MethodsX 8 (2021) 101288

Contents lists available at ScienceDirect

MethodsX

journal homepage: www.elsevier.com/locate/mex

Method Article

# Automation of binaural headphone audio calibration on an artificial head

Kenneth Ooi, Yonggang Xie, Bhan Lam*, Woon-Seng Gan

*School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore*

A B S T R A C T

In studies with auralisation of audio stimuli over headphones, accurate presentation of headphone audio is critical for replicability and ecological validity. Audio stimuli levels are usually calibrated by placing studio quality headphones on an artificial head and torso simulator. Manual adjustment of audio tracks becomes laborious when the number of stimuli is large, especially for applications with large datasets. To increase reliability and productivity, we devised a stimulus-agnostic, automated calibration procedure for headphone audio via an artificial head and torso simulator, with a LabVIEW implementation available at doi:10.21979/N9/0KYIAU.

- The procedure uses a National Instruments NI-9234 data acquisition module and works with any ITU–T P.58:2013 and ANSI/ASA S 3.36:2012 compliant artificial head measurement systems.
- The procedure works by an adjustment to a generic guess, followed by a modified binary search, wherein the audio stimuli are calibrated to within a user-specified tolerance level.
- Each stimulus in a validation run to calibrate 250 stimuli to 65.0 ± 0.5 dB was played back an average of 2.22 ± 0.92 times before successful calibration, thus demonstrating the robustness and efficiency of our proposed method.



A R T I C L E I N F O

*Method name:* Automation of binaural headphone audio calibration on an artificial head
*Keywords:* Soundscape, Spatial audio, Auralisation
*Article history:* Received 4 November 2020; Accepted 19 February 2021; Available online 26 February 2021

* Corresponding author.
*E-mail address:* blam002@e.ntu.edu.sg (B. Lam).

https://doi.org/10.1016/j.mex.2021.101288

**Specifications Table**

| | |
|---|---|
| Subject Area | Physics and Astronomy |
| More specific subject area | Acoustics |
| Method name | Automation of binaural headphone audio calibration on an artificial head |
| Name and reference of original method | NIL |
| Resource availability | https://doi.org/10.21979/N9/0KYIAU |

## 1. Introduction

Calibrating audio tracks to fixed or user-specified $L_{eq}$ (equivalent continuous sound pressure level) values when played back over headphones is a perennial concern for many studies in psychoacoustics and soundscapes. For example, laboratory studies may seek to fix a track at a particular $L_{eq}$ in order to remove the $L_{eq}$ as a possible confounding variable while investigating the effects of a different variable [1,2], or may require audio tracks calibrated to different $L_{eq}$ values in order to study the effect of $L_{eq}$ on a separate dependent variable [3]. It has been found that the $L_{eq}$ that a given audio stimulus is presented over headphones affects its subjective perception [3], and that reproduction system quality affects the ecological validity of the results of laboratory experiments [4]. Hence, accurate and reliable calibrations need to be performed to ensure the consistency and replicability of experiments that depend on human judgement and perceptions of sound.

However, due to nonlinearity and noise in many playback systems or environments, there is no closed form formula that allows one to calculate the exact gain required to calibrate audio tracks (represented as samples in the range [-1,1]) to desired $L_{eq}$ levels for all playback systems. Moreover, due to potentially different hardware components used in different playback systems and the different fit in different headphones, a gain that works for one system may not be directly transferable to another. Hence, empirical measurements are frequently required for accurate calibration, which necessitates the tuning of gains for each stimulus by hand. This can increase manpower requirements, and is time-consuming if many stimuli need to be calibrated for a large dataset, such as one used in a comprehensive experiment or one used to train a deep neural network. These requirements for calibration tasks could be reduced with an efficient automated calibration system and procedure.

Therefore, we describe our method to set up and operate such a system to calibrate audio tracks to user-specified $L_{eq}$ values in this article.

## 2. Hardware setup

The hardware required for the calibration of binaural headphone playback is dependent on the intent of the calibration procedure. For example, to play back recorded acoustic scenes for soundscape investigations, it is recommended to adhere to the hardware recommendations for binaural measurements in Annex D of ISO 12913-2 [5], as summarized in Table 1. Although ISO 12913-2 does not dictate the type of headphones to be used in headphone-based listening tests, guidelines

**Table 1**
Recommended hardware specifications for soundscape investigations.

| Type | Recommended | Used |
|---|---|---|
| Headphones | Circumaural reference monitor headphones [6] | Beyerdynamic Custom One Pro |
| Artificial Head and Torso Simulator^ | Compliant with<br>• ITU-T P.58:2013, 5.2<br>• ANSI/ASA S3.36:2012, Table 1 | GRAS 45BB KEMAR Head and Torso |
| Soundcard | Any high-quality soundcard | Creative Sound Blaster E5 |
| Analog-to-digital converter (ADC) | Any ADC compliant with the head and torso simulator, which can interface with LabVIEW<br>• Sampling rate: 44.1 kHz minimum<br>• Resolution: 24 bits minimum | • NI 9171<br>• NI 9234 |
| Acoustic environment | Compliant with ITU-R BS.1116–3, 8.2.1 [6] | See Fig. 2. |

^Based on Annex D of ISO 12913–2 [5].

**Fig. 1.** Interior of custom-designed anechoic box with calibration system setup.

in ITU-R BS.1116-3 could be adopted [6]. Finally, the acoustic environment in which the calibration is conducted is also recommended to at least meet the background noise requirement of ITU-R BS.1116-3.

In our setup, the audio track was presented through a pair of circumaural headphones (Custom One Pro, Beyerdynamic GmbH & Co. KG, Germany) driven by an external USB soundcard (Sound Blaster E5, Creative Technology Ltd., Singapore). The sound slider of the Custom One Pro-headphones was set to position 2 to achieve the flattest frequency response advised by the manufacturer. To ensure a sufficiently quiet calibration environment, the headphones were placed over the ears of an artificial head and torso simulator (45BB KEMAR Head and Torso, G.R.A.S. Sound & Vibration A/S, Holte, Denmark) located in a custom-designed, low-cost anechoic box (Fig. 1). The 200 cm by 120 cm by 180 cm ($L \times W \times H$) anechoic box is constructed from 2.5 cm thick medium density fiberboards and its interior was lined with sound baffles. Silicone sealant was used on gaps between the fiberboards to make it as airtight as possible. Analog signals from the microphones in the KEMAR 45BB were pre-amplified (12AA, G.R.A.S. Sound & Vibration A/S, Holte, Denmark) before analog-to-digital conversion (NI 9234 with NI 9171, National Instruments, Austin, TX, USA). The connection diagram of the hardware setup is illustrated in Fig. 2.

The custom anechoic box fell just outside the maximum background noise requirement in ITU-R BS.1116-3, wherein the 1/1 octave band background noise of NR 17.4 is above the recommended limit of NR 15, as shown in Fig. 3. The octave band plots of the noise floor were measured with a

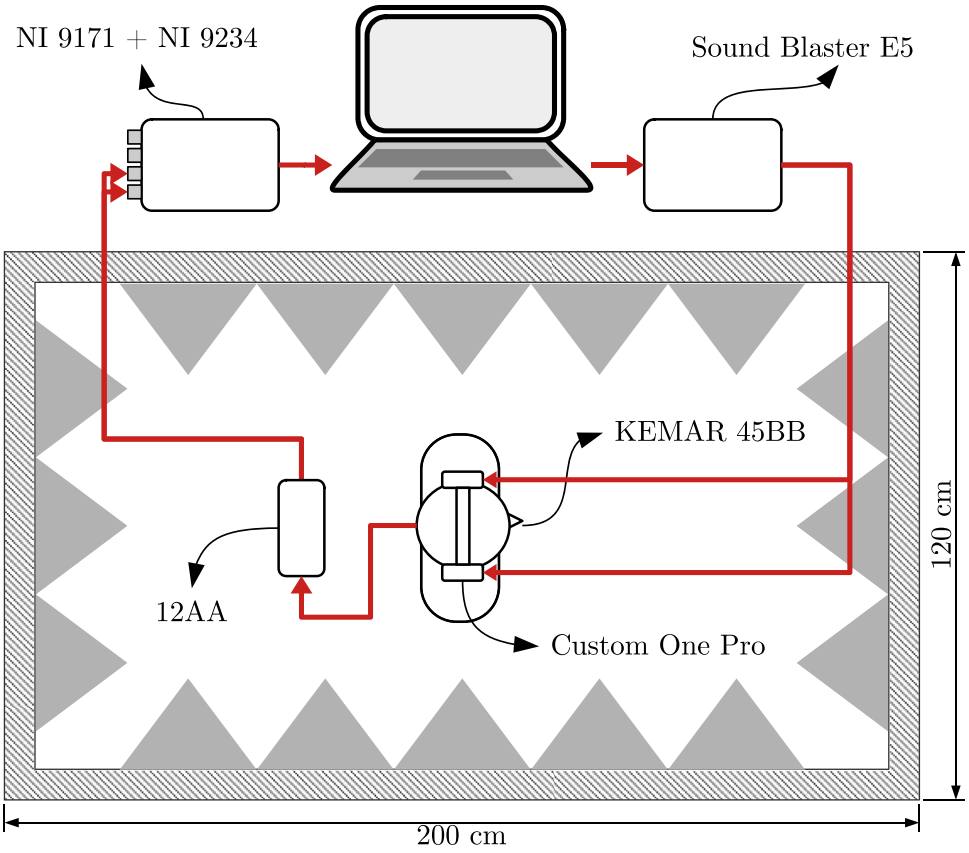


**Fig. 2.** Hardware connection diagram of the calibration system, whereby physical connections are indicated in red. The KEMAR 45BB was placed in a custom 200 cm by 120 cm by 180 cm ($L \times W \times H$) anechoic box (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.).

calibrated sound level meter (B&K 2245, HOTTINGER BRÜEL & KJÆR A/S, Nærum, Denmark) placed at the ear height of the KEMAR and were averaged over five 30 s long measurements. It is worth noting that the noise floor of the B&K 2245 is 21 dB (Z-weighted), indicating that the noise floor of the anechoic box is likely below the noise floor of the measurement system.

## 3. LabVIEW software architecture

To drive the hardware setup described in Section 2, we have also created a LabVIEW Virtual Instrument (VI) that performs the entire end-to-end calibration process for an arbitrary number of stimuli represented as .wav files to arbitrarily specified $L_{eq}$ values. The time average is taken across the entire length of each stimulus, so for a stimulus that is t minutes long, the calibration would be done to an $L_{eq,\ t-min}$ value, although we drop the length of time in this article for brevity.

The overall flow of the VI is shown in Fig. 4, and the subsequent sections will elaborate on the individual components of the VI in detail. Given a desired $L_{eq}$ that we want to calibrate all stimuli to (say $D$ dB), the VI will automatically do the following for each .wav file:

1. Play the stimulus $\boldsymbol{x}$ over the circumaural headphones that are placed on the head and torso simulator with a user-specified gain $G$.

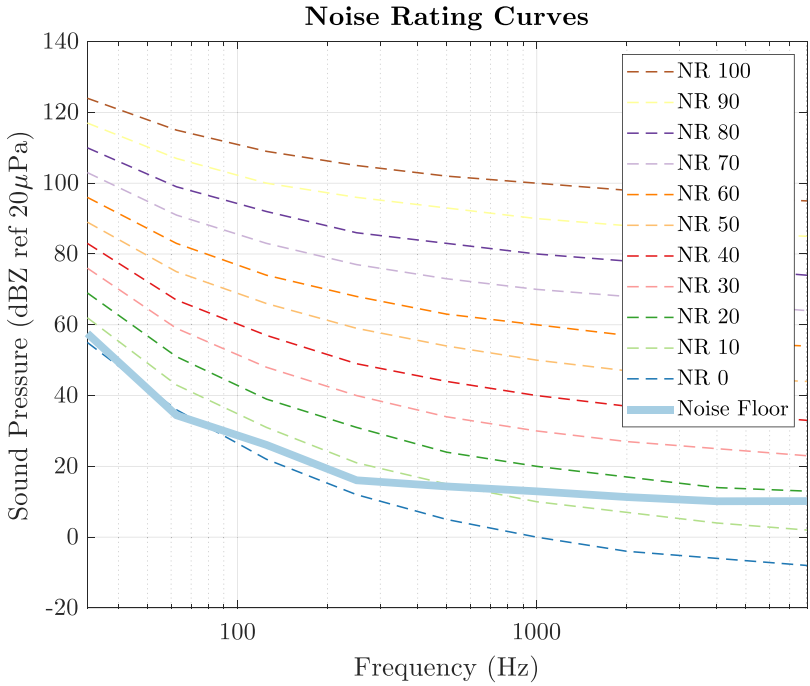


**Fig. 3.** Octave band plot of the noise rating (NR) curves from ISO 1996:1972 and the noise floor of the anechoic box measured with B&K 2245.

2. Calculate the $L_{eq}$ of $\boldsymbol{x}$ based on instantaneous sound pressure levels measured by the head and torso simulator.
3. Via search algorithms, repeat steps 1 and 2 while adjusting the gain $G$ of the stimulus until it reaches a value such that $G\boldsymbol{x}$ is measured by the head and torso simulator to have an $L_{eq}$ of $D$ dB within a tolerance of $T$ dB.
4. Save the adjusted gain $G$ into a *.csv* file for reference.

The saved gain values $G$ can then be applied to the stimuli $\boldsymbol{x}$ outside of the VI using any audio editing software in order to calibrate the stimuli. The search algorithms used are a boundary search and a binary search parameterized by a multiplier $M$ that controls the coarseness of the search. They are guaranteed to converge for deterministic signals, but due to noise and latency in measurements in practice, they may not converge for the real-life stimuli that the VI is designed to calibrate. Hence, we introduce an additional parameter $N$ that controls the maximum number of iterations that the search algorithms are allowed to take when running the VI. When this maximum number is exceeded, the VI categorically outputs a value of -1 for the gain $G$ into the *.csv* file to highlight to the user that the search has failed, since a successful search would always produce a positive value for the gain.

In addition, it is also possible for the saved gain values $G$ to cause clipping of the stimuli, especially if the variance of the stimuli is high. This is not normally desired, so the VI appends an additional remark in the *.csv* file to values of $G$ that result in clipping.

### *A. Playback and recording VI (MeasureLeq)*

The core component of the LabVIEW VI is the “MeasureLeq” function, which plays a given stimulus over a pair of circumaural headphones, and records the output with the microphones embedded into both ears of the head and torso simulator. LabVIEW inherently splits the power of mono tracks in order to play the same signal through both channels of a pair of headphones, but does not do so for

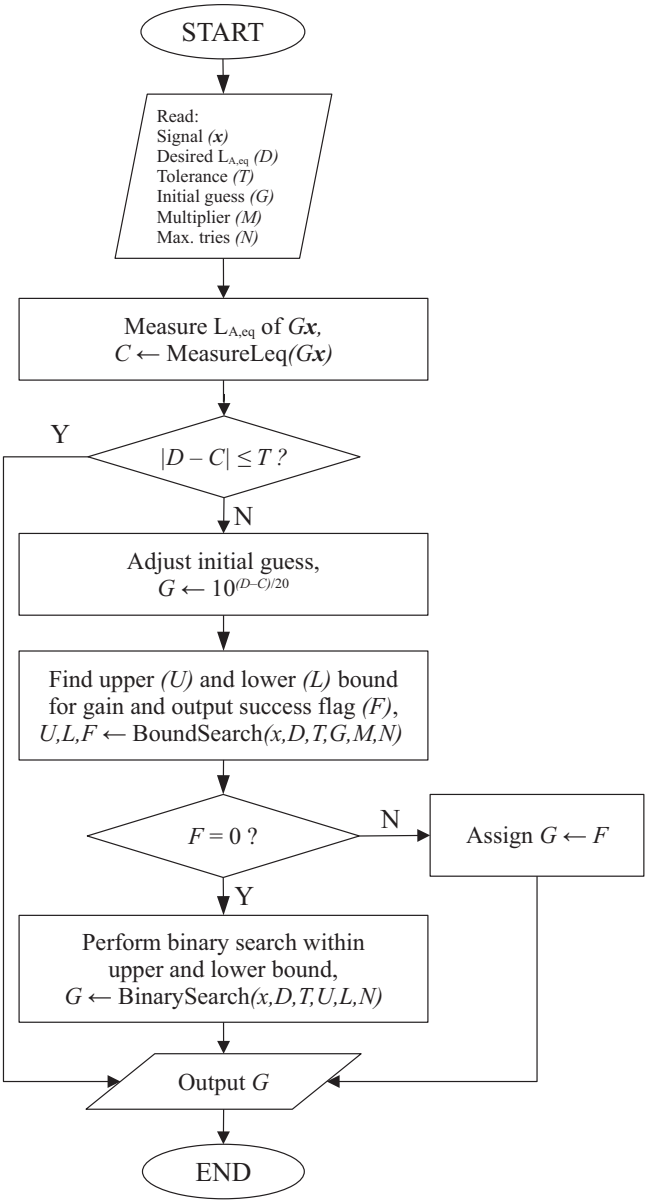


**Fig. 4.** Overall system flow for automatic calibration software in LabVIEW VI.

stereo tracks. Hence, to prevent an unwanted loss in power, MeasureLeq always generates a stereo track from a mono track by copying the same samples into both channels before using LabVIEW to initiate the playback. Binaural or stereo tracks are not modified by MeasureLeq before playback because they already have two channels.

During recording, MeasureLeq converts the microphone measurements in mV to instantaneous sound pressure in Pa based on the given microphone sensitivity that is read as user input to the function. The VI also allows the user to apply IEC 61672:2003 compliant weighting filters to the

instantaneous sound pressure measurements before feeding the signal to a built-in LabVIEW function that returns the $L_{eq}$ values in decibels. The final $L_{eq}$ measurement returned by MeasureLeq is the mean of the values obtained from the microphones in both ears of the head and torso simulator.

### *B. Initial guess adjustment*

Our VI accepts an initial guess for the gain $G$ required to calibrate the stimulus to the desired $L_{eq}$, which can be interpreted as a pre-gain applied to all stimuli. This could be useful if the user has prior knowledge of the power of the stimuli that they are attempting to calibrate, but in the absence of that information, a default value of $G = 1$ is assumed. An initial call to MeasureLeq is called with the initial guess for the gain, and the initial guess is then adjusted based on the measured $L_{eq}$.

The adjustment is based on the assumption of a perfectly linear system. Treating the raw audio data in each stimulus $\boldsymbol{x}$ as a random column vector, the average power of the signal is $P = E\{\boldsymbol{x}^T\boldsymbol{x}\}$, so applying a gain of $G'$ to it will result in the average power changing to $E\{(G'\boldsymbol{x})^T(G'\boldsymbol{x})\} = (G')^2P$. This means that the difference, in decibels, after applying a gain $G'$ is $20\lg(G')$. If MeasureLeq outputs a value of $C$ dB for the measured $L_{eq}$ of the stimulus and we want to calibrate the stimulus to an $L_{eq}$ of $D$ dB, then this difference is exactly $D - C = 20\lg(G')$. Rewriting this in terms of $G'$, we can obtain $G' = 10^{\frac{D-C}{20}}$, which is what the VI adjusts the initial guess $G$ to.

Although the assumption of perfect linearity is invalid for real-life systems, at high tolerance levels, this adjustment is mostly accurate, and the VI program stops here. However, at low tolerance levels, the error after adjustment tends to be larger than the specified tolerance and further adjustments to $G'$ have to be made empirically. These adjustments are performed by a boundary search "BoundSearch", which finds upper and lower bounds for these adjustments, and a binary search "BinarySearch", which searches between the upper and lower bounds for the correct adjustment. In this scenario, the adjusted initial guess also helps to provide a more optimal starting point for the boundary and binary search than the user-defined initial guess, thus reducing the time required for automatic calibration.

### *C. Boundary search (BoundSearch)*

Since the value for an appropriate gain required to calibrate a given track is theoretically unbounded, if the initial guess adjustment does not give the desired $L_{eq}$, the VI will next attempt to find a lower and upper bound for the appropriate gain in a boundary search. The flow for the boundary search is shown in Fig. 5.

The upper and lower bounds are first initialized based on an initial guess, which is actually the adjusted guess from Section 3.B. If the measured $L_{eq}$ at the upper bound value for the gain is lower than the desired $L_{eq}$, then both the upper and lower bounds are scaled by a user-defined factor of $M > 1$, whereas if the measured $L_{eq}$ at the lower bound value for the gain is higher than the desired $L_{eq}$, then both the upper and lower bounds are scaled by $\frac{1}{M}$. The process is repeated until the measured $L_{eq}$ at the upper and lower bound values for the gain are respectively higher and lower than the desired $L_{eq}$, in which case the appropriate gain does indeed lie between the upper and lower bound values. The boundary search then stops and the VI passes the upper and lower bounds to the binary search function described in Section 3.D. We can see that by inputting larger values of $M$, the user can allow the boundary search to converge faster at the expense of a larger search interval for the subsequent binary search.

As a method to eliminate redundancy in the search process, if one of the bounds happens to be measured as an appropriate gain $G$ to calibrate a given track to the desired $L_{eq}$, then the VI immediately writes the corresponding bound to the *.csv* file instead of continuing with the binary search. This is implemented in the VI by the BoundSearch function outputting a flag $F$ that is exactly $G$ if this occurs and 0 if this does not occur. In addition, to prevent a potential infinite loop from occurring due to noise or unexpected events during measurements, if the number of iterations in the boundary search exceeds a user-defined limit, the VI outputs a value -1 for the flag $F$. The overall system in Fig. 4 then executes the necessary actions based on the value of the flag received.

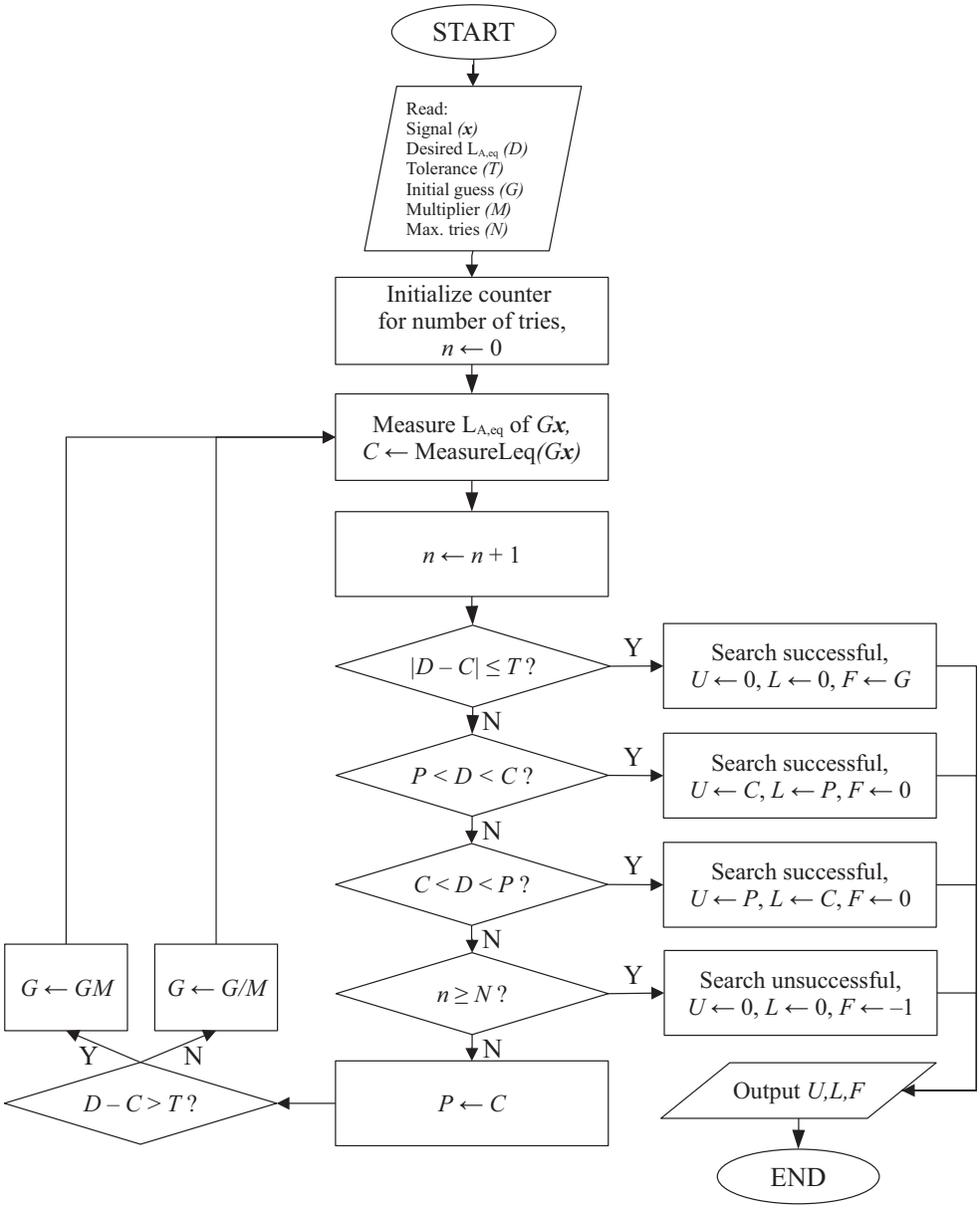


**Fig. 5.** Subsystem flow for boundary search, BoundSearch(x,D,T,G,M,N).

*D. Binary search*

The final step for the VI in finding an appropriate gain to calibrate a stimulus to the desired $L_{eq}$ is a standard binary search, if such a gain has yet to be found. The upper and lower bounds are obtained from the output of the BoundSearch function and the binary search is performed with those values as the initial bounds. The flow for the binary search is shown in Fig. 6.

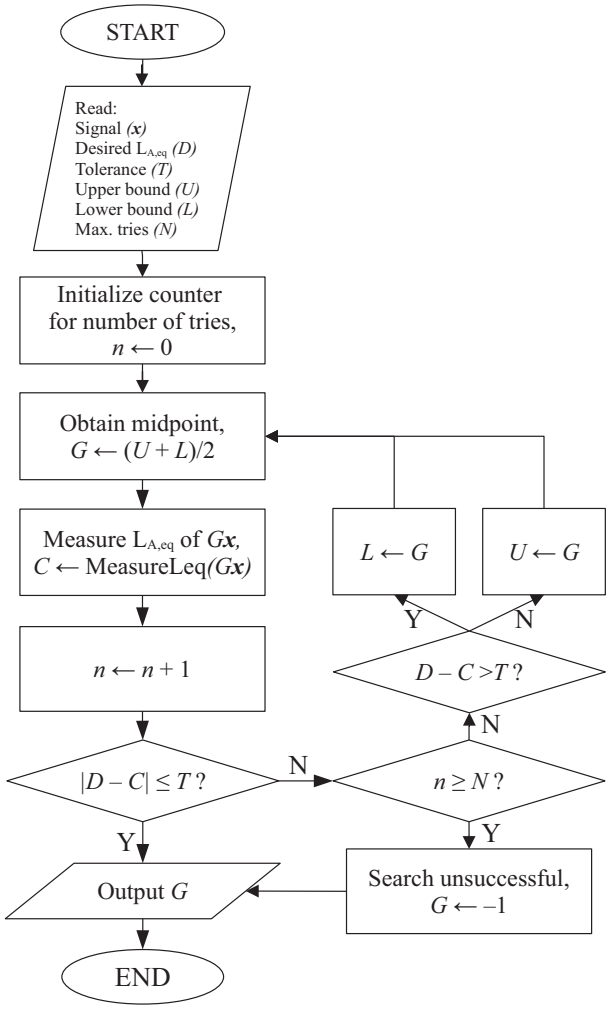


**Fig. 6.** Subsystem flow for binary search, BinarySearch(x,D,T,U,L,N).

Similar to the boundary search in Section 3.C, if the number of iterations in the boundary search exceeds a user-defined limit, the VI outputs a value -1 for the gain *G*. This serves an identical purpose as a value for -1 for the flag *F* in the boundary search, and is a catch-all to prevent infinite or overly long runtime.

## 4. Method details

In order to calibrate a set of stimuli to a desired $L_{eq}$, the specific steps for our method are as follows:

1. Set up hardware as shown in Fig. 2.
2. Ensure that the desired $L_{eq}$ is above the noise floor of the calibration environment.
3. Prepare all audio files to be calibrated as mono or stereo .wav files.
4. Run the LabVIEW VI described in Section 3 after inputting parameters as necessary into the panels of Fig. 7. The device ID is the index of the soundcard on the computer running the

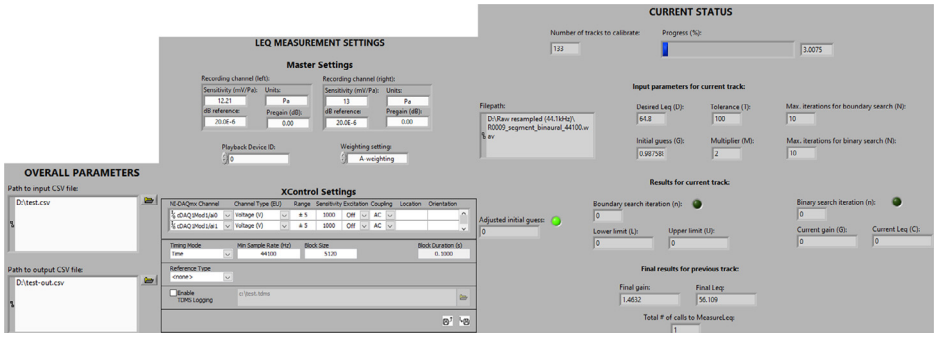


**Fig. 7.** LabVIEW VI interface showing input interface (left) adjustable parameters for $L_{eq}$ measurement (middle), and live status console with adjustable parameters for binary and boundary search (right).

LabVIEW VI, and the sensor sensitivity (in mV/Pa) for both the left and right channels is usually stated on the datasheet for the head and torso simulator. The weighting setting can be set to any standard weighting filter or no filter, but A-weighting is used for most soundscape studies. The binary and boundary search parameters are as described in Section 3.

5. If an average $L_{eq}$ of multiple headphone positions is desired, reposition headphones and perform step 2 again.
6. Take average gain of all positions, as stated in the output *.csv* files, if desired.
7. Process stimuli in original files by multiplying average gain sample-wise using any audio editing program, such as Audacity, Python, or LabVIEW.

Internally, the LabVIEW VI detects clipping when it attempts to play samples outside of the [-1,1] range, which could happen when it digitally sets the gain to large values. The VI sets such samples to the value of the bound that they are closest to as a software-based method to prevent playback errors and damage to the playback system. However, the user is notified in the output *.csv* file if this occurs. Upon receiving the notification, the user should increase the headroom of the playback system to prevent clipping if it is not desired. This could be done by increasing the power provided to the playback system hardware (such as by turning up the gain of the soundcard) or by reducing the desired $L_{eq}$. The latter solution is used when the playback system is running at its maximum rated power, because while it is theoretically possible to use the VI to calibrate a stimulus to an arbitrarily high $L_{eq}$, real-life systems are unable to reach arbitrarily high $L_{eq}$ levels. Since the LabVIEW VI is designed to work with any playback device and soundcard recognized by the computer running the program, it has no access to driver or physical hardware settings to prevent clipping, so the rectification must be performed manually by the user.

Furthermore, in the rare event that a track fails to be calibrated, we will run the LabVIEW VI again for that track, with the adjusted initial guess from the failed run as the initial guess for the new run. This value can be read from the “Current Status” panel of the VI, as shown on the right of Fig. 7. Alternatively, the maximum number of iterations for the search algorithms can be increased in the *.csv* file specified in the “Overall Parameters” panel.

## 5. Method validation

In practice, the MeasureLeq function takes the longest time to run because any stimulus must be played back in full before the VI can output its $L_{eq}$ as measured by the microphones in the head and torso simulator. The length of time required for the MeasureLeq function to complete running is therefore dependent on the length of the audio stimuli that we want to calibrate, so for a consistent benchmark, so we measure the efficiency of our proposed method in number of calls to the MeasureLeq function. All other processing steps in the VI take negligible time to run compared to the MeasureLeq function, so the length of time required to output a valid gain $G$ that adjusts a stimulus

**Table 2**
Mean number of calls to MeasureLeq (± SD) made to calibrate 250 different audio stimuli, each normalized to a random range, to an A-weighted $L_{eq}$ of 65 dB at various tolerance levels with and without initial guess adjustment. With initial guess adjustment, an additional call to MeasureLeq is made outside of both the boundary and binary search.

| Tolerance (dB) | With initial guess adjustment | | | Without initial guess adjustment | | |
|---|---|---|---|---|---|---|
| | Total | Boundary | Binary | Total | Boundary | Binary |
| 0.1 | 2.67 ± 2.05 | 1.10 ± 0.31 | 0.57 ± 1.75 | 7.67 ± 1.73 | 3.72 ± 1.13 | 3.95 ± 1.31 |
| 0.2 | 2.29 ± 1.25 | 1.05 ± 0.25 | 0.24 ± 1.02 | 6.73 ± 1.75 | 3.69 ± 1.13 | 3.04 ± 1.19 |
| 0.3 | 2.15 ± 0.87 | 1.02 ± 0.23 | 0.13 ± 0.68 | 5.99 ± 1.71 | 3.66 ± 1.14 | 2.32 ± 1.11 |
| 0.4 | 2.14 ± 0.85 | 1.02 ± 0.23 | 0.12 ± 0.66 | 5.72 ± 1.70 | 3.65 ± 1.14 | 2.07 ± 1.05 |
| 0.5 | 2.06 ± 0.52 | 1.01 ± 0.17 | 0.05 ± 0.37 | 5.42 ± 1.62 | 3.63 ± 1.14 | 1.79 ± 1.01 |

$\boldsymbol{x}$ to a desired $L_{eq}$ $D$ dB is almost directly proportional to the number of calls to the MeasureLeq function.

To validate the increased efficiency of the proposed initial guess adjustment, we attempted to calibrate 250 different audio stimuli to an A-weighted $L_{eq}$ of 65.0 dB at various tolerance levels using the setup in Fig. 2. The stimuli were generated from publicly available binaural recordings[1] of the "Urban Soundscapes of the World" project [7]. The choice of this set of recordings as validation stimuli was motivated by the diversity of soundscapes in the dataset across different cities, which would allow for the investigation of whether the proposed procedure would be affected by the audio material used.

Specifically, recordings R0001 to R0125 were used. Each 60-second binaural recording was divided into two equal sections, each 30 s in length, to generate the 250 different audio stimuli. Each stimulus was then normalized to the range $[-A, A]$ before calibration, where $A$ is a uniform random variable in $(0, 1)$. The initial guess, multiplier, maximum tries for the boundary search, and maximum tries for the binary search were set to 1, 2, 10, and 10, respectively. The number of calls to MeasureLeq made by each search algorithm and the total number of calls made by the VI are shown in Table 2.

All 250 stimuli were calibrated successfully, without the VI outputting a value of -1 for either the boundary search flag $F$ or gain $G$, thus demonstrating that the automation by the VI was successful was not adversely affected by the diverse acoustic characteristics of the validation stimuli. A Wilcoxon rank-sum test was also performed between the total number of guesses with and without adjustment at each tolerance level. The p-values for all 5 tolerance levels in Table 2 were below 0.01, thus demonstrating that the initial guess adjustment significantly reduced the number of calls to the MeasureLeq function and allowed the automation process to complete faster on average.

## 6. Conclusion

In this article, we have proposed a method and provided a freely-available LabVIEW VI for the automatic calibration of arbitrary audio stimuli to desired $L_{eq}$ values within the acoustic headroom of a playback and recording system. The LabVIEW VI is designed to work with any computer able to run the LabVIEW software and any playback and recording system complying with the specifications in Annex D of ISO 12913-2:2018. The calibration algorithm makes use of an adjustment to the initial guess, followed by a boundary and binary search. A trial run of the entire calibration system, where we attempted to calibrate 250 different audio stimuli from the "Urban Soundscapes of the World" dataset, finally allowed us to validate the efficacy and performance of the system.

## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

[1] The recordings are available at http://urban-soundscapes.org/soundscapes/.

## Acknowledgments

This research/project is supported by the National Research Foundation, Singapore, and Ministry of National Development, Singapore under its Cities of Tomorrow R&D Program (CoT Award: COT-V4–2020–1), as well as the Singapore Ministry of Education Academic Research Fund Tier-2 (Research grant: MOE2017-T2-2-060). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the view of National Research Foundation, Singapore and Ministry of National Development, Singapore. The authors would like to thank Hottinger Brüel & Kjær (South Asia Pacific) for the loan of the B&K 2245.

## References

[1] L. Maffei, M. Masullo, A. Pascale, G. Ruggiero, V.P. Romero, Immersive virtual reality in community planning: acoustic and visual congruence of simulated vs real world, Sustain. Cities Soc. 27 (2016) 338–345.
[2] H.I. Jo, J.Y. Jeon, Effect of the appropriateness of sound environment on urban soundscape assessment, Build. Environ. 179 (2020) 106975.
[3] J.Y. Hong, et al., Effects of adding natural sounds to urban noises on the perceived loudness of noise and soundscape quality, Sci. Total Environ. 711 (2020) 134571.
[4] C. Guastavino, B.F.G. Katz, J.D. Polack, D.J. Levitin, D. Dubois, Ecological validity of soundscape reproduction, Acta Acust. united Acust. 91 (2) (2005) 333–341.
[5] International Organization for StandardizationAcoustics — Soundscape — Part 2: Data collection and Reporting Requirements, International Organization for Standardization, Geneva, Switzerland, 2018.
[6] International Telecommunication Union Radiocommunication Sector, ITU-R BS.1116-3: methods for the subjective assessment of small impairments in audio systems, Geneva: International Telecommunication Union, 2015.
[7] B. De Coensel, K. Sun, D. Botteldooren, Urban soundscapes of the world: selection and reproduction of urban acoustic environments with soundscape in mind, in: Proceedings of the 46th International Congress and Exposition on Noise Control Engineering, 2017 Taming Noise Mov. Quiet, vol. 2017-January.