User’s Guide:
A Low-Latency Noise Reduction Smartphone App for Hearing Improvement Studies

A. SEHGAL, A. BHATTACHARYA, T. CHOWDHURY, N. KEHTARNAVAZ
UNIVERSITY OF TEXAS AT DALLAS

MARCH 2017

This work was supported by the National Institute of the Deafness and Other Communication Disorders (NIDCD) of the National Institutes of Health (NIH) under the award number 1R01DC015430-01. The content is solely the responsibility of the authors and does not necessarily represent the official views of the NIH.
Table of Contents

SECTION 1: APP SCOPE 3

SECTION 2: APP DEVELOPMENT PLATFORM 4

SECTION 3: OPERATING SYSTEM 5
  3.1 ANDROID 5
  3.2 iOS 7

SECTION 4: APP CODE ORGANIZATION 8
  4.1 ANDROID 8
  4.2 iOS 10

SECTION 5: RUNNING THE APP ON SMARTPHONES 12
  5.1 ANDROID 12
  5.2 iOS 13

SECTION 6: USER OPTIONS 14
  6.1 OPTIONS 14
  6.2 TEXT VIEWS 15
  6.3 USER INTERFACE 16

SECTION 7: RUNNING THE APP WITH HEARING AIDS 17

SECTION 8: REFERENCES 19
Section 1: App Scope

This user’s guide covers how to run and use a developed noise reduction smartphone app capable of interacting with hearing aids equipped with a low-latency Bluetooth wireless link. The app is also designed not to suffer from musical noise artifact that appears in many noise reduction algorithms. This app provides a real-time implementation of two noise reduction algorithms that were discussed and reported in [1] - [3]. The interested reader is referred to these references for the details of the algorithms.

The primary purpose of this noise reduction app is to assist users of hearing aids in noisy environments, in particular in babble type of environments such as restaurants, malls, public gathering places, etc. The app runs on a smartphone placed in front of a speaker whose speech is of interest by picking up the speech signal of the speaker together with the background noise signal via its microphone. The noise component of the speech signal is reduced via an algorithm running on the ARM processor of the smartphone. The noise reduced speech signal is then transmitted to hearing aids via a low-latency Bluetooth link.
Section 2: App Development Platform

The noise reduction algorithms were initially coded in MATLAB and then converted to C by using the MATLAB Coder utility as described in Section 3 of the user’s guides [4] and [5]. The interfacing with the app is carried out using the shells discussed in these user’s guides.

Fig. 1 illustrates the app development flow.

Fig. 1 Development flow of the noise reduction app
Section 3: Operating System

3.1 Android

For Android smartphones and mobile platforms, the software development environment of Android Studio and Android NDK were utilized to develop this app as described in the references [6] and [7]. To gain low-latency, the Superpowered audio engine was utilized. Superpowered is designed for both Android and iOS operating systems. It can be downloaded from the link noted in [8] as a zip file named “SuperpoweredSDK.zip” and can be unzipped under the noise reduction app folder “SuperpoweredSDK\Examples_Android\”.

To use the app:

- Open Android Studio. A dialog box appears. Click on “Open an existing Android Studio project”.
- Go to the location of the app and select it to open.
- Initially, the console “Messages Gradle Sync” appears showing an error regarding the missing NDK. As shown in the user’s guide in [4], apply the correct NDK path, “C:\\Android\\ndk” in “local.properties”.

![Fig. 2 Android NDK installation error](image)

- Select “local.properties” from the tab “Gradle scripts” at the left and then at the right, give the full location of “SuperpoweredSDK\Superpowered” for “superpowered.dir”.

Fig. 2 Android NDK installation error
The app is developed for the Android platform 23 and build tools version 23.0.2. It is required to install this version for proper synchronization. This can be done by using the links noted below in Figs. 4 and 5.

Then, the Gradle script syncs the app environment and makes it ready to build.
3.2 iOS

The app version for iPhones or iOS mobile platforms was developed using the Xcode development tool. It is required to have an Apple ID to work with the app. Visit the Apple developer online link noted in [9] for more details.

To work with the app, keep the app project at a desired location in a Mac computer, open Xcode and click on “Open another project…” or double click on the “.xcodeproj” inside the project folder. This opens the project in Xcode which can be run on a connected iOS device.
Section 4: App Code Organization

4.1 Android

Go to the project tab drop down menu and select project. The folder organization of the app will appear as shown in Fig. 7.

Fig. 6 Selecting Android project folder

Fig. 7 Android app organization

This provides the project framework for the noise reduction app. The detailed folder description for a generic Android app can be accessed using the Android developer website [10].

- The folder “build” contains the build output.
- The folder “src/main” contains the required resources and codes.
- The subfolder “java” contains the main code written in Java for the Android app. It contains the app layout and the initial parameters.
- The subfolder “jni” contains the codes written in C/C++ and the required definitions for the Java codes to interact with the native codes as described in [11], i.e. the codes written in C/C++.

The converted C codes for the noise reduction algorithms written in MATLAB are placed inside the subfolder “jni”.
Fig. 8 Java native interface contents of the noise reduction app

- C-converted code from Matlab
- Handles audio processing, calls noise reduction functions
- Handles timer properties
- C-converted code from Matlab
4.2 iOS

This is the folder organization of the iOS version of the app (see Fig. 8).

- The file ViewController.m contains the code for the user interaction in the “objective C” language.
- The folder “Native Code” contains the C-converted codes of the noise reduction algorithms from MATLAB using the MATLAB Coder
Fig. 9 iOS app code organization
Section 5: Running the App on Smartphones

5.1 Android

The following steps need to be taken to run the app on Android smartphones:

- Enable the developer mode on the Android smartphone [12].
- Connect the Android smartphone to the computer using a USB cable.
- Allow access to the smartphone data (see Fig. 10a) and allow debugging (see Fig. 10b).
- Click on the run button from Android Studio and select the device (see Fig. 11). The app will be installed on the smartphone.

![Fig. 10 Selecting device to install app](image-url)
5.2 iOS

The following steps need to be taken to run the app on iOS smartphones:

- Enable the developer option for iPhones [13].
- Connect the iPhone to a Mac computer using a USB cable.
- Run the app from Xcode (command+R). The app will be installed onto the iPhone or the iOS device that is connected to the Mac.
Section 6: User Options

6.1 Options

Fig. 1 shows the app view both for the Android and iPhone smartphones. The following options are made available to users:

- **Sampling Frequency**: This indicates the sampling frequency required for sampling the incoming speech signal in real-time. The default is 48kHz, which is the preferred sampling frequency in most modern smartphones to have the lowest latency.

- **BufferSize (frame size)**: This indicates how many samples appear in a data frame. The preferred frame size for the lowest latency is dependent on the smartphone i/o hardware. The app is developed for a buffer or frame size of 512. The app would also run for values below this size.
Tolerable Noise SPL (Quiet Threshold): This option sets a tolerance level for the noise sound pressure. The default is 37dB which indicates the noise level above which the app is activated. Users have the option to change it (Android users by typing the value and iOS users by using a horizontal slide bar).

Noise Reduction Algorithm: This option allows users to select between the Wiener and LogMMSE algorithms. In general, as noted in [1], [2], the use of the LogMMSE algorithm could be more effective at lower SNRs compared to the Wiener noise reduction algorithm.

6.2 Text Views

There are two text views:

Frame time (for iOS, it is shown by ‘/’ beside frame size view): This time indicates the frame time according to the sampling frequency and the buffer or frame size.
For example, with the sampling frequency of 48kHz and frame size of 512, this time is $512 \times 1/48000 = 10.66$ milliseconds (ms).

- **Frame processing time:** This time indicates the processing time of the incoming speech signal per frame. This time should always be less than the frame time so no frame is skipped and a real-time throughput is achieved. The app runs about 4ms for the SPL set to 20dB with the frame size of 512 and the sampling frequency of 48kHz when the noise reduction entry is turned on.

6.3 User Interface

The app is easy to use and the interaction is done through the following three buttons:

- **Start button:** This button starts the app. It is recommended to set up all the conditions before starting the app.

- **Stop button:** This button stops the app. For changing any options or parameters after starting the app, it is recommended to stop it first and then run again with the modified options or parameters.

- **Noise reduction slide button:** When this button is turned on, the noise reduction algorithm is activated when the noise level exceeds the specified SPL.
Section 7: Running the App with Hearing Aids

The Starkey “Halo 2” hearing aid [14] can be used to interact with the app. This hearing aid provides a low latency Bluetooth link for iPhones. This app is intended to help users with hearing impairments to hear the noise reduced speech from a speaker whose speech is of interest or any other audio source. To connect the hearing aid to an iPhone, simply go to Setting->General->Accessibility (see Fig. 13).
Then, navigate to the option Hearing Devices and turn the Hearing Aid Mode on and also the Bluetooth option on.

Another supporting iOS app called TruLink [15] is required to disable the microphone of the hearing aid so that only the speech signal coming through the smartphone microphone can be heard.

At the time of this writing, there is no hearing aid with the same low-latency Bluetooth connection for the Android operating system. Once such a hearing aid becomes available, the Android version of the app can be used in a similar way.
Section 8: References


