting one speaker at a time. With a higher number of speakers, restrictions of measurement microphone setup, the annoyance factor due to traditional stimuli, and background noises, the process of measuring the impulse response of a multi-channel system in real-time can be cumbersome. With FFT computation restrictions on a smartphone DSP, the accuracy and resolution of the impulse responses are compromised. This paper addresses all of these concerns with a novel approach to implementing the Simultaneous Deconvolution of a multichannel speaker system. It uses a set of circularly shifted Sine-Sweep stimuli to excite the speakers and calculate the impulse responses in real-time on a smartphone app over a cloud-based architecture. An independent recording and playback system, along with manual delays or system delays due to Bluetooth, Wi-Fi, or cloud-based communication, pose further challenges to the accuracy of our measurements. To surmount these complications, we discuss a time-alignment method that uses bin-wise matched filtering of spectrograms, followed by a statistical analysis of its results.

1 Introduction

Measurement of an impulse response of a speaker-room acoustical system was traditionally implemented using deterministic log sweep, pseudo-random MLS (Maximum Length Sequence), IRS (Inverse Repeated Sequence), time-varying frequency signal: Time-Stretched Pulses and stochastic white noise, pink noise and multitone. All the above stimuli were traditionally used for measuring one speaker at a time. This paper revolves around implementation of circularly shifted N-Channel Log Sine-Sweep [1] [2]. Along with being complex and time consuming the traditional approaches are erring fallible and economically expensive. An implementation which can be carried out by a layman in the real-world setup needs to be quick, reliable and accessible. A smartphone app which implements measurement of an N-channel speaker system requires: a high-resolution FFT computation capability, accurate time synchronization of smartphone and speaker system and accurate calculation of delays and levels for the main channel along-side a well equalized cross-over region for the sub-woofer and the main channels.

We also highlight the most suitable stimuli for our application in this section along with other commonly used stimuli. MLS is a pseudo-random signal which is stochastically similar to a pure white noise and is derived using a simple register shift which uses circular cross-correlation to generate impulse response of the acoustical system. Length of MLS sequence affects

the time domain properties as well as frequency resolution of the signal. This implies that longer MLS sequences yields better impulse responses and more accurate measurements. Time-Aliasing error is significantly observed when length L of one period is shorter than the length of impulse response. Any noise (white or impulsive) will lead to a uniform distortion of the impulse response not constrained in a limited region along time/frequency axis. [2]

In order to overcome the limitation of MLS and IRS(discussed in[2]), Logarithmic Sine Sweeps can be used to calculate impulse response. Considering the specific requirements of developing an App for measurement of speakers, where speed of measurement, noise immunity(refer to [3]) and linearity throughout the frequency range is of importance, Log Sine Sweep is a better choice. For faster implementation of the impulse response calculation, we use the technique described by Bharitkar in [1] [4]. Next section speaks in detail about the stimuli, algorithm and the implementation.

$$x(t) = \sin\left(\frac{\omega_1 T}{\log\left(\frac{\omega_2}{\omega_1}\right)}\right)\left(e^{\frac{t}{T}\log\left(\frac{\omega_2}{\omega_1}\right)} - 1\right) \quad (1)$$

where: $x(t)$ represents single channel sweep which will be explored in the next section, ω_1 and ω_2 are angular frequencies, T is the duration, t is the time variable.

2 Simultaneous Deconvolution

Since in-room speaker system measurements, with Log sine sweeps are time efficient, less prone to noise and easy to implement, a Simultaneous deconvolution based smartphone application in this paper is implemented using the techniques developed by Bharitkar. [1] [4].

2.1 Algorithm

Simultaneous Deconvolution Algorithm calculating impulse response estimate uses the log-sweep autocorrelation inverse spectrum and the cross correlation shown in Equation (2) [1]

$$h_j(n) = w_j(n) \circledast \rho_{(x_j(n),y(n))} \quad (2)$$

where,

$w_j(n) = \mathscr{F}^{-1}\left\{\frac{1}{S_{x_j,x_j}}\right\}$

$S_{(x_j,x_j)} = \mathscr{F}\{\rho_{(x_j,x_j)}\}$

Input signal: $x(n) = x(t)\big|_{t=nT_s}$ where $x(t)$ is from Equation(1)

$$x_i(n) = \begin{pmatrix} x(<n-(i-1)M>p) \\ x(<n-(i-1)M-1>p) \\ \vdots \\ x(<n-(i-1)M-P+1>p) \end{pmatrix}$$

$(i = 2, \ldots, 12)$

Calculation for a single channel sweep is illustrated in Fig.1. Note that for an N channel speaker system, j^{th} channel input is one of the log-sweeps which is circularly shifted by <M>P.

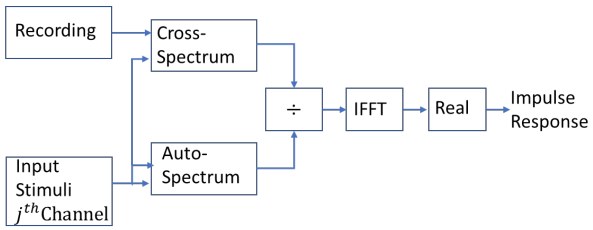

Fig. 1: Impulse Response using a Cross-Spectrum based Deconvolution.

2.2 Implementation of the Simultaneous Deconvolution(SD) algorithm

Accurate impulse response calculations from a deconvolution operation calls for high resolution FFT. An on-device implementation of Simultaneous Deconvolution Algorithm as Smartphone App will be restricted with the length of FFT which can be implemented on the processor. The App often runs into overruns attempting to implement high resolution FFT. In order to solve the problems of hardware overruns, we can delegate the processing to a cloud-based processor.

An App was implemented using the MATLAB Mobile [5]. The architecture to implement the Simultaneous Deconvolution App was built using the MATLAB Mobile's connect to Mathworks Cloud feature. This

enables us to implement FFT of length in the magnitude of 2^{19} (required for our app) and above, which is sufficient to deconvolve sweeps of about the same length.

Fig.2 illustrates the architecture of the measurement set up used. We have evaluated measurements taken with a Smartphone with its internal microphone and with an external measurement microphone(i437L and i458C). The recording from the phone is sent to the cloud where the algorithm is implemented. The proposed architecture suggests the use of a cloud based control to synchronize the start and stop time for the recording and the playback from the speaker system. We will also discuss the caveats of having an asynchronous measurement setup and a spectrogram based synchronization solution in the next section.

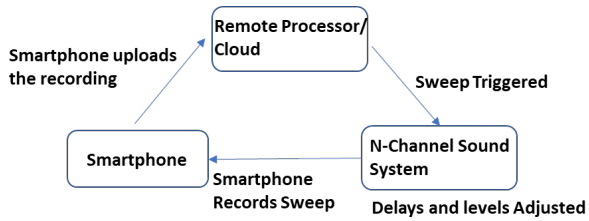

Fig. 2: Architecture of the measurement setup

2.2.1 App Interface and readings

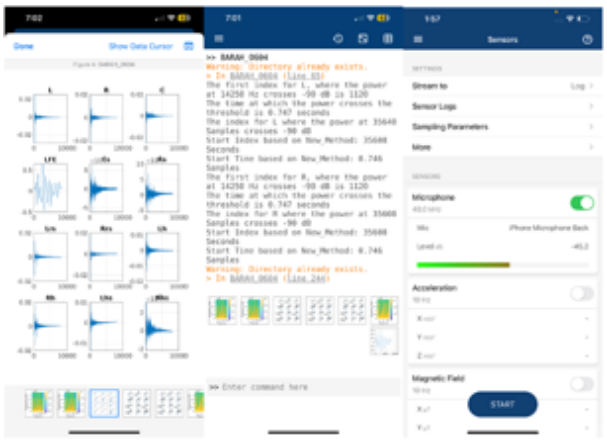

Fig. 3: Matlab Mobile GUI demonstrating impulse responses,results from the algorithm on command window

Fig.3 shows the interface of the smartphone based measurement application created on the Matlab Mobile platform [5]. The impulse responses generated simultaneously for the 12 channels are displayed for visual analysis. We also access some sanity checks on the command window of the App, for the recording taken. Matlab mobile also gives us control to select if we wish to use the external microphone or one of the multiple in-built microphones.

3 Start and End Time Alignment

In a real-time smartphone application for measuring a speaker system using a N channel deconvolution ($N \geq 1$) operation, we need to align the recorded signal with the input stimuli. This is a major challenge in measurement systems where the playback and recording systems are independent and non-synchronized. If the recorded signal and the actual stimuli are not aligned, we see skewed and noisy impulse responses, hence inaccurate calculations of delays and levels. Time alignment using matched-filtering, cross-correlation or thresholding-based methods are heavily influenced by room-reflections, background noise and buffer time between input and recorded signals. This leads to distorted impulses and unreliable transfer function centric calculations. Farina's work in [3] speaks about skewed impulse responses due to mismatched clocks which is closely related, but not the exact topic addressed here.

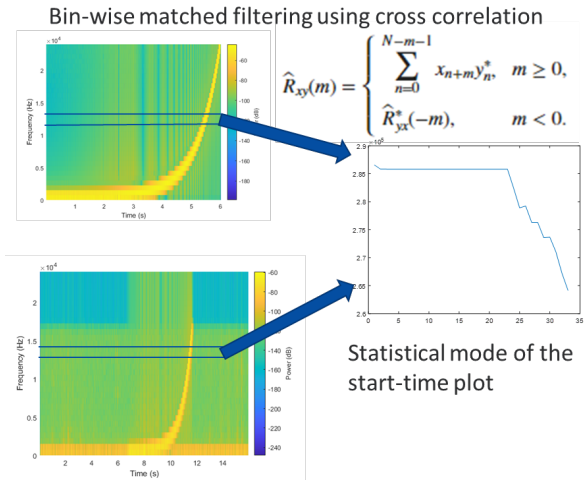

Fig. 4: Time Alignment Algorithm implementation using a single channel sweep

A high time resolution spectrogram (window : 64 samples and overlap : 50%) based time alignment is robust and reliable in real-time applications. In our application we use a frequency bin-wise macthed filtering of the two spectrograms shown in Fig. 5 and Fig. 6. These figures represent recording spectrogram and the input stimuli spectrogram respectively. Note that this technique is independent of the number of channels being deconvolved and hence we can use this technique to time align ($N \geq 1$) channels at a time. Fig. 4 shows graphically how this time alignment technique can be implemented on a single channel exponential sine sweep. Statistical analysis of the start-times derived for each frequency bin can yield us the actual start time of the stimuli captured in our recording. We use a statistical mode in this paper. The following equations gives us clarity on the algorithm implemented.

Cross-correlation of two frequency bins S_M and S_{ideal}:

$$\text{corr}(S_M, S_{\text{ideal}})[m] = \sum_{n=0}^{N-1} S_M[n] \cdot S^*_{\text{ideal}}[n-m] \quad (3)$$

Where, $S_M[n]$ and $S_{\text{ideal}}[n]$ are the nth elements of measured and ideal stimuli spectrogram bins. This cross correlation is executed over length N with shift m.

$$\max_{\text{corr}} = max(\text{corr}(S_M, S_{\text{ideal}})) \quad (4)$$

$$\text{note that : } \text{lag}(S_{\text{peak\_index}}) = index(max_{\text{corr}})$$

$$\text{arrivalTime} = \text{lag}(S_{\text{peak\_index}}) \quad (5)$$

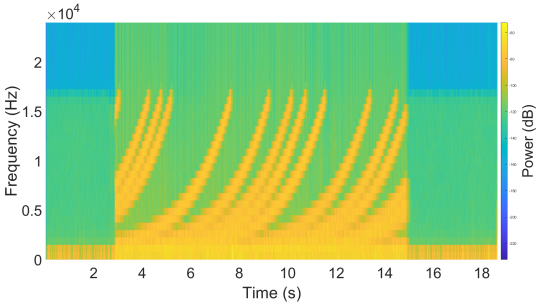

Fig. 5: High time resolution spectrogram of the recording with the phone

Fig. 7 shows time domain plot of the recording taken from our App. This recording was taken in a noisy environment with speech and impulsive noises (non-stationary). This was done in order to replicate a real-world scenario in which an end user of this smartphone app can be located in. Red color regions represent pre-stimuli and post stimuli buffer region. The blue color region represents the recording of stimuli which should be used for deconvolution with the original stimuli. If we use an amplitude threshold based approach for start time detection, we see a false impulsive noise detected as the start time. We detect the actual stimuli using the proposed technique with an accuracy of 64 sample(frequency resolution of our spectrogram).

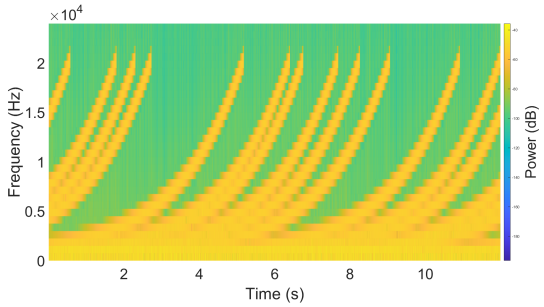

Fig. 6: High time resolution spectrogram of the input stimuli

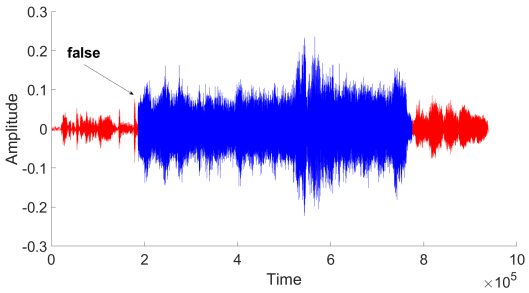

Fig. 7: Time Domain of a noisy recording of 12 CH circularly shifted sweep

Fig. 8 shows the spectrogram of the noisy recording shown in Fig.7. This spectrogram is used for calculating the start and stop time of the noisy recording accurately. A clean and accurate impulse response is calculated using the deconvolution operation proposed in section 2.1 and is shown in Fig. 10. If an alternative start time detection, based on threshold in time or frequency domain is used, we detect start time erroneously

and hence obtain a noisy, inaccurate impulse response shown in Fig. 9.

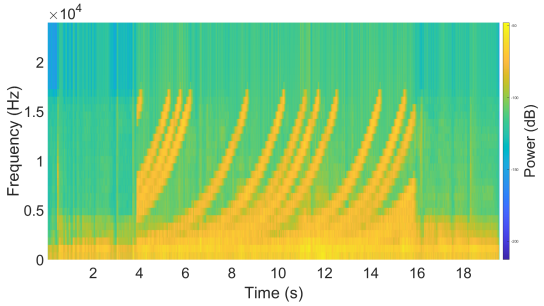

Fig. 8: High time resolution spectrogram of the noisy recording (non-stationary)

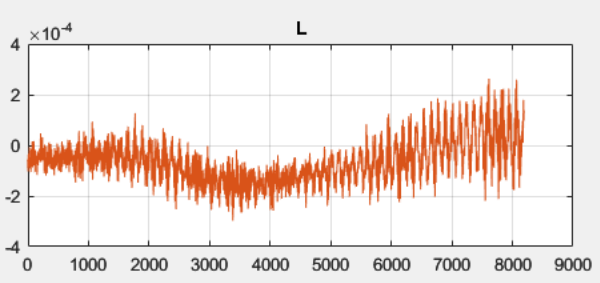

Fig. 9: Impulse response plotted on the Smartphone App using the false start-time(calculated using the threshold based approach) under non-stationary noise

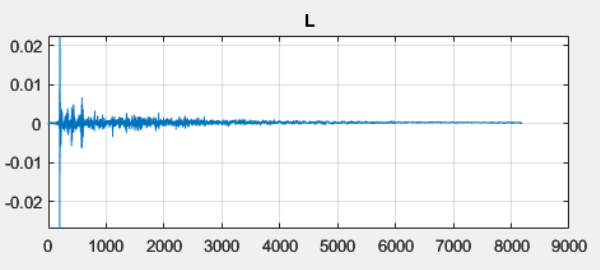

Fig. 10: Impulse response plotted on the Smartphone App using the accurate start-time(calculated using the spectrogram based approach) under non-stationary noise

The start time detected by our method vs erroneously by the traditional method varies about 0.8sec and the impact caused by the false start time is drastically observed in Fig. 9.

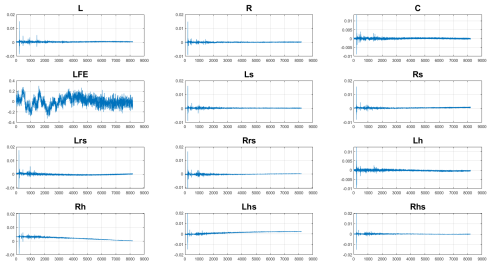

Fig. 11: Impulse Response using the App with 0dB SNR(stationary noise)

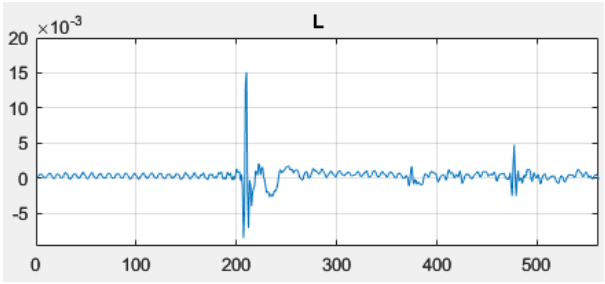

Fig. 12: Impulse Response of main channel using the App with 0dB SNR(stationary noise)

The start and end time alignment using spectrogram was also tested with Vacuum Cleaner noise(stationary) at 0dB SNR. In Fig. 11, we can see that the impulse response generated for the speakers were clean and useful for delay and level correction except the subwoofer impulse response which gets corrupted due to the low frequency content of the noise. This can be seen in the spectrogram of the noisy recording in Fig. 13. A closer look into the clean impulse can be seen in Fig. 12.

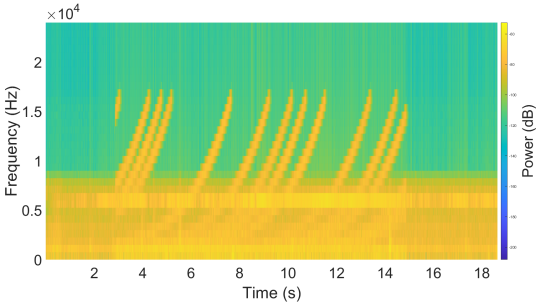

Fig. 13: High time resolution spectrogram of the 0dB SNR noisy recording (stationary)

4 Calculations and Results

4.1 Delay and Error Calculation

Fig. 14 shows the impulse responses of the 12 channels measured using the external microphone. Channel 4 shows the impulse response of the sub-woofer labeled as LFE. As expected we observe only the low frequency component present its impulse response.

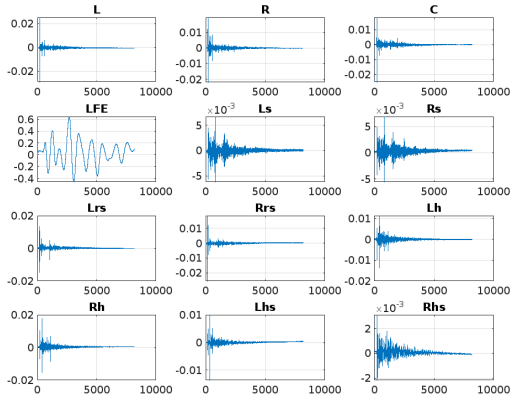

Fig. 14: 12 Channel impulse responses generated using Simultaneous Deconvolution for Soundbar

Relative delays for each channel is calculated by prominence of peak. The function for prominence is implemented using 'findpeaks'[5]

$$P = H - \min(L, R) \tag{6}$$

In Equation (6), P is the prominence of the peak,H is the height or amplitude of the peak, L and R are the left and right valley.

Now we find the first peak of the impulse response which is above a certain threshold of the prominence.

$$peak\_index = \arg\min_{i} \left(P(|IR[i]|) > P_t\right) \tag{7}$$

In Equation (7), P(x[i]) is prominence of peak at i^{th} sample in impulse response(IR), and P_t is the threshold value for prominence to detect the valid peak.

In order to calculate relative channel delay we use Equation (8)

$$MeasuredDelay_{i,j} = TOA_i - TOA_j \tag{8}$$

In Equation (8), TOA_i is the time of arrival for the i^{th} channel, calculated from the known sampling frequency and the peak_index in Equation (7).

$$RelativeDist_{i,j} = ActualDist_i - ActualDist_j \tag{9}$$

In Equation (9), $RelativeDist_{i,j}$ is the actual relative distance between speakers for channel i and j respectively. This distance is calculated using the difference between the measured distance of the microphone from the speakers for channel i and j. This measurement in our experimental setup was done using a laser rangefinder.

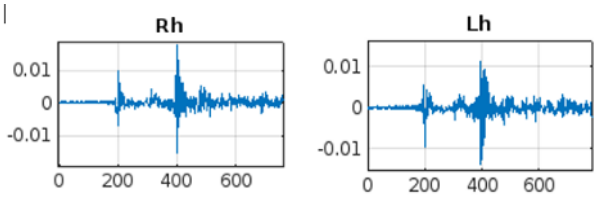

Fig. 15: A closer look of the impulse responses of the height channels of the Soundbar for ceiling reflection analysis

Direct flight delays for Height channels can be calculated with Equation (6) to (8) on the first impulse response and the first reflection delay with (6) to (8) on the second impulse response observed in Fig. 15. Since the height channels on the soundbars are directed

towards the ceiling, we see that the first reflection impulse response is more prominent than the direct flight impulse response.

$$Error_{i,j} = Delay(Relative_{i,j}) - Delay(Measured_{i,j}) \tag{10}$$

Equation (10) shows that $Error_{i,j}$ is the difference between actual relative delay: $Delay(Relative_{i,j})$ and the delay measured using Equation (8): $Delay(Measured_{i,j})$

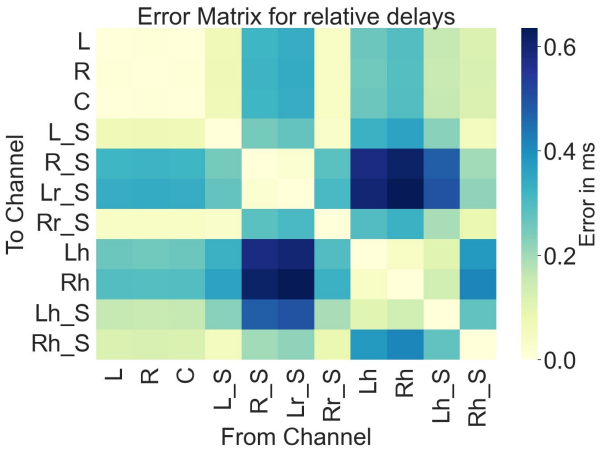

Fig. 16: Relative Channel Delay Error

Error Matrix for relative channel delays are displayed using a Heatmap in Fig. 16. shows that with the proposed algorithm and its implementation we have an relative channel delay error under 1 ms for all the channels.

Delay for sub-woofer/LFE Channel is implemented based on maximizing the summation of sub-woofer and main channel's frequency response. This maximization is evaluated by delaying the impulse response of the sub-woofer and the main channel iteratively and then minimizing the standard deviation of the frequency response over a cross-over region.

For this set of frequency responses, we find the response with the least standard which is plotted in Fig. 17

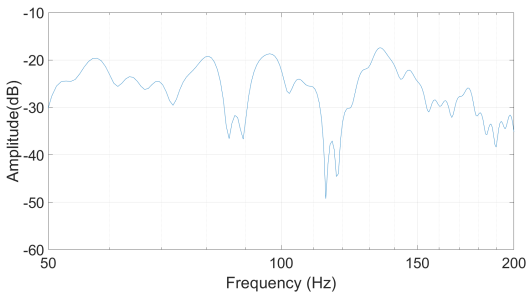

Fig. 17: Crossover of main and sub-woofer channel with least Standard deviation in the frequency of concern

Once we find that N sample (t ms) delay of either the main or sub-woofer yields the least standard deviation of frequency response in the cross-over region, we can conclude the required delay for our system and time align all the channels suited best for the listner's position.

4.2 Level Calculation

Sound Power level can be calculated using the impulse responses derived from the simultaneous deconvolution app. Level equalisation for the speakers can be conducted for the primary listener's position based on this.

$$\begin{aligned} H &= \mathscr{F}(Impulse\_Response_i) \\ X &= \mathscr{F}(Pink\_Noise) \end{aligned} \tag{11}$$

In Equation (11), H and X are the FFTs of the Impulse response(Derived with SD App for the i^{th} channel) and a Pink noise(with a reference SPL) respectively. The FFT length used is : length($Impulse\_Response_i$) + length(Pink_Noise) - 1.

$$y_i = (Real(\mathscr{F}^{-1}(H .* X'))); \tag{12}$$

In Equation (12), y_i is the time domain convolution of the pink noise(measured) and the impulse response(calculated using the app).

$$spl\_ch_i = 20 * log10(rms(y_i)/ref) \tag{13}$$

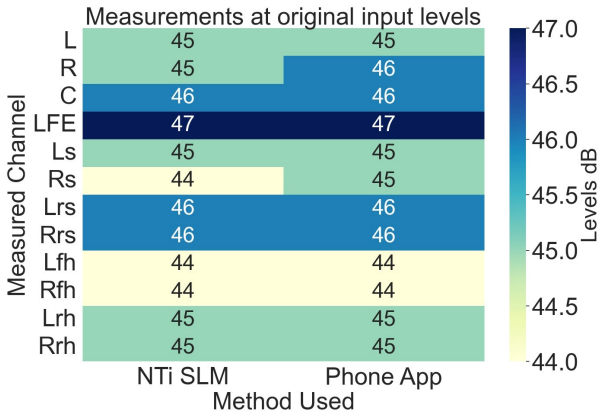

Fig. 18: Level Measurement Heatmap for accuracy of measured Levels with phone vs NTi Sound Level Meter

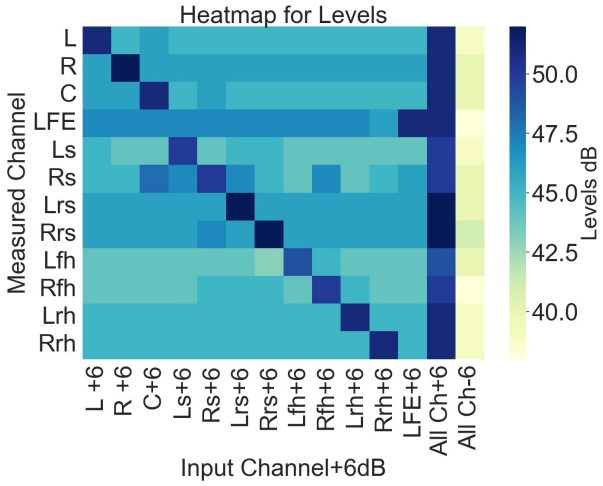

Fig. 19: Level Measurement Heatmap for accuracy of measured Levels using Phone App

Results obtained for level calculation of the soundbar are shown in the Heatmaps above. In Fig. (18), we have the comparison of the levels calculated using the app vs with measurements done using an NTi XL2 SLM(sound level meter). The Heatmap in Fig.(19) shows Levels calculated by increasing the input to each channel by +6dB (x-axis) and we observe it reflected in the measured data (y-axis).

5 Conclusions and Future Direction

Real-time implementation of a speaker measurement smartphone app (for levels and delay) is a challenging process. The challenge of high resolution FFT on device is solved with an alternative approach to deploy the calculations over a cloud based processor. Along with other pros discussed in this paper, circularly shifted log sine sweeps are time efficient and hence suitable for our purpose. Another major challenge of having non-synchronized playback and recording systems is addressed using the statistical features from a high time-resolution spectrogram. This approach can be extensively explored in the future with deconvolution using various stimuli showing stimulus independence. A low computation based machine learning approach for noise cancellation will be explored to make this measurement technique more robust and further accurate in noisy conditions (especially sub-woofer response).

6 Acknowledgement

Samsung Electronics and Samsung Research America supported this work. The authors would like to thank Samsung's US Audio Lab staff, who helped with all aspects of this research, offered insightful suggestions, and contributed to this work.

References

[1] Bharitkar, S., "Deconvolution of Room Impulse Responses from Simultaneous Excitation of Loudspeakers," *AES 151st Convention*, 2021.

[2] Stan, Guy-Bart and Embrechts, Jean Jacques and Archambeau, Dominique, "Comparison of different impulse response measurement techniques," *Journal of the Audio Engineering Society*, 50, pp. 249–262, 2002.

[3] Farina, A., "Advancements in Impulse Response Measurements by Sine Sweeps," *AES 122nd Convention*, 2007.

[4] Bharitkar, S., "Bayesian Optimization of Simultaneous Deconvolution of Room Impulse Responses," *IEEE Workshop on Signal Processing Systems(SiPS)*, 2022.

[5] The MathWorks Inc., "MATLAB version: 9.13.0 (R2022b)," 2022.