Acoustic Calibration - Manual for MAPD

I. Mac Setup

1. Ensure that Xcode version 12.3 is installed on a Mac computer.
2. Open Xcode, go to **Preferences**, and under the **Accounts** tab, log in with the MAPD apple ID.
3. Ensure that Unity version 2018.4.15f1 is downloaded on the Mac computer
    a. Ensure that **Visual Studio** and **iOS** files are installed.
4. Open Unity, and open the MAPD project.
    a. Go to **Edit**, under the **Project settings** tab, change the **System Sample Rate to 50000**
5. Download the ***mls18.wav***, ***gap.wav***, and ***sync.wav*** onto the Mac computer and import them into the **Resources** section in the **Assets** tab in **Unity**.

II. Measuring the value of **cal0dB**

1. Set up the B&K sound level meter.
2. Under the MAPD folder, typically under /users/Desktop/MAPD, create a folder and name it **builds**
3. Open the MPAD unity file on the Mac computer.
4. When the Unity App loads, a sample screen will be displayed in the center on the screen with multiple menus available.
5. Under the **Scenes** section in the **Assets** tab near the bottom left, open the **CalibrationScene**
6. Under the **Scripts** section in the **Assets** tab near the bottom left, open the **iPadCali** script.
    a. Microsoft Visual Studio app will open with the **iPadCali** script.
7. In the ***iPadCali*** script, the variables **sync, gap,** and **mls** specify the actual audio file corresponding to each name.
    a. Ensure that the name of the audio file on **line 60** assigned to the variable **sync** is **“sync”**
    b. Ensure that the name of the audio file on **line 61** assigned to the variable **gap** is **“gap”**
    c. Ensure that the name of the audio file on **line 62** assigned to the variable **mls** is **“mls18”**
8. Set the **cal0dB** on **line 71** variable to **0**
9. Set the **stimdBSPL** on **line 72** variable to 60dB
10. After ensuring that each variable is properly updated, save the **iPadCali** file.
11. Go back to the **Unity** app, go to **File,** open **Build Settings.**
    a. Ensure that iOS is selected.
    b. Click **Build and Run**

12. When the pop-up window appears, name the file “**cal0dBTesting**”and save it in the **Builds** folder and click **save.**
13. Connect the iPad to the Mac using a **USB cable**
14. Wait for the computer to load the Xcode application.
15. When Xcode opens:
    a. Double click on “**Unity-iPhone**” on the top left
    b. Click “**Signing & Capabilities**” located in the center of the screen
        i. Check the box next to **Automatically manage signing**
        ii. When the pop-up window appears, click on **Enable Automatic**
        iii. Under the **Team scroll down menu,** select **Sravana Nuti(Personal Team) option**
    c. After this, press the run button in Xcode.
        i. This process will take time. Watch the progress on the bar on the top. Ensure to follow the instructions that might appear during the uploading process
16. Once the app has been uploaded on the iPad, disconnect the **USB cable** attaching the Mac and iPad.
17. Open iPad **Settings**, click on **General**, open **Device Management**, and click **Trust** to allow the MAPD app to run on the iPad.
18. Connect the **Sennheiser headphones** to the iPad using the appropriate jack.
19. Turn on the **B&K sound level meter**
20. In the sound level booth, place the headphones on a flat surface.
21. Ensure that the iPad is being played at max volume. Press the buttons on the side to adjust volume to the maximum level.
22. Place the sound level meter on the headphones and press **Play** on the iPad.
    a. Ensure that the pointed end of the sound level meter is on the headphones.
23. Hit **Record** on the sound level meter.
24. The value being displayed on the Sound level meter is the **cal0dB** used later.

<u>III. Directory Setup on the Desktop</u>

1. Turn on the desktop that is connected to the DAQ.
2. Download the ***iPadCalibration.py*** and ***daq.py*** files.
3. Ensure that in the same directory, **DAQ*.py*** file is present.
4. Download the ***iPadAnalyzeCalibration.m*** file.
5. Download ***reweightN.m, reweightSpeakerEdited.m, inverseFiltering.m*** files.
6. Ensure that all of these files are present in the same directory.
7. Create a folder in the same directory to store the calibration files. Name this folder as **data.**
8. Create a folder in the same directory to store the plots. Name this folder as **plots.**

IV. Changing dB SPL on the iPad

1. Open the MPAD unity file on the Mac computer
2. When the Unity App loads, a sample screen will be displayed in the center on the screen with multiple menus available.
3. Under the **Scenes** section in the **Assets** tab near the bottom left, open the **CalibrationScene**
4. Under the **Scripts** section in the **Assets** tab near the bottom left, open the **iPadCali** script.
    a. Microsoft Visual Studio app will open with the **iPadCali** script.
5. In the ***iPadCali*** script, the variables **sync, gap,** and **mls** specify the actual audio file corresponding to each name.
    a. Ensure that the name of the audio file on **line 60** assigned to the variable **sync** is **"sync"**
    b. Ensure that the name of the audio file on **line 61** assigned to the variable **gap** is **"gap"**
    c. Ensure that the name of the audio file on **line 62** assigned to the variable **mls** is **"mls18"**
6. The **cal0dB** variable is the sound that was originally measured to calibrate the iPad.
    a. Ensure that the variable is set to the value that was initially measured in **Section II** using the sound level meter.
7. The **stimdBSPL** variable specifies the sound level at which the iPad will play the sound at. This calibration file will calibrate the sound file to this **stimdBSPL.**
    a. Set the **stimdBSPL value to 10** initially.
    b. Ensure that this variable is updated every time a new dB SPL is tested.
    c. NOTE: This is the only variable that will be constantly changed to reflect each dB SPL. The range is 10dB SPL - 85 dB SPL with increments of 5dB SPL.
8. After ensuring that each variable is properly updated, save the **iPadCali** file.
9. Go back to the **Unity** app, go to **File,** open **Build Settings.**
    a. Ensure that iOS is selected.
    b. Click **Build and Run**
10. When the pop-up window appears, name the file appropriately and save it in the **Builds** folder and click **save.**
    a. **Name** each **build** with the recording date and the dB SPL value.
11. Connect the iPad to the Mac using a **USB cable**
12. Wait for the computer to load the Xcode application.
13. When Xcode opens:
    a. Double click on "**Unity-iPhone**" on the top left
    b. Click "**Signing & Capabilities**" located in the center of the screen
        i. Check the box next to **Automatically manage signing**
        ii. When the pop-up window appears, click on **Enable Automatic**

    iii. Under the **Team scroll down menu,** select **Sravana Nuti(Personal Team) option**
  c. After this, press the run button in Xcode.
    i. This process will take time. Watch the progress on the bar on the top. Ensure to follow the instructions that might appear during the uploading process
14. Once the app has been uploaded on the iPad, disconnect the **USB cable** attaching the Mac and iPad.
15. This will upload the MAPD file with the updated dB SPL value.

V. Sound Booth Setup

1. Ensure that KEMAR is properly placed in the sound booth and is resting securely on a flat surface.

Figure 1: Picture of KEMAR. It is securely placed on a chair.

2. Connect the **G.R.A.S 4 Channel CCP Power Module** to the outlet and turn on the Power Module.

Figure 2: Picture of the Power Module. Connect the USB power adapter to its corresponding port on the back and turn the power on.

3. Attach cables to **KEMAR** to the **left port** and **right port.**

Figure 3: Picture of the back of KEMAR. Attach the **cable marked R** to the **Right Ear CCP** Port and the **unmarked cable** to the **Left Ear CCP** port.

4. Attach the cables from KEMAR to the **Power Module** to **input channel 1** and **input channel 2.**

Figure 6: Picture of the back side of the Power Module. Attach the cable **marked R to input channel 1** and the **unmarked cable to input channel 2**.

5. Connect the **National Instruments DAQ** to a power outlet. Ensure that the cables are properly attached to the wires.

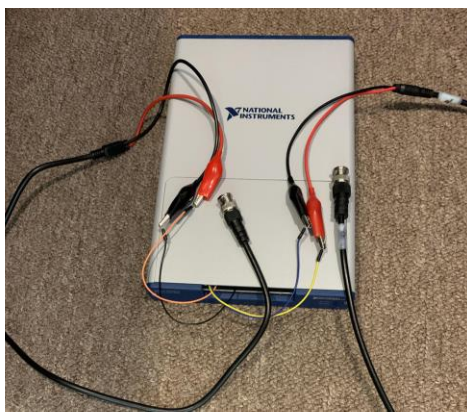

Figure 5: Picture of the National Instruments DAQ. Connect the **positive clip** of the **cable marked R to the yellow wire** and the **negative clip** to the **blue wire**. Connect the **positive clip** of the **unmarked cable** to the **orange wire** and the **negative clip** to the **black wire**.

6. Attach the cables from **DAQ** to the **Power Module** to **output channel 1** and **output channel 2.**

Figure 6: Picture of the back side of the **Power Module** with all four cables connected. Connect the **cable marked R from the DAQ to output channel 1**. Connect the **unmarked cable from the DAQ to output channel 2**.

7. Make sure that wires are properly connected and that there is no loose contact among the wires.

8. Connect the Scheneisser headphones to the iPad using a USB-C dongle. Place the headphones on KEMAR.

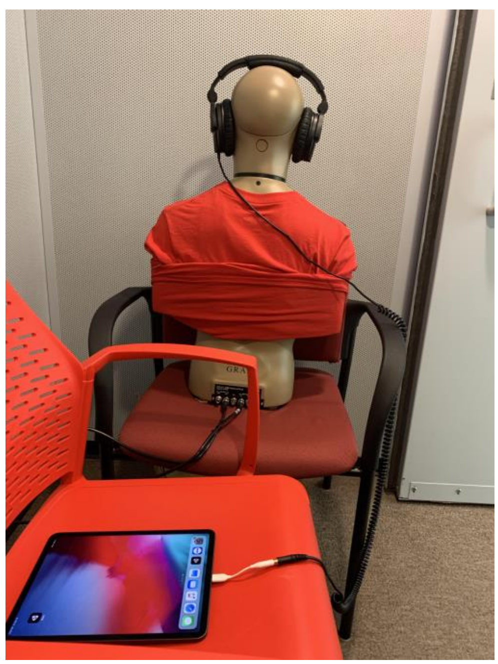

Figure 7: Picture of iPad - Headphones placement. Ensure that the headphones are centered over KEMAR's ear canals.

9. Place the KEMAR, Power Module, DAQ, and the iPad close by. Ensure that the iPad is not placed on the ground.

Figure 8: Picture of the final setup inside the sound booth. The setup consists of KEMAR, DAQ, Power Module, iPad, and headphones.

10. Ensure that the iPad is playing at maximum volume by using the volume buttons on the iPad to set the volume to max.
11. Open the **MAPD App** on the iPad. Press on the **Calibration** option. Press **Play**.
12. Ensure that the lights are turned off in the sound booth. Lights create a background hum.
13. Ensure that the monitor is turned off in the sound booth. The monitor also would otherwise generate background hum.
14. When exiting the booth ensure that both the doors are firmly shut.

VI. iPadCalibration.py File

1. Open the ***iPadCalibration.py*** file using **SPYDER**.
2. The **cal0dB** variable is the sound that was originally measured to calibrate the iPad.
    a. Ensure that the variable is set to the value that was initially measured in **Section II** using the sound level meter.
    b. This is the same value that is being used to calibrate the sound files in Unity as well.
3. The **dataFile** variable specifies the folder where the output of this python program must be saved and creates a file for each output calibration data.

   a. Ensure that the variable reflects the folder you created previously to store the calibration files in its path.
   b. The dataFile variable should be: **'data/calibration-%s-%.1fdbspl-%d' % (startTime, stimSPL, repeats)**
4. The **inpFs** variable specifies the input sampling frequency.
   a. Ensure that the variable is set to its **default value of 50,000 Hz.**
5. The **stimDur** variable specifies the duration of the MLS noise being played by the iPad.
   a. Ensure that the variable is set to its **default value of 5.944s**
   b. Ensure that this is the actual audio length in seconds of the MLS noise audio file being played by the iPad.
6. The **repeats** variable specifies the number of times the algorithm runs for.
   a. Ensure that the variable is set to its **default value of 100 times**
   b. This value can be changed as needed.
7. The **stimSPL** variable specifies the dB SPL value for that run. **NOTE:** This is the only variable that will be changed between each run.
   a. Ensure to change the variable every time a new sound intensity is played.
   b. Ensure that this value is the same as the value used to upload the updated MAPD app on the iPad.
8. After ensuring that the variables previously mentioned are accurate, press the run button.
9. After the program is finished running, a calibration file for the dB SPL defined will be saved in the **data** folder.

10. After the program is finished running, a plot will be outputted showing the time signal which was processed.

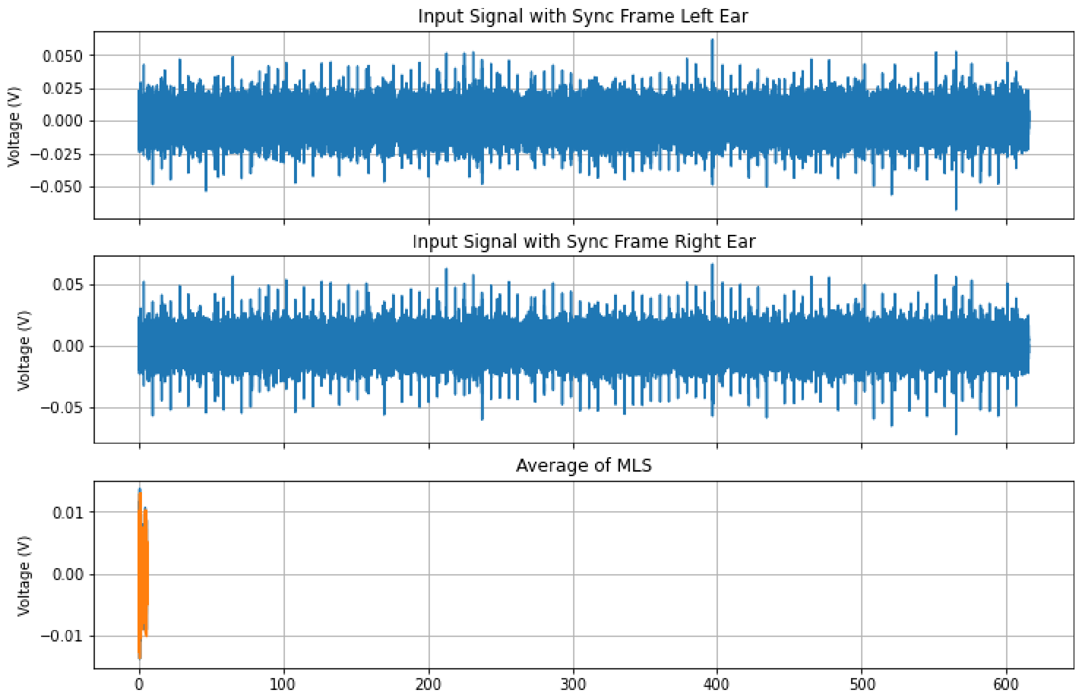

Figure 9: This is an example of the output plot of the ***iPadCalibration.py*** file. The plot above shows the signal collected from KEMAR for **10 dB SPL.** The top panel shows the Left Ear Input to the Python file. The middle panel shows the Right Ear Input to the python file. The bottom panel shows the processed signal, which is the output of this Python file. The signal represented in the bottom panel is saved in a separate **Calibration File** in the **data Folder.**

11. Repeat these steps for every single dB SPL being calibrated.

VII. iPadAnalyzeCalibration.m file

1. Open the ***iPadAnalyzeCalibration.m*** file.
2. The **dataPath** variable specifies where the calibration data is stored in the directory.
    a. Ensure that the variable reflects the folder where the calibration files are located.
3. The **plotPath** variable specifies where the calibration data is stored in the directory.
    a. Set this variable equal to **'plots'.** This way the resulting plots from this program will be stored in this folder.

4. The **files** variable specifies the individual calibration files that will be analyzed in this MATLAB program.
    a. Open your **data** folder.
    b. For each dB SPL, there will be two calibration files. One file as a **.py** and one file as a **.mat** file.
    c. Copy paste the name of the **.mat** file into the **files** variable.
5. On line 307, all the data is saved together in a single file called **impz**
    a. Ensure to change the **‘impz’** variable to reflect the dB SPL for that run.
    b. For example: if **10dB SPL** calibration file is being analyzed, change the variable name to **impz10 to reflect the 10dB SPL**
6. After ensuring that the variables previously mentioned are accurate, press the run button.
7. After the program is finished running, the **impz data structure** will be saved as the appropriately named file in the **impz folder.**
8. After the program is finished running, the program will output a figure containing four plots. These plots represent the amplitude spectrum and the phase spectrum.

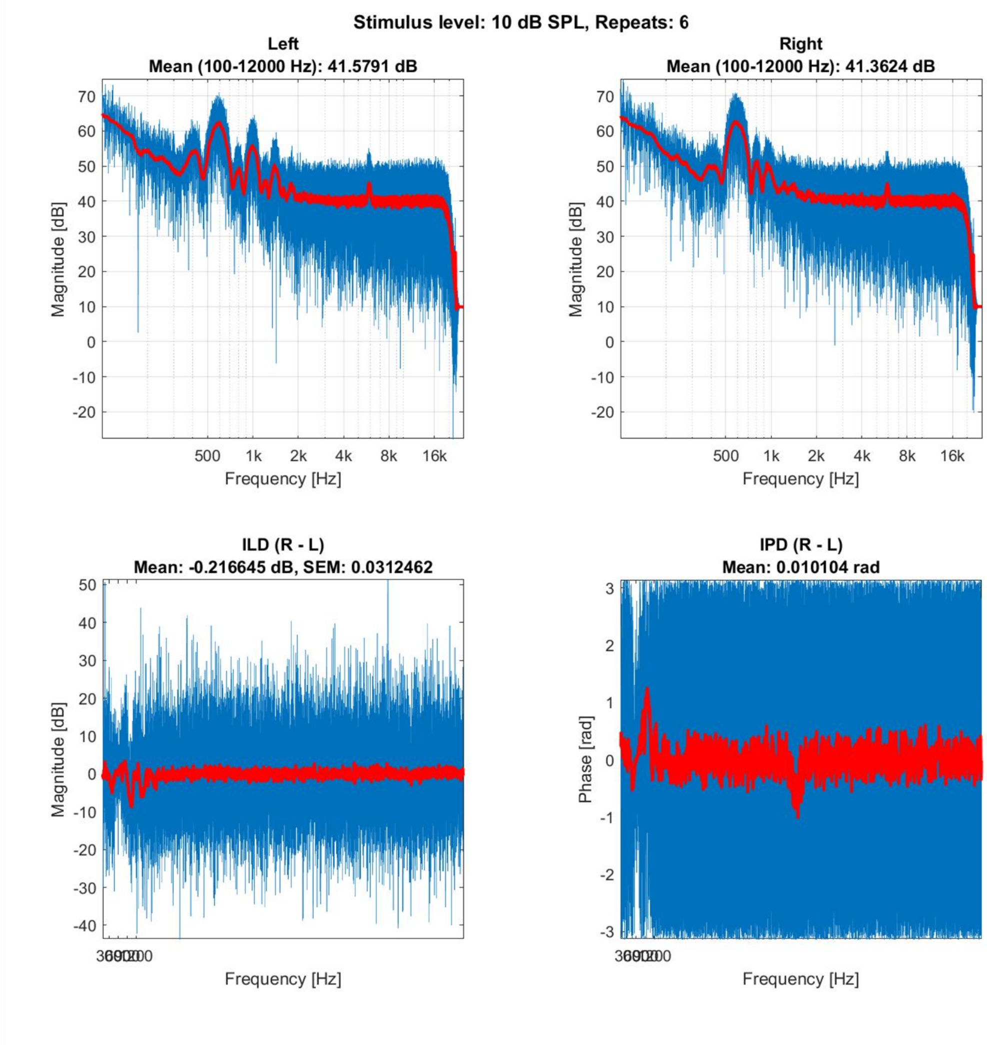

Figure 10: This is a sample output of the ***iPadAnalyzeCalibration.m*** file. The figure shows the amplitude spectrum and phase spectrum of the calibration file recorded for **10 dB SPL**. The two plots on the top are the frequency response of the unfiltered noise heard in the left ear and the right ear respectively. The frequency response is not flat as we expected.

9. Repeat these steps for every calibration file corresponding to each individual dB SPL.

<u>VIII. Inverse Filtering</u>

1. Open ***reweightN.m, reweightSpeakerEdited.m, inverseFiltering.m*** files.
2. In the ***reweightSpeakerEdited.m*** the variable **file** specifies the impz file that was created in the ***iPadAnalyzeCalibration.m*** file for each dB SPL.
    a. Ensure that this variable name is changed to reflect the proper **impz** file for each dB SPL.
    b. Ensure that this variable is updated for every dB SPL that is being reweighted.
3. In the ***inverseFiltering.m*** the variable **x2** calls the ***reweightSpeakerEdited*** function. The third parameter in the function reflects the dB SPL that is currently being reweighted.

   a. Ensure that this parameter is changed to reflect the current dB SPL being weighed.
   b. Ensure that this variable is updated for every dB SPL that is being reweighted.
4. The ***inverseFiltering.m*** also creates a **.mat** dataset for the reweighted sound for each dB SPL as well as a **.wav** file.
   a. Ensure that the names for each file being saved reflects properly for the dB SPL being reweighted for that run.
   b. Ensure that the name is updated for every dB SPL that is being reweighted.
5. After ensuring that the variables previously mentioned are accurate, press the run button in the ***inverseFiltering.m*** file.
6. The output of this program is the filtered noise for this particular sound level.
7. This program also outputs the graph of the transfer function.

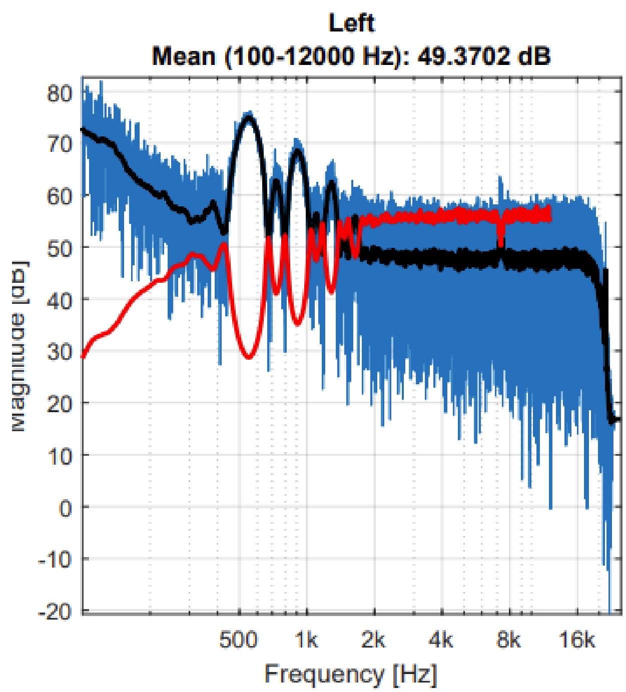

Figure 11: The black graph is the frequency response of the unfiltered sound heard in the left ear at **10 dB SPL.** The red graph shows the output of the ***inverseFiltering.m.*** The red graph is the transfer function of the inverse filter at **10 dB SPL.**

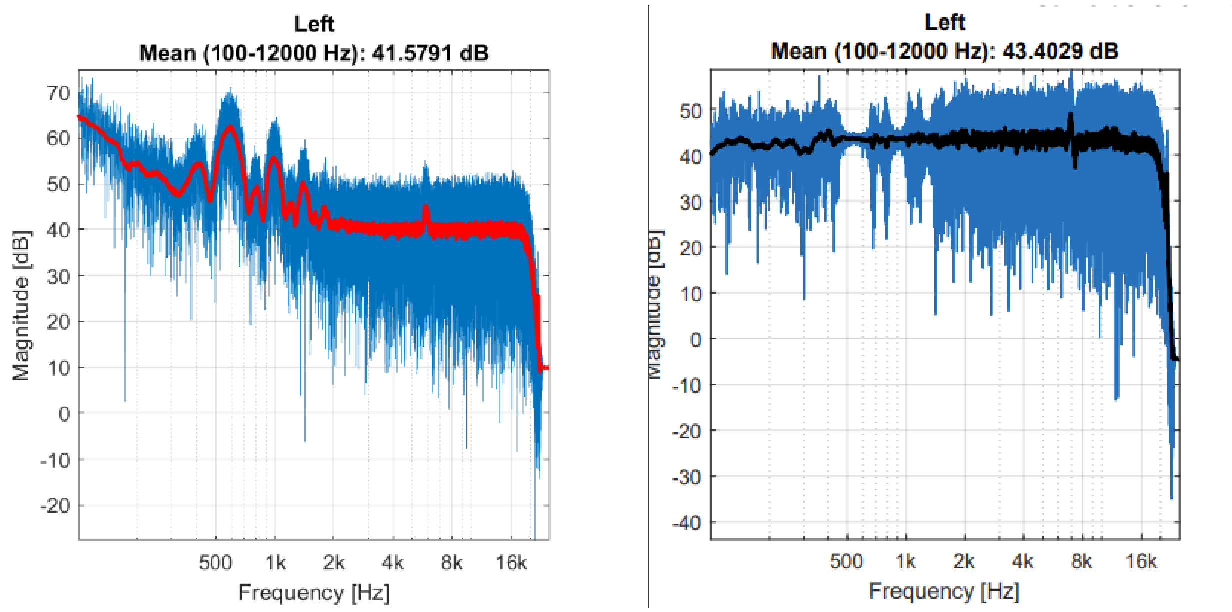

Figure 12: This is a before and after comparison of the frequency response for the left ear of the noise played at 10 dB SPL. The plot on the left shows the frequency response of the unfiltered sound. The frequency response is not flat as we would expect. The plot on the right shows the frequency response of the filtered sound. The filtered sound was achieved by inputting the unfiltered sound through the inverse filter. The frequency response is flat, which is the expected outcome of this calibration process. The goal of this calibration process is to achieve this kind of of flat frequency response for all sound intensities.

Repeat Sections IV - VIII for each dB SPL being measured. The dB SPL range is from 10dB SPL - 85dB SPL incrementing by 5dB SPL.