Proceedings of the ASME 2017 International Mechanical Engineering Congress and Exposition
IMECE2017
November 3-9, 2017, Tampa, Florida, USA

IMECE2017-71430

PARTICIPATORY NOISE MAPPING: HARNESSING THE POTENTIAL OF SMARTPHONES THROUGH THE DEVELOPMENT OF A DEDICATED CITIZEN-SCIENCE PLATFORM

Eoin A. King
Acoustics Program and Lab
Mechanical Engineering,
University of Hartford
West Hartford, CT

Akin Tatoglu
Mechanical Engineering,
University of Hartford
West Hartford, CT

Robert D. Celmer
Acoustics Program and Lab
Mechanical Engineering,
University of Hartford
West Hartford, CT

ABSTRACT

This paper presents results of an ongoing project which aims to develop a purpose-built platform for using smart phones as alternative to sound level meters for citizen-science based environment noise assessment.

In order to manage and control environmental noise effectively, the extent of the problem must first be quantified. Across the world, strategic noise maps are used to assess the impact of environmental noise in cities. Traditionally, these maps are developed using predictive techniques, but some authors have advocated the use of noise measurements to develop more reliable and robust noise maps.

If adopted correctly, smartphones have the capability to revolutionize the manner in which environmental noise assessments are performed. The development of smartphone technology, and its impact on environmental noise studies, has recently begun to receive attention in the academic literature. Recent research has assessed the capability of existing smartphone applications (apps) to be utilized as an alternative low-cost solution to traditional noise monitoring. Results show that the accuracy of current noise measurement apps varies widely relative to pre-specified reference levels. The high degree of measurement variability associated with such apps renders their robustness questionable in their current state. Further work is required to assess how smartphones with mobile apps may be used in the field and what limitations may be associated with their use.

To overcome the above issues, this project is developing a platform specifically for citizen science noise assessment. The platform consists of a smartphone app that acquires a sound signal and transfers the data to a server via a web based API for post processing purposes. This then returns key information to the user, as well as logging the data for use in a massive noise mapping study. The structure of the proposed platform maintains a clear separation between client (phone) and server. This approach will allow implementation of future open source client side apps for both Android and iOS operating systems.

INTRODUCTION

Smartphones have become a must-have for the majority of adult citizens in the world’s developed nations. As of October 2014, 64% of U.S. adults own some form of smartphone [1]. The development of smartphone technology and its impact on environmental noise studies has only recently begun to receive attention in the academic literature [2-4]. If adopted correctly, smart phones have the capability to revolutionize the manner in which environmental noise assessments are performed. This paper presents results of an ongoing project which aims to develop a purpose built platform for using smart phones as an alternative to sound level meters for citizen-science based environment noise assessment. The key feature of the proposed approach is a clear separation between client (phone) and server as illustrated in Fig.1.

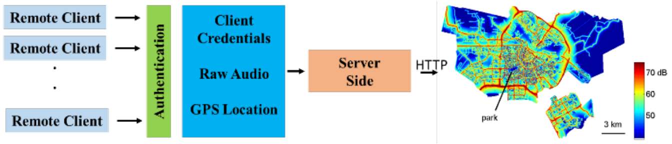


Fig. 1 Working principle of platform: Data is received from remote users and transferred to server with appropriate credentials. After server processes all data, it publishes the noise map via web.

Previous Research

Some recent academic studies suggest that smartphones are capable of replacing traditional noise assessment devices such as sound level meters in the near future. Kanjo (2010) outlined the possibility of developing a mobile phone platform for measuring noise in cities and highlights the potential of such avenues for the future [2]. Similarly, D'Hondt et al (2013) have demonstrated the possibility of smartphone-based noise apps to be utilized by ordinary citizens as a form of crowd-sourced participatory noise assessment in cities [3]. Studies such as these suggest that the future of noise assessment, whether it is in cities or elsewhere, will likely be tied closely to developments in smartphones and other forms of innovative mobile technology that are relatively affordably and easily accessed by ordinary citizens, especially in developed nations.

Most recently, using the acoustical test facilities at the University of Hartford, the authors led a study to assess the capability of existing mobile apps to be utilized as an alternative low cost solution to traditional noise monitoring using a sound level meter [4]. The methodology consisted of testing 100 smartphones in The Paul S. Veneklasen Research Foundation Reverberation Room. Broadband white noise was used to test the ability of smartphones to measure sound pressure levels at background, 50, 70 and 90 dB(A), and these measurements were compared with true noise levels acquired via a calibrated Type 1 sound level meter. Tests were conducted on phones using both the Android and iOS platforms. For each smartphone, tests were completed separately for several leading noise monitoring apps, culminating in 1472 tests. The apps that were tested are listed in Table 1 with overall results summarized in Fig. 2.

Table 1. List of apps tested for previous study [4]

Name	Developer
Sound Level Analyzer Lite (iOS)	Toon LLC
SPLnFFT (iOS)	Fabien Lefebvre
Decibel Meter Pro (iOS)	Performance Audio
UE SPL (iOS)	Logitech Inc.
Sound Meter (Android)	Smart Tools co.
Noise Meter (Android)	JINASYS
Decibel Pro (Android)	BSB Mobile Solutions Tools

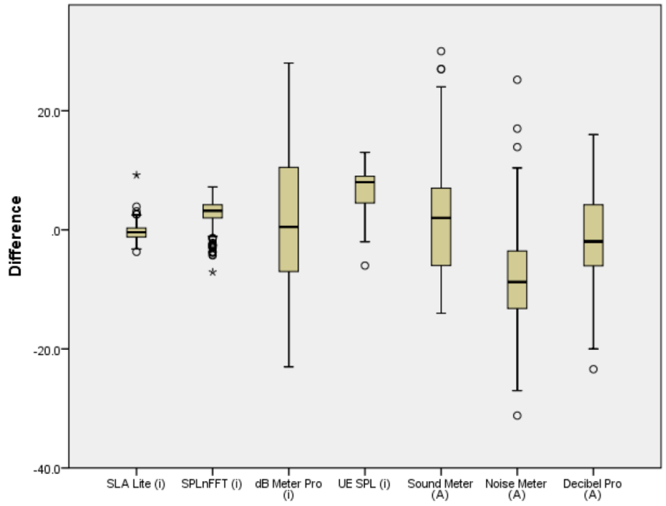


Fig. 2 Boxplot showing data distribution of difference between reference and measured values by smartphone application [4]. Outliers are indicated by asterisks and circles.

The results of this study indicated that two of the apps tested were within ±2 dB of true noise levels across the measurement range, but apps (and phones) had considerable nonlinear measurement variability. Overall, the accuracy of noise measurement apps varied widely relative to pre-specified signal intensity reference levels. iOS apps performed better than Android based apps [4], however the high degree of measurement variability associated with such apps (and phones) renders their reliability questionable in their current state.

Objective

It is clear that the use of smart phones for environmental noise assessment is still very much in its infancy. Further work is required to assess how smart phones with mobile apps may be used in the field and what limitations may be associated with their use. The objective of the current project was to develop a preliminary mobile app for the Android platform, developed specifically for environmental noise measurement. The Android platform was chosen as it is an open-source operating system ideal for preliminary development. The following key capabilities of the app were identified as minimum requirements for an app capable of performing environmental noise assessment: i) reporting a noise levels in terms of $L_{\mathrm{Aeq,T}}$; ii) storing the geographical location of a measurement point using the smart phone's GPS capabilities; iii) taking a photo/video of the scene where data is logged; iv) a feature to capture a short audio sample of the noise source being recorded.

In order to capitalize on the phones networking capabilities, it was decided that this app should feed information directly into an online platform that would store all data. This would allow the computation of intensive tasks and enhanced analyses via post processing of data through this platform. Thus the app acquires a sound signal and transfers the data to a web based API for processing of the signal. This then returns key information to the user, as well as logging the data for use in a massive noise mapping study.

The structure of our platform maintains a clear separation between client and server. This decouples the two parties so that one can be replaced or changed provided the interface stays consistent. This will also make the platform accessible to both Android and iOS operating systems in the future.

This paper reports on the development of this platform. All preliminary testing has been performed using a Samsung Galaxy S6 smartphone (with Android operating system). Initial tests were performed on a single phone (and single app) to ensure consistency in results.

MOBILE PLATFORM

Mobile Operating System

The Android mobile operating system is based on Linux kernel and designed to optimize system resource use. Its cross-platform architecture is designed to allow for quick and easy modulation on any device. Operating system offers various low level services and their APIs to communicate and control devices. There are two main continuously running system services: Media server and system server. While the system server manages communication procedures between higher/user level and core services protocol, a media server handles all requests to access system hardware such as camera, audio, GPS, IMU and other components. A generic structure is presented in Fig. 3.

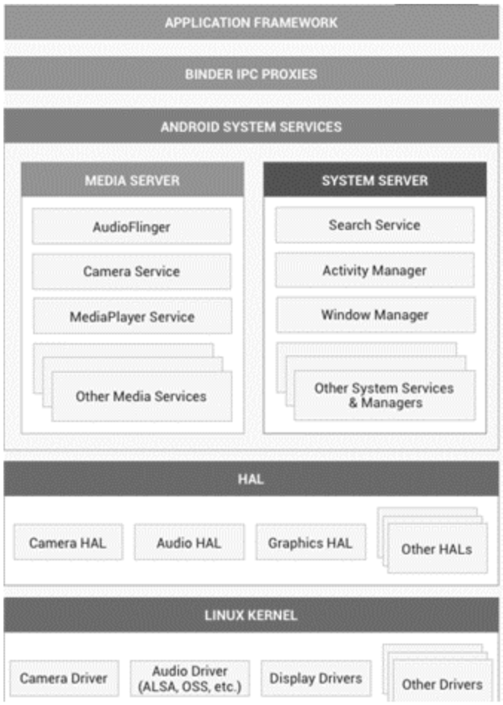


Fig. 3 Mobile Operating System Core Structure

Client Login

For remote client login and easy data transfer, an external API was used. Initially, a new blank activity is created, which is an application component that initializes a new window object for users to interact with in order to complete a task. It contains both a java class file and an xml file. The java file is the code that preforms all the tasks in the activity, while the xml file is the code that creates the UI.

Audio Access

APIs for this app includes "MediaRecorder" object which accesses the microphone. This routine was used to save data in various types including compressed and raw formats as shown in Table 2. Different types of formats were tested and finally data are saved in raw AAC format for further analysis.

Table 2. List of available media formats

Format	Extension	Media Type
AAC ADTS (Audio Data Transport Stream)	ADTS	Audio
AMR_NB (Adaptive Multi-Rate Narrowband)	AMR	Audio
MPEG_4	MP4	Audio/Video
RAW_AMR (Adaptive Multi-Rate)	AMR	Audio
THREE_GPP	3GP	Audio/Video
WEBM	WEBM	Audio/Video

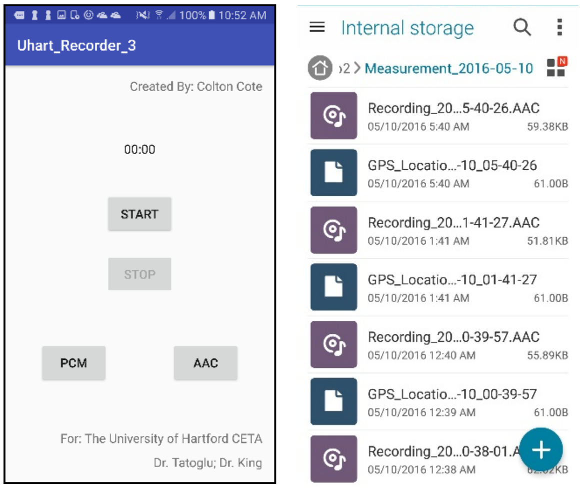
Other APIs used include "LocationListener" for GPS location logging and "GraphView" to plot the data. All the imported libraries are listed in Table 3.

Table 3. List of Device Drivers and APIs

Imported Library	Uses
import android.media.AudioFormat;	Setting the audio format
import android.media.AudioRecord;	Recording raw PCM
import android.media.MediaRecorder;	Audio file
import android.location.Location;	GPS location
import com.google.api.services.drive.Drive;	Google Drive

User Interface and File Structure

After a successful login operation, a user-friendly interface was rendered as shown in Fig.4, left. Once an individual experiment was completed, files were saved as shown in Fig.4, right.



Recording_20...5-40-26.AAC 05/10/2016 5:40 AM 59.38KB
GPS_Locatio...-10_05-40-26 05/10/2016 5:40 AM 61.00B
Recording_20...1-41-27.AAC 05/10/2016 1:41 AM 51.81KB
GPS_Locatio...-10_01-41-27 05/10/2016 1:41 AM 61.00B
Recording_20...0-39-57.AAC 05/10/2016 12:40 AM 55.89KB
GPS_Locatio...-10_00-39-57 05/10/2016 12:39 AM 61.00B
Recording_20...0-38-01.A 05/10/2016 12:38 AM

Fig. 4 Sample Interface and File Hierarchy

Remote Sensing Mobile App

A generic flow of the algorithm is presented in Fig. 5. After user interface object initialization, a sign-in API automatically calls if a client had previously logged-in. Necessary folders are generated and the audio driver, encoder as well as initialization of GPS objects. At this point, the app expects a user action to start the experiment. Once these data are logged, files are initially saved to remote client shared folder to be uploaded to the server.

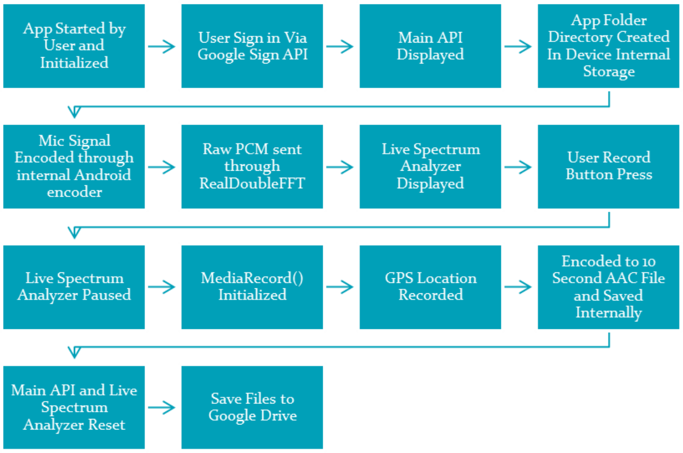

App Started by User and Initialized
User Sign in Via Google Sign API
Main API Displayed
App Folder Directory Created In Device Internal Storage
Mic Signal Encoded through internal Android encoder
Raw PCM sent through RealDoubleFFT
Live Spectrum Analyzer Displayed
User Record Button Press
Live Spectrum Analyzer Paused
MediaRecord() Initialized
GPS Location Recorded
Encoded to 10 Second AAC File and Saved Internally
Main API and Live Spectrum Analyzer Reset
Save Files to Google Drive

Fig. 5 Flow of the application

Back-end Processing

For this project, an open source operating system - Ubuntu - was used to handle server end operations. Ubuntu was selected for its wide community support and required programming environment including PHP 5 and Python scripts as well as easy to manage database system MySQL.

The routines implemented were based on REST API structure. REST is not standardized and is considered more of a guideline or style when designing an API. Some characteristics that define a REST API can be very obvious and are a given when presenting data over HTTP, such as data should be conceptually separate from the method in which they are stored, meaning that the data should be presented in XML or JSON and not as a database. The response should also specify its data type so that the client can read the data reliably.

The API should also maintain a clean separation between server operation and remote client process. This separation means that clients should not be concerned with any of the server-side data storage and the server should not be concerned with any user interface components: they should be tested and implemented individually.

Another advantage of this separation is to avoid server-side code updates when another mobile operating system is used: all data managed via HTTP requests and any remote client can access the server with appropriate credentials. In addition to that, this setup was very suitable for various level student involvement into the project.

MEMS Structure

MEMS microphones are miniaturized condenser microphones. They have diaphragms that are made up of two capacitor plates and vibrate in respect to each other with small changes in atmospheric pressure. This creates a variation in the capacitance and is then amplified by the integrated circuit to produce a digital output signal.

Fig. 6 illustrates the displacement of a wafer of a basic condenser microphone structure with the attached circuit. A DC voltage, VDC is applied to the microphone it creates an electric charge QDV on the surface of the membrane. This voltage creates an electrostatic force which causes a static average displacement to the membrane, x_{DC}.

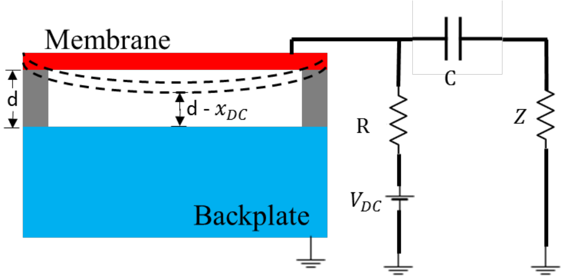


Fig. 6 Basic structure of condenser microphone

Once an acoustic wave strikes the membrane surface, the distance of the membrane to the back plate changes and its average x is given by

$$x = d - x_{DC} + x_{ac} \qquad (1)$$

where x_{ac} is the dynamic average displacement of the membrane while vibrating. The frequency response, displacement behavior, as well as boundaries and amplifier gains vary for different manufacturers, therefore for different mobile phones. A future goal of this project is to generate a function for each mobile system type that different audio signals could be normalized while representing real signal amplitude. In the following section, behavior analysis of the microphone with respect to known input signal is discussed.

ACOUSTIC CHARACTERIZATION TESTS

In order to perform robust and accurate acoustic measurements, the acoustics characteristics of the platform (smartphone microphone and app) were determined following acoustic tests in the Paul S. Veneklasen Foundation Anechoic Chamber at the University of Hartford (qualified for free-field measurements for one-third octave bands of 100 Hz and above per ISO 3745-2003 [5]).

Test Set-up and Preliminary Tests

A Genelec 8030A loudspeaker was used to reproduce a known output signal for all tests. The frequency response of this loudspeaker is flat or within a negligible 1 dB for an angular variation of 15° on either side of its acoustical axis (Fig. 7).

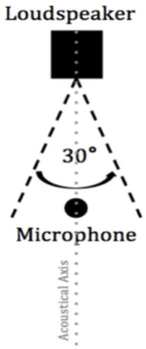


Fig. 7 Range of measurement within 1dB

Thus a microphone, placed within the 30° range presented in Fig. 7, will receive the expected output from the loudspeaker, with no significant variation in sound level. Any variation in the recorded signal will be due to the measurement platform.

Fig. 8 show the general set-up of all tests. The smartphone was placed on a tripod near the center of the chamber, with the phone in the correct z-plane of the loudspeaker's acoustical axis (140mm up from base of loudspeaker). The loudspeaker was placed on a tripod and kept at a constant distance of 1m from the microphone throughout all tests. To ensure the loudspeaker was performing as expected, all tests were first conducted with a Brüel & Kjær calibrated Type 4190 omnidirectional microphone in the position of the smartphone. This microphone was used to calibrate the entire test set-up and served as a baseline dataset. Initial testing found that the Genelec 8030A includes a driver unit protection circuit where the loudspeaker rolls back its output at approximately 97 dB (re: 20µPa) @ 1 meter to avoid loudspeaker damage. Thus all tests were limited to a source maximum of 88 dB. This is acceptable as environmental noise levels outdoors would rarely exceed this level.

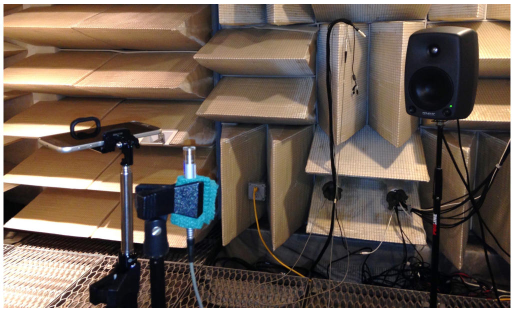


Fig. 8 Basic Test set-up

Sensitivity

Sensitivity is most commonly determined by generating a known 1000Hz sine wave signal at a sound pressure level of 94 dB (re: 20µPa) and measuring the output voltage produced by the microphone. However, for this case tests were performed over a range of sound pressure levels to test for linearity in sensitivity across the dynamic range of interest. This consisted of playing multiple 30 second bursts of a 1000Hz tone, initially at 34dB (at the receiver position) and increasing in increments of 6dB up to 88dB and recording the response for each burst. Sample output obtained from the app is presented in Fig.9.

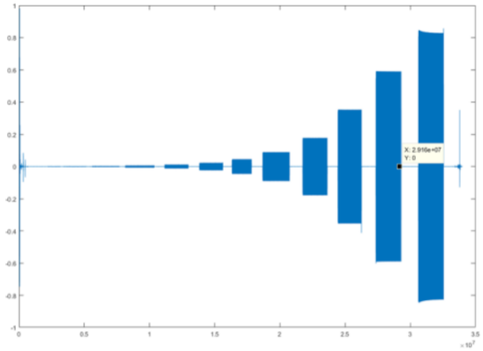


Fig 9. Example Normalized acoustic intensity vs. Time Plot

Fig. 9 presents output from multiple 30 second signals, increasing in magnitude from left to right on the time scale. This data was exported to MATLAB and analyzed to determine a sensitivity at 1000Hz. Results are presented in Fig. 10. The app performed in a typical fashion expected of a microphone, with linearity and minimal variation across the dynamic range of testing.

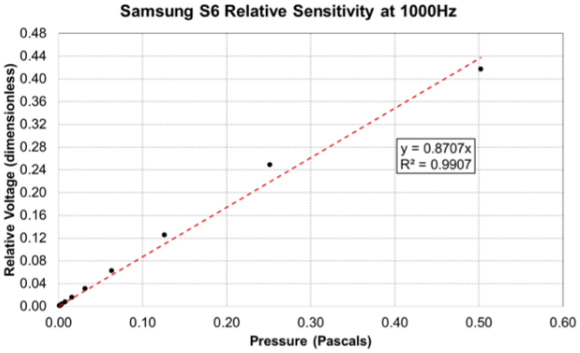


Fig 10. Linear relationship between voltage and pressure at 1000Hz.

Frequency Response

The second set of tests was performed to assess how the app responded across the frequency spectrum. Tests focused on the frequency range from 125Hz to 2000Hz as most of the energy in environmental noise would be contained in this range. For road traffic noise, about 70% of A-weighted sound energy is produced at around 1000 Hz [6]. Initially the sensitivity tests described above were repeated for each frequency. Similar results were observed for all frequencies (Fig. 11 and Fig. 12 display sample results for 500Hz and 2000Hz).

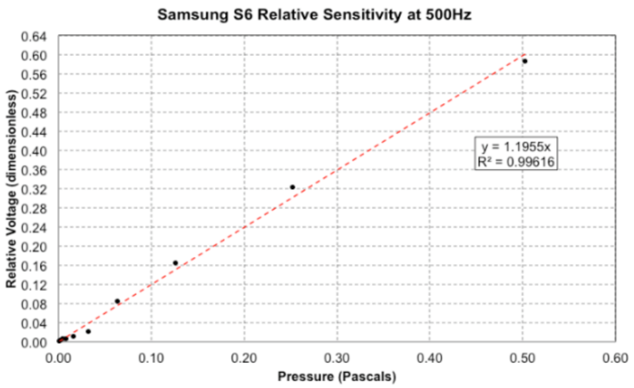


Fig 11. Relationship between voltage and pressure at 500Hz.

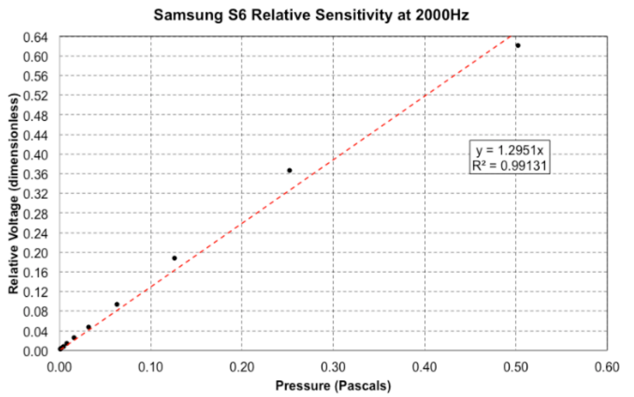


Fig 12. Relationship between voltage and pressure at 2000Hz

A more complete picture of the platform's sensitivity can be created by testing across a broader range of frequencies and examining an overall frequency response plot. This was determined by generating pink noise (random noise having equal energy per octave) through the Genelec 8030A and recording the app's output. Constant Percentage Bandwidth one-third octave filters were applied to the output and the frequency response of the system was determined. The overall response was subsequently converted to dB using the average sensitivity from all frequencies.

To assess the overall response of the app, results were compared to the reference class 1 microphone in the chamber (Fig. 13). The deviation from the reference microphone (baseline) was relatively consistent from approximately 200Hz to 3150 Hz, a range which includes the dominant frequencies contained in environmental noise. However, the average error in this frequency range was approximately 4dB.

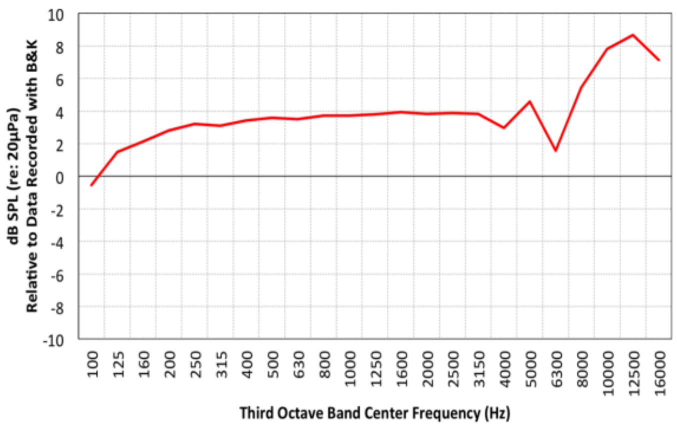


Fig 13. Plot of frequency response of developed app.

Tests were repeated using the voice recording native to the Samsung Galaxy S6 instead of the developed app. Output data was processed in the same way and compared to the reference microphone. Fig. 14 displays the resulting frequency response.

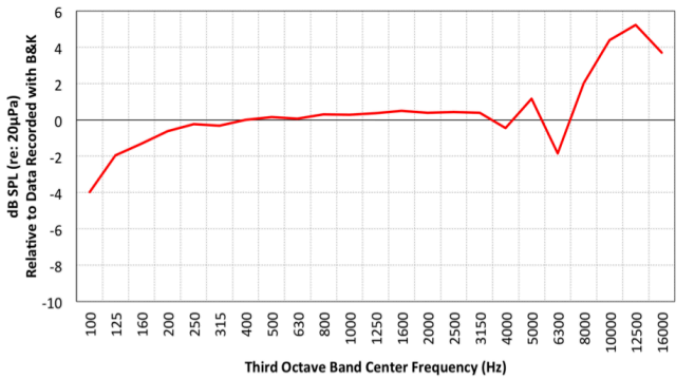


Fig 14. Plot of frequency response of native voice recorder app

Data acquired using the native voice recorder exhibited an almost identical frequency response to the data acquired with the developed app. This suggests the smartphone can be reliably used between 200Hz to 3150 Hz, and potentially at higher frequencies if a correction factor is applied.

A fixed offset in data recorded with the app was observed. This was believed to be due to internal processing in the voice recorder application (perhaps through the use of a filter or a non-linear sensitivity to normalize the sound level data). This normalization filter effect would not be captured in the developed app data because the app only uses the smartphone's microphone hardware, bypassing any filtering that may be present in the voice recorder software. It is recommended that future testing with the phone should make use of this voice recorder sensitivity in order to achieve the most accurate sound level results.

From both tests it is clear that above 4000Hz the frequency response varies significantly. This large peak in sound level in this high frequency range is a common feature of smartphone MEMS microphones, and is related to the Helmholtz effect. The wavelengths in this high frequency range are small enough that they reverberate within the air cavity of the microphone chamber itself, causing a resonance that amplifies the higher frequencies. On the opposite end of the spectrum, the slight roll-off in the 100-125Hz range is believed to be due to the inability of these larger wavelengths to negotiate the microphone ventilation hole geometry [7].

Overall, for general environmental noise studies using A-weighting this frequency range should be acceptable. A-weighting attempts to replicate the response of the human ear and weights low frequency noise. The use of A-weighting has become the de facto accepted descriptor for environmental noise and numerous studies have shown that A-weighted sound levels provide an acceptable correlation with human response to different noise sources [8]. However, it appears the app will be limited in any attempt to measure low frequency noise, and is therefore not appropriate for any assessment that may require C-Weighting (Fig. 15). Impulsive noise such as sledgehammer blows on a construction site may be assessed in terms of C-Weighting.

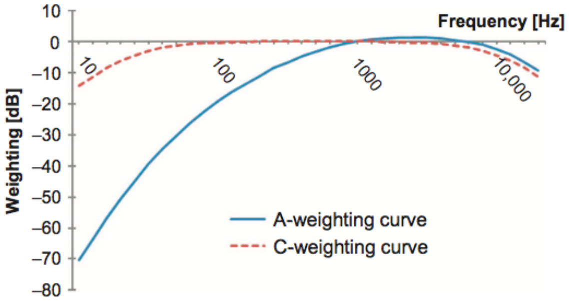


Fig 15. Standard weighting curves for A and C Weighting

This means that smartphones will likely not be able to detect the presence of low frequency content, which is often related to noise complaints. Further testing at low frequencies could produce an empirically-derived correction factor in this range for Samsung Galaxy S6 smartphones that could be applied to the final app's code.

Directivity

The final characteristic assessed was the directivity pattern of the smartphone. The directivity pattern indicates how the microphone sensitivity varies according to the direction from which sound arrives.

To assess the directivity pattern, a similar test regime to previous tests was followed. The source test signal was pink noise at 80 dB at the phone location. The spectral range of this random noise was between 100-16,000Hz in order to avoid data contamination in the lower frequency range due to doors slamming, HVAC rumble, etc. To change the incident angle of the sound arriving at the microphone, the phone was rotated about its z axis, keeping the microphone and loudspeaker in the same x-y plane, as shown in Fig. 16 below. The incremental angle of rotation (θ) was 10 degrees, up to 90 degrees in both directions. The phone's microphone angle was adjusted relative to the loudspeakers acoustical axis. Fig. 17 show the experimental set up in the anechoic chamber. The green tape on the floor indicates the 30 degree range in which the loudspeaker's output is uniform to within 1dB, meaning the smartphone had to remain within this region. A tripod mount was used to hold the phone instead of a person to ensure consistency in results. Future tests will be conducted to evaluate the effect reflections from a person holding the phone would have.

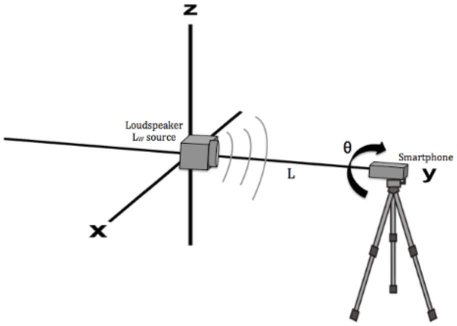


Fig 16. Orientation of smart phone with respect to loudspeaker.

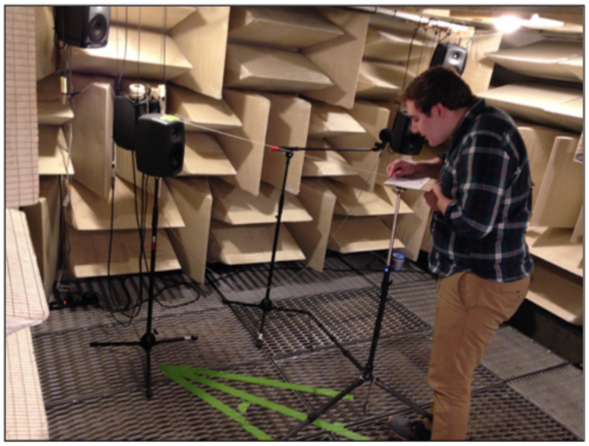


Fig 17. Altering the orientation of smarthphones

To create a polar plot of the measured directivity pattern all measurements are referenced to results from $\theta = 0^{\text{o}}$ incidence degree angle (Fig. 18). This plot displays the the four targeted frequencies of interest (125Hz, 500Hz, 1000Hz, and 2000Hz) extracted from pink noise signal. Results indicate that the phone is accurate to within 1dB at all frequencies regardless of orientation (within 180^{o}).

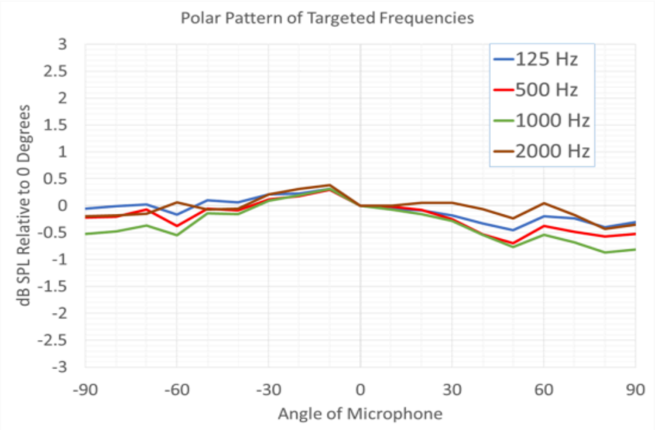


Fig 18. Polar pattern of app relative to $\theta = 0^{\circ}$

CONCLUSION & FUTURE WORK

This paper presents details of the development of an environmental noise monitoring platform harnessing the capabilities of smartphones for noise assessment. The platform is structured in two main parts: i) the remote client side (phone) and ii) the server. The server has the capability for enhanced analyses to be applied to any acoustic signal recorded by a mobile client. The server also serves as a central data repository and in the future will be developed to collate and present all measured data. Using this mass data source, it will be possible to incorporate crowd sourced measurements into the noise mapping process.

A separate research question that must be addressed is whether of not it is possible to use a smartphone as a scientific tool? This paper attempts to answer this by determining the acoustic characteristics of a Samsung Galaxy S6 smartphone utilizing our developed app. The sensitivity, frequency response and directivity of the smartphone were assessed to determine if it is capable of measuring environmental noise.

Results indicate that the platform operated reliably within the dynamic and frequency ranges typically experienced in environmental noise studies. However, it appears the smartphones will be limited in any attempt to measure low frequency noise, and are therefore not appropriate for any assessment that may require C-Weighting. It may be possible to expand the operating ranges of a smartphone by developing a governing equation, specific to individual hardware, that could transform acquired data to ground truth conditions. More research is required to determine how acoustic characteristics, and as such the governing equations, vary from phone to phone.

As with any research project, during the course of testing we encountered several issues outside of our original project scope that need to be further investigated with future work. For example, the influence of a human holding the smartphone versus a tripod is something that has received limited attention in the academic literature. The next battery of tests for our app will consider this and determine the optimum position for holding the phone relative to the body. It will then be possible for the app itself to guide the user to hold the phone in an optimum position; for example, the app would guide the user to change the orientation, tilt, or position with respect to the body before undertaking a measurement.

Future work will also see a computational model of a typical MEMS microphone assembly being developed and used to apply correction factors to various configurations. The final addition would include the development of a method to remotely calibrate the system.

ACKNOWLEDGMENTS

The authors would like to the Paul S Veneklasen Research Foundation for their support of this project under *PSVRF Grant Request Number 16.05 Development of a Mobile App for Citizen Science Noise Assessment.*

We also wish to acknowledge Blaise Peschel, Colin Worrich, Colton Cote, Benjamin Steers and Jeffrey Stone, students of the University of Hartford, who provided research assistance for lab experiments and the development of the overall platform.

REFERENCES

[1] Pew Research Center (2014) “Cell Phone and Smartphone Ownership Demographics”,http://www.pewinternet.org/data-trend/mobile/cell-phone-and-smartphone-ownershipdemographics/

[2] Kanjo, E. (2010) “NoiseSPY: A real-time mobile phone platform for urban noise monitoring and mapping”, *Mobile Networks Applications*, 15, 562-574.

[3] D’Hondt, E., Stevens, M., Jacobs, A. (2103) “Participatory noise mapping works! An evaluation of participatory sensing as an alternative to standard techniques for environmental monitoring”, *Pervasive and Mobile Computing*, 9, 681-694.

[4] Murphy E., King E.A. (2015) “Testing the accuracy of smartphones and associated applications for measuring environmental noise” *Applied Acoustics*, Volume 106, May 2016, Pages 16-22

[5] ISO 3745:2003 Acoustics - Determination of sound power levels of noise sources using sound pressure - Precision methods for anechoic and hemi-anechoic rooms.

[6] Sandberg, U., 2001. Tyre/road noise – myths and realities. Plenary paper published in the, In: *Proceedings of the 2001 International Congress and Exhibition on Noise Control Engineering*, The Hague, The Netherlands.

[7] STMicroelectronics, 2014, “Tutorial for MEMS Microphones.”, ST, Geneva Switzerland.

[8] Murphy, E.A. King, Environmental Noise Pollution: Noise Mapping, Public Health and Policy, Elsevier, May 2014