PROJECT SPONSOR NATION INSTITUTE OF HEALTH, NIDCD.

THIS WORK IS SUPPORTED BY THE NATIONAL INSTITUTE ON DEAFNESS AND OTHER COMMUNICATION DISORDERS, OF THE NATIONAL INSTITUTE OF HEALTH (NIDCD/NIH). R01DC015436: "A REAL-TIME, OPEN, PORTABLE, EXTENSIBLE SPEECH LAB" TO THE UNIVERSITY OF CALIFORNIA, SAN DIEGO.

HARINATH GARUDADRI

OPEN SPEECH PLATFORM USER GUIDE

OSP TEAM

Copyright © 2019 Harinath Garudadri

PUBLISHED BY OSP TEAM

OPENSPEECHPLATFORM.UCSD.EDU

Copyright 2017 Regents of the University of California

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

First printing, June 2019

Contents

Abstract 8

Setting up the RT-MHA 9

2.1 *Installing the Software* 9

Release 2019a Test and Validation 11

2.2 *Connecting Audio I/O Devices* 11

2.2.1 *On OS X machines* 11

2.2.2 *On Linux machines* 12

2.3 *Setting up Audio I/O* 13

2.3.1 *Internal Audio Codecs* 13

2.3.2 *External USB Audio Codecs* 14

Verifying the RT-MHA is Working Properly 16

3.1 *Internal Audio Codecs* 16

3.1.1 *Testing with audio files* 16

3.1.2 *Live Testing with headphones* 17

3.1.3 *Live Testing with Andrea SB-405 Super beam headphones* 17

RT-MHA Command line arguments 18

4.1 *OSP commands* 18

4.1.1 *-h for Help* 18

4.1.2 -F<option> for choosing Sampling frequency 18
4.1.3 -r to enable/ disable Rear mics 19
4.1.4 -a for enabling/disabling Adaptive Feedback cancellation (AFC) 19
4.1.5 -d<attenuation in dB> to set the attenuation factor 19
4.1.6 -T<option> for selecting AFC algorithm 19
4.1.7 -N<option> for choosing noise estimation type 20
4.1.8 -S<option> for enabling/disabling spectral subtraction 20
4.2 OSP modes of control 20
4.2.1 TCP daemon interface 20
4.2.2 Playback mode interface 21
4.2.3 Playback mode with TCP daemon interface 21
4.2.4 Command Line interface 21
4.3 Recommended Settings for ANSI 3.22 Testing 22
4.4 Recommended Settings for Testing with Webserver 22

Setting up the Embedded Web Server (EWS) 23
5.1 Running RT-MHA on Local Computer 24
5.2 Connecting to OSP Remotely 24

Verifying the EWS is Working 31
6.1 Researcher Page 31
6.2 4AFC 32

Bibliography 33

List of Figures

2.1 Terminal output after the `./install` command. You will need to enter your password at various times. *snip* represents a large portion of the terminal output deleted in this figure. This process can take 30 – 90 minutes, depending on your computer and network speed. The script ends with instructions on invoking osp and ews in the terminal. 10

2.2 USB Audio Input Configuration. 12
2.3 USB Audio Input Configuration. 12
2.4 List of devices displayed by running the **pa_devs** command 13
2.5 This is what your finder window will look like when you're looking up Audio MIDI Setup. 13
2.6 This is what the Audio MIDI Setup panel should look like when you select this device for sound output. 14
2.7 This is what the System Preferences>Sound should look like when you select this device for sound output. 14
2.8 This is what the System Preferences>Sound should look like when you select this device for sound input. 15

5.9 "This is what your Finder window should look like when you have selected your Hard Drive 25
5.10 Here are all the files you have made visible with this command, with the etc folder highlighted 25
5.11 Here are all the files you have made visible with this command, with the etc folder highlighted 26
5.12 This is how you Open the httpd.conf file using Atom 26
5.13 This is what it should look like after you delete the # on the appropriate line 27
5.14 This is what it should look like after you delete the # on the appropriate line 27
5.15 This is what your Terminal window will look like when you use the whoami command. The output is highlighted. Your output will be your username, not tzubatiy. 28

5.16 This is what your Atom window should look like when have added your name to the user line, in this case this is my name. 28
5.17 This is what your Atom window should look like when have added these lines of code in the right place. 28

Abstract

This version corresponds to Release 2019a. We present all of the necessary tools that need to be installed to make the Open Speech Platform (OSP) run smoothly, and how to make sure they are working properly. The OSP is comprised of a real time master hearing aid (RT-MHA) portion and an Embedded Web Server (EWS) portion. This guide is organized based on these two chapters, so first we present the tools to get the RT-MHA all set up, and then we present the setup process for the EWS.

`https://github.com/nihospr01/OpenSpeechPlatform-UCSD`.

Setting up the RT-MHA

2.1 Installing the Software

1. **Download 2018c**. Download the latest release from `https://github.com/nihospr01/OpenSpeechPlatform-UCSD`. You can either "clone the software on your computer" or "download ZIP file to your computer."

   (a) If you wish to clone, type `cd` in to your terminal, to make sure you are in the local directory. Then press clone in github, and copy the line of code generated into your terminal. Once it has cloned, type `cd nameofdirectory` (for example cd OpenSpeechPlatform-UCSD). Then, type `git pull` to get the latest version. This is the preferred approach.

   (b) When you download a .zip file from github, you will have to manually download newer versions with bug fixes.

2. **Install 2018c** Open a terminal. `cd ./Software/` and `./install`. This script does several things.

   (a) Identify the operating system (OS) on your computer – currently OS X and Linux.

   (b) Install software packages including portaudio (for realtime audio input/output); MySQL (a relational database), PHP (a server side scripting language) and other packages for LAMP software stack (LAMP stands for Linux, Apache, MySQL and PHP).

   (c) Build RT-MHA and EWS.

   (d) Finally, it installs osp in `/usr/local/bin/osp` and a script to invoke ews in `/usr/local/bin/ews`.

   If everything went well, your screen will look similar to Figure 2.1.

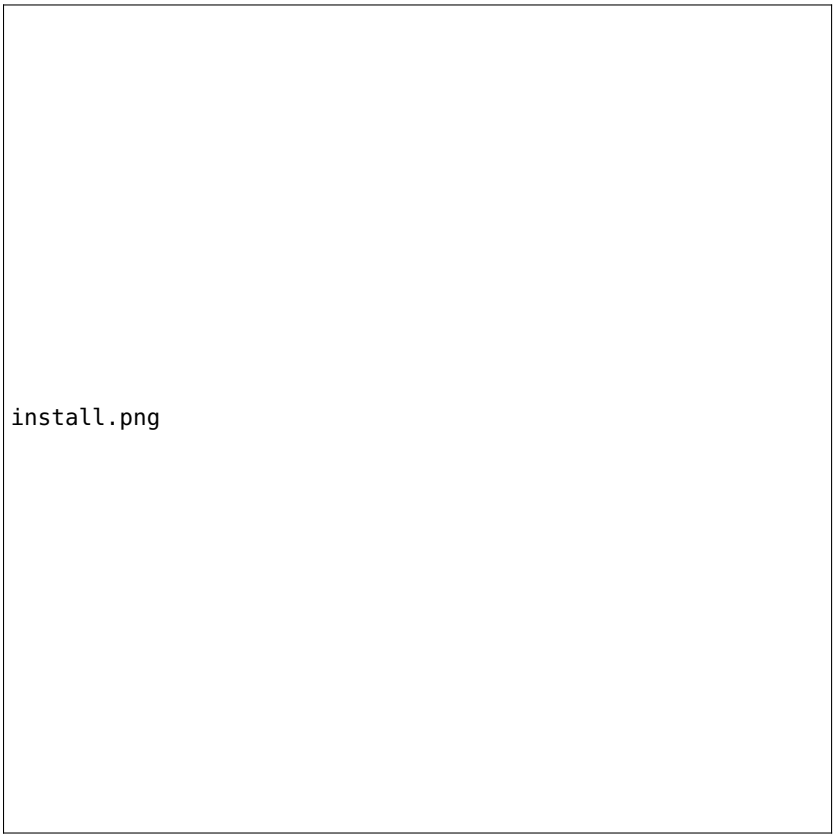

Figure 2.1: Terminal output after the `./install` command. You will need to enter your password at various times. *snip* represents a large portion of the terminal output deleted in this figure. This process can take $30 - 90$ minutes, depending on your computer and network speed. The script ends with instructions on invoking osp and ews in the terminal.

Release 2019a Test and Validation

This chapter describes sanity tests to validate your versions of `osp` and `ews`.

2.2 Connecting Audio I/O Devices

There are many audio input/output options for OSX and Linux computers. One inexpensive option is the Andrea Communications 3D Surround Sound Recording CANS. They are supra-aural headsets, with left and right mics.

You can also use a high end audio device such as Zoom TAC-8 or Focusrite Scarlett 2i2.

2.2.1 On OS X machines

1. Plug the headsets in to an available USB port.
2. On Mac computers, **Open Audio MIDI Setup**: This can be found in `Finder | Applications | Utilities`, as shown in Figures 2.2 and 2.3.
3. Select 48,000 Hz option on both screens and levels to 1..0 as shown.
4. If you use a high end audio interface box (such as Zoom TAC-8) and if you have 98,000 Hz and 24 bit option, you should choose this.

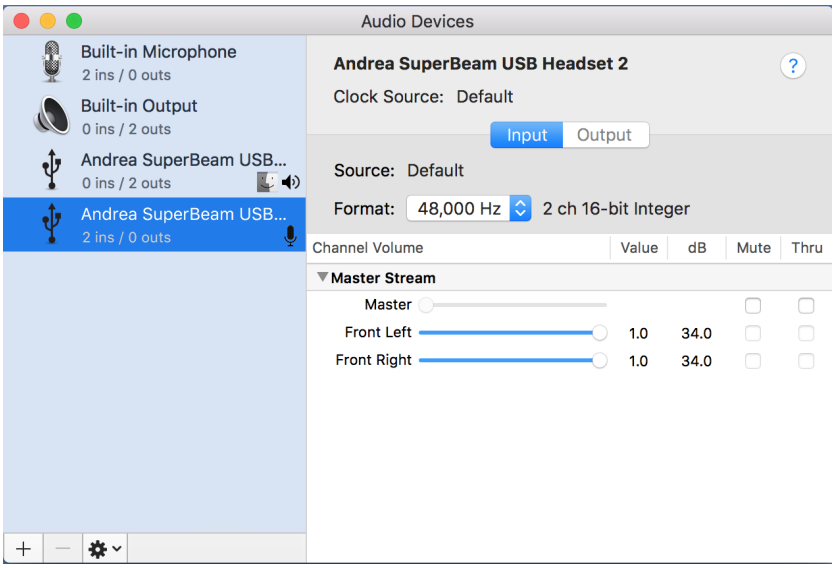

Figure 2.2: USB Audio Input Configuration.

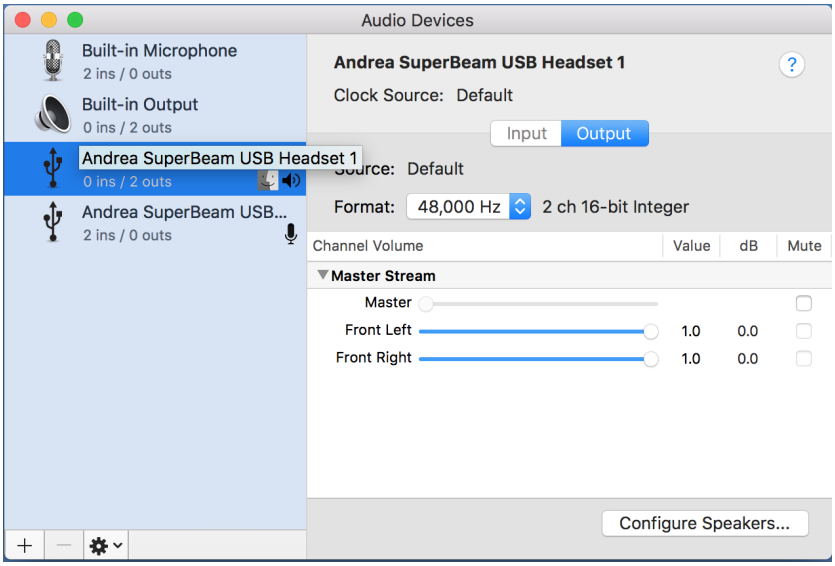

Figure 2.3: USB Audio Input Configuration.

2.2.2 *On Linux machines*

1. On Linux machines the default audio is not set. Run the command `pa_devs` to give the list of the available devices and the device numbers they correspond to

2. Run the command `osp -input_device x -output_device y`, where `x` and `y` are values attained from the previous command. With reference to Figure 2.4, the command issued is `osp -input_device 6 -output_device 6`

```
Default low input latency   =  -1.0000
Default low output latency  =   0.0058
Default high input latency  =  -1.0000
Default high output latency =   0.0348
Default sample rate         = 44100.00
Supported standard sample rates
 for half-duplex 16 bit 8 channel output =
       32000.00, 44100.00, 48000.00, 88200.00,
       96000.00, 192000.00
---------------------------------- device #6
Name                        = Andrea SuperBeam USB Headset: Audio (hw:1,0)
Host API                    = ALSA
Max inputs = 0, Max outputs = 2
Default low input latency   =  -1.0000
Default low output latency  =   0.0087
Default high input latency  =  -1.0000
Default high output latency =   0.0348
Default sample rate         = 44100.00
Supported standard sample rates
 for half-duplex 16 bit 2 channel output =
       44100.00, 48000.00
---------------------------------- device #7
Name                        = sysdefault
Host API                    = ALSA
```

Figure 2.4: List of devices displayed by running the **pa_devs** command

Can we use white background in the terminal shots for the document? Black is good for code development, but I am not sure for documents

2.3 Setting up Audio I/O

2.3.1 Internal Audio Codecs

It is possible to test proper functioning of RT-MHA using the internal audio codecs built into a MacBook.

1. **Open Audio MIDI Setup**: This can be found in Finder: Applications > Utilities, as shown in Figure 2.5.

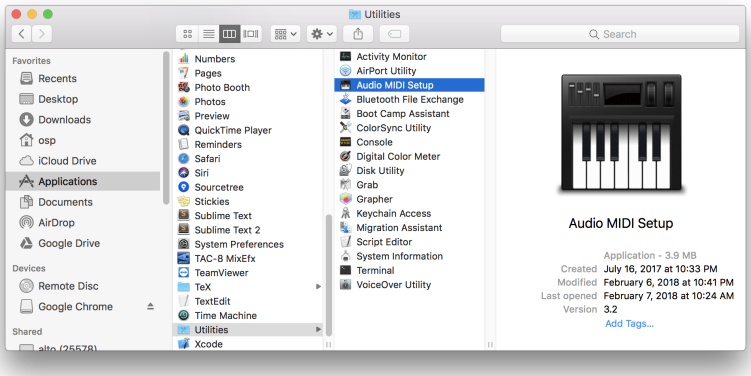

Figure 2.5: This is what your finder window will look like when you're looking up Audio MIDI Setup.

2. **Select primary input**: In Audio MIDI Setup, right-click (CTRL+click) Built-in Microphone on the left hand side. Select "Use This Device

For Sound Input. Change the format to 96,000 Hz."

2.3.2 *External USB Audio Codecs*

1. **Select primary output**: In Audio MIDI Setup, click on the Zoom Tac-8, click on Input, select the format 96,000 Hz. Then click Output, and also select 96,000 Hz."

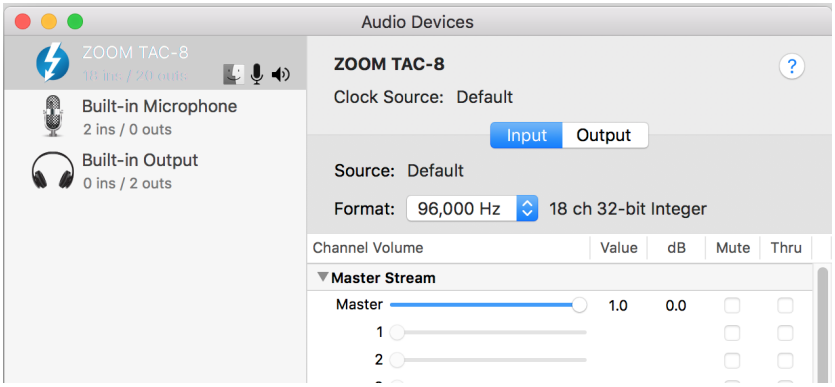

Figure 2.6: This is what the Audio MIDI Setup panel should look like when you select this device for sound output.

2. **Edit Sound Preferences**: Open System Preferences>Sound and select the Output tab. Click on Zoom Tac 8. Select the Input tab and also click on Zoom Tac 8.

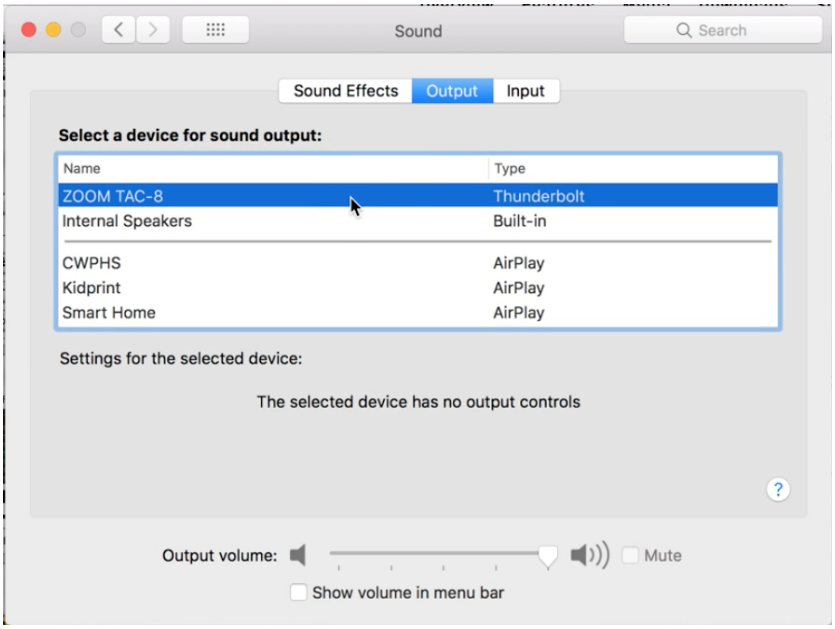

Figure 2.7: This is what the System Preferences>Sound should look like when you select this device for sound output.

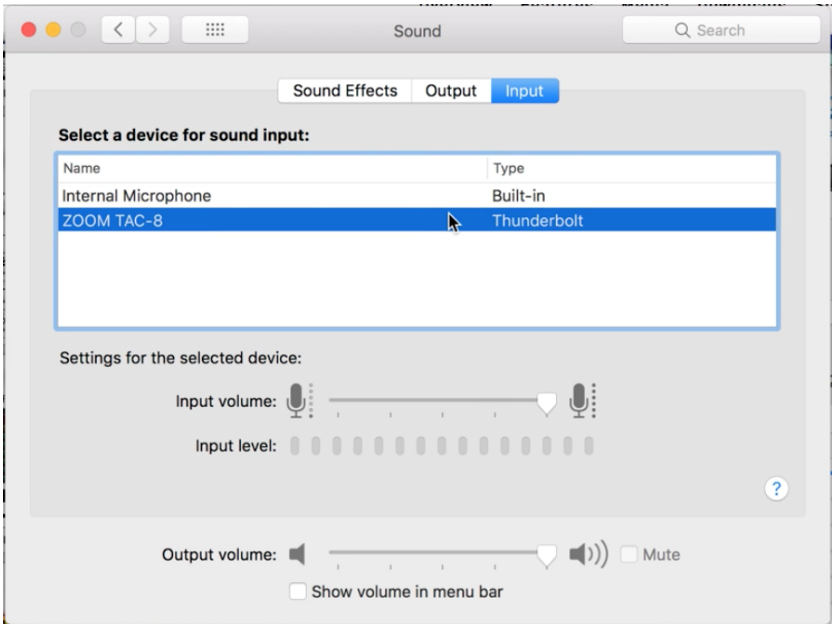

Figure 2.8: This is what the System Preferences>Sound should look like when you select this device for sound input.

Verifying the RT-MHA is Working Properly

3.1 Internal Audio Codecs

3.1.1 Testing with audio files

In this section, we will use audio samples provided with the OSP software to test the Real time Master Hearing Aid.

1. In OSP we have provided 96kHz sample speech files for testing the software. All of this is found in the following directory: OSP > Matlab > InputFiles-96kHz.

2. On the terminal run the following command to make OSP process audio files and play them through the speaker. Here we use the -p argument to play sound through the software.

```
osp -F1 -M1 -a -r -d-40 -p ~/Downloads/OSP/matlab/Input\ Files-32k/TIMIT_female_sa2.wav
```

   This would play the audio file "TIMIT_female_sa2.wav" through the entire OSP process, and you would hear the sound through the speaker/headphones.

3. We have also provided the option of playing the sounds with NH, N2, N4 and S2 settings. These can be executed using the following command:

```
osp -F1 -M1 -a -r -d-40 -p ~/Downloads/OSP/matlab/Input\ Files-32k/TIMIT_female_sa2.wav -s1
```

   Here the argument -s<setting_number> gives control to the user to choose the gain settings for OSP and play the audio with those settings. Setting_number are set the following way: 1= N2, 2 = N4, 3 = S2 and 0 = NH.

3.1.2 *Live Testing with headphones*

1. **For live testing** you need a headset with microphone. We found headphones commonly used for making calls to be adequate. Plug this into your computer.
2. **Make sure you have set the volume on your computer to the maximum allowed.**
3. Enter the following command on the terminal to start osp in command line interface mode.

```
osp -F1 -M1 -a -r
```

   Here choose F1 if your system supports 96kHz or else choose F0 for 48kHz.

4. Next you can increase the gain to 10dB on all bands by typing in "g10". This way your own voice would be heard louder through the headphones.

3.1.3 *Live Testing with Andrea SB-405 Super beam headphones*

1. **For live testing** you need a headset with microphone. We found headphones commonly used for making calls to be adequate. Plug this into your mac at the USB port.
2. Make sure the mute switch is in the "off" position.
3. **Make sure you have set the volume on your computer to the maximum allowed.**
4. Enter the following command on the terminal to start osp in command line interface mode.

```
osp -F1 -M1 -a -r
```

   Here choose F1 if your system supports 96kHz or else choose F0 for 48kHz.

5. Next you can increase the gain to 10dB on all bands by typing in "g10". This way your own voice would be heard louder through the headphones.

RT-MHA Command line arguments

In this section we explore the various functions of the RT-MHA on the command line interface. There are multiple starting configurations and these configurations are specified using arguments.

4.1 OSP commands

4.1.1 -h for Help

This would show all the available options/configurations the RT-MHA can be executed in, as shown Figure**??**.

```
osp -h
```

4.1.2 -F<option> for choosing Sampling frequency

The first thing you should do is choose your sampling frequency using the extension -F This should correspond with the sampling frequency you picked in step 1.11 and 1.12. The Sampling rates supported are 0, 48kHz and 1, 96kHz.

To set the sampling rate to 48kHz, type the command below into your terminal and press Enter:

```
osp -F0
```

Notice how in the terminal window it now says:"Choosing 48kHz sampling frequency"

To stop running OSP, type on the keyboard "control"+"c" do this now, and we will continue verifying that everything is working properly. You will need to do this step after every step in this section

To set the sampling rate to 96kHz, type the command below into your terminal and press Enter:

```
osp -F1
```

Notice how in the terminal window it now says:"Choosing 96kHz sampling frequency"

Type on the keyboard "control"+"c" to stop running OSP before going on to the next step.

4.1.3 *-r to enable/ disable Rear mics*

Open a Terminal window, type in the line below and press Enter

```
osp -r
```

This is the recommended default setting as we do not have rear mic support yet. Notice how in the terminal window, it says "Disabling rear mics".

Type on the keyboard "control"+"c" to stop running OSP before going on to the next step.

4.1.4 *-a for enabling/disabling Adaptive Feedback cancellation (AFC)*

In a Terminal window, type in the line below and press Enter:

```
osp -r -a
```

You are now running OSP from the command line with the rear mics disabled and AFC off. On the first line in the terminal window it now says: "4AFC has been disabled" **Type on the keyboard "control"+"c" to stop running OSP before going on to the next step.**

4.1.5 *-d<attenuation in dB> to set the attenuation factor*

To run OSP with an attenuation factor (let's say -40dB attenuation), type the line below into a terminal and press Enter:

```
osp -r -a -d -40
```

you can also not add a space between the -d and the -40 as such:

```
osp -r -a -d-40
```

Notice how in the terminal window it now says: " Attenuation factor is -40.000000 dB" **Type on the keyboard "control"+"c" to stop running OSP before going on to the next step.**

4.1.6 *-T<option> for selecting AFC algorithm*

To start OSP with a predefined AFC type use the extension -T

The types of AFC supported are '0' = FXLMS, '1' = PNLMS and '2' = SLMS. For example, say you want to set the setting to SLMS. Then you would type the line below into your terminal and press Enter:

```
osp -T2 -r
```

Notice how in the terminal window it now says: Setting adaptation type to SLMS **Type on the keyboard "control"+"c" to stop running OSP before going on to the next step.**

4.1.7 *-N<option> for choosing noise estimation type*

To run OSP with a predefined Noise Estimation algorithm, use the extension -N-noiseExtension

The types of Noise estimation algorithms RT-MHA supports are '1' = Arslan Power Averaging Procedure, '2' = Hirsch and Ehrlicher weight averaging procedure and '3' = Cohen and Berdugo MCRA Procedure.

For example, say you wanted to set the Noise Estimation algorithm to the Cohen and Berugo. To do this, type the line below into your temrminal and press Enter:

```
osp -N3 -a -r
```

Notice how in the terminal window it now says: Cohen and Berdugo MCRA Procedure Noise estimation method chosen **Type on the keyboard "control"+"c" to stop running OSP before going on to the next step.**

4.1.8 *-S<option> for enabling/disabling spectral subtraction*

To run OSP with spectral subtraction, use the extension -S1 or -S0 By setting spectral subtraction to '-S0', spectral subtraction is disabled; by setting to '-S1', spectral subtraction is enabled.

```
osp -S1 -a -r
```

Notice how in the terminal window it now says: Spectral subtraction enabled. **Type on the keyboard "control"+"c" to stop running OSP before going on to the next step.**

4.2 *OSP modes of control*

4.2.1 *TCP daemon interface*

-t starts the application in TCP daemon mode. In this mode, the application is waiting for a connection on port 8001 from an OSP client. Once a client is connected, all parameters (such as -g50 and -g80 gains, attack/release time, knee points, etc) come from the client.

The command line version of the application cannot be changed manually. To stop the application once it's started in TCP mode, the user must hit CTRL+C.

```
osp -t
```

Notice how in the terminal window it now says:"Using TCP daemon mode" and "waiting for a connection from client..."

4.2.2 *Playback mode interface*

-p<playback_file> starts the application in playback-only mode. The audio is not enabled, and instead, audio data is read from <playback_file> into the application where the RT-MHA algorithm is applied to the audio data, and this is played out in the hearing aid/ speakers/ headphones. This mode can be used for verifying the functionality of the software.

```
osp -p ~/Downloads/OSP/matlab/Input\ Files-32k/TIMIT_female_sa2.wav
```

Notice how in the terminal window it now says:"Playback mode specified, setting up file /Users/kc/Downloads/OSP/matlab/InputFiles-32k/TIMIT_female_sa2.wav"

4.2.3 *Playback mode with TCP daemon interface*

-l starts the application in TCP daemon and

4.2.4 *Command Line interface*

When neither of -t, -l or -p<wav file location> have not been specified. OSP can be interacted with on the command line. The following commands are can be used to interact with OSP on the command line (terminal).

1. 'g <gain_value>': If the operator chooses to adjust the gain to 10dB, he/she can simply enter 'g 10' followed by Enter on the command line.
2. 's' If the operator wants to turn on/off AFC, he/she can simply enter 's' followed by enter to toggle AFC mode.
3. 'p <spectral_subtraction_parameter>' If the operator chooses to adjust the spectral subtraction to 0.5, he/she can simply enter 'p 0.5' followed by Enter on the command line. The range for spectral subtraction parameter is from 0 to 1.
4. 'c' to save the converged afc filter coefficients.

5. 'f' to toggle feedback.
6. 'q' to quit the application.

4.3 Recommended Settings for ANSI 3.22 Testing

For ANSI 3.22 testType the following line of code into your Terminal and press Enter:

```
osp -a -r -M0 -d-40
```

Here the attenuation factor is -40. Please use -40dB when using with the breakout board (bob) and jellybeans.

4.4 Recommended Settings for Testing with Webserver

textFor testing with webserver, Type the following line of code into your Terminal and press Enter

```
osp -t -a -r -M1 -d-40
```

Setting up the Embedded Web Server (EWS)

1. **Enter the following line of code into your Terminal** to make sure you are located in your home directory:

```
cd
```

2. **Type the following line of code into your Terminal and press Enter** to navigate to the folder that the EWS is locate in:

```
cd downloads/osp/Embedded\ Webserver
```

3. **Type the following line of code into your Terminal and press Enter** to get Composer

```
brew install composer
```

4. **Type the following line of code into your Terminal and press Enter** to install Composer

```
composer install
```

5. **Type the following line of code into your Terminal and press Enter** to get NPM

```
brew install npm
```

6. **Type the following line of code into your Terminal and press Enter** to install PHP

```
brew install php
```

7. **Type the following line of code into your Terminal and press Enter**

```
cp .env.example .env
```

8. **Type the following line of code into your Terminal and press Enter**

```
php artisan key:generate
```

9. **Type the following line of code into your Terminal and press Enter** to install React

```
npm install react
```

10. **Type the following line of code into your Terminal and press Enter** to install React-dom

```
npm install react-dom
```

11. **Type the following line of code into your Terminal and press Enter** to install NPM

```
npm install
```

12. **Type the following line of code into your Terminal and press Enter** to run NPM

```
npm run dev
```

5.1 Running RT-MHA on Local Computer

1. **Enter the following line of code** In the same Terminal Window in which you ended the previous step (npm run dev), type in the following line of code and press Enter

```
php artisan serve
```

2. **Press the 'Run' Triangle** in the Upper left hand corner of your XCode window to run the OSP code

3. **Open a new Terminal Window** Type in the following line of code and press enter:

```
php artisan socket:start
```

5.2 Connecting to OSP Remotely

1. The following section is specifically for running OSP on a Mac and connecting to it remotely.

2. **The first thing that you should do is download a text editing software such as Atom or Sublime** To download, go to this link and press download: https://atom.io/

3. **When it finishes downloading double click on the downloaded item**, follow the instructions to move it into your Applications folder, and open a new window in Atom when it is finally loaded.

4. **Open a new Finder Window and double click on your Hard Drive** to see the contents stored there.

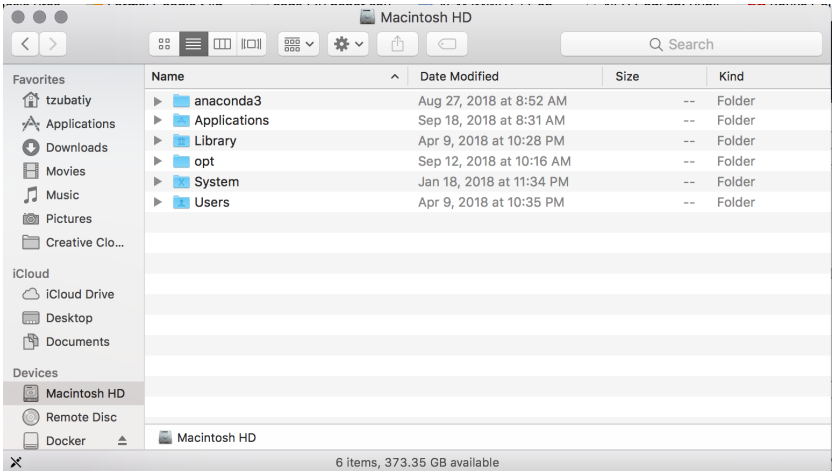

Figure 5.9: "This is what your Finder window should look like when you have selected your Hard Drive

5. **Press following three together: "command"+"shift"+">"** Once you do this, you should see some grayed out folders appear in the Finder window, that were not there before

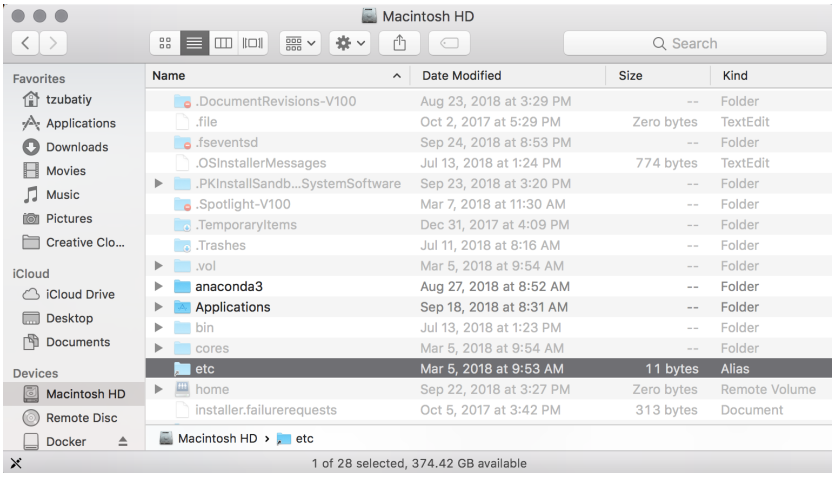

Figure 5.10: Here are all the files you have made visible with this command, with the etc folder highlighted

6. **Double click the grayed out folder "etc", then go into the folder titled "Apache", and find the file titled "httpd.conf"**

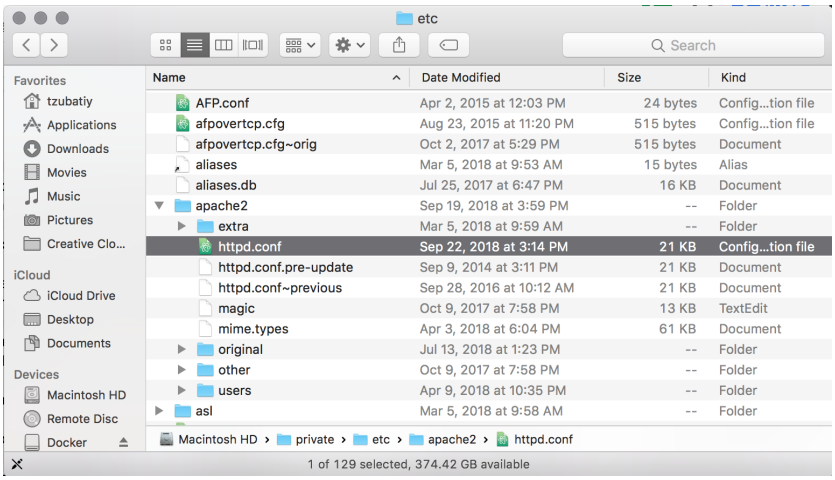

Figure 5.11: Here are all the files you have made visible with this command, with the etc folder highlighted

7. **Right click on this httpd.conf file and select Open With> Atom** To open the file in your text editor. When you do this, it will open the contents of this file for you to edit using Atom.

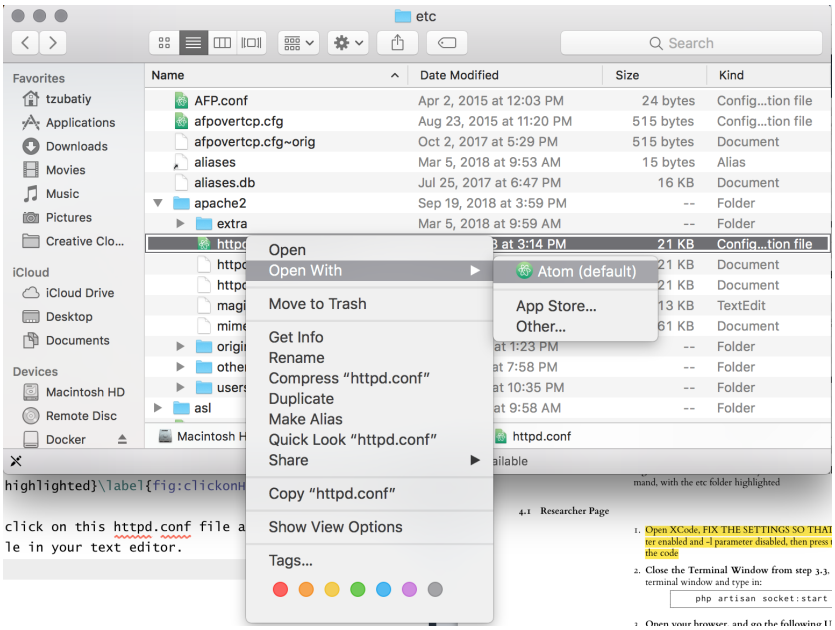

Figure 5.12: This is how you Open the httpd.conf file using Atom

8. **Once you’re looking at the file in Atom, and can see all of the text, press the following buttons together "Command"+"f" and then search for the text in the line below:**

```
#LoadModule php7_module libexec/apache2/libphp7.s
```

When you get to this line, **delete the hashtag (#) at the beginning of the line to complete this step**

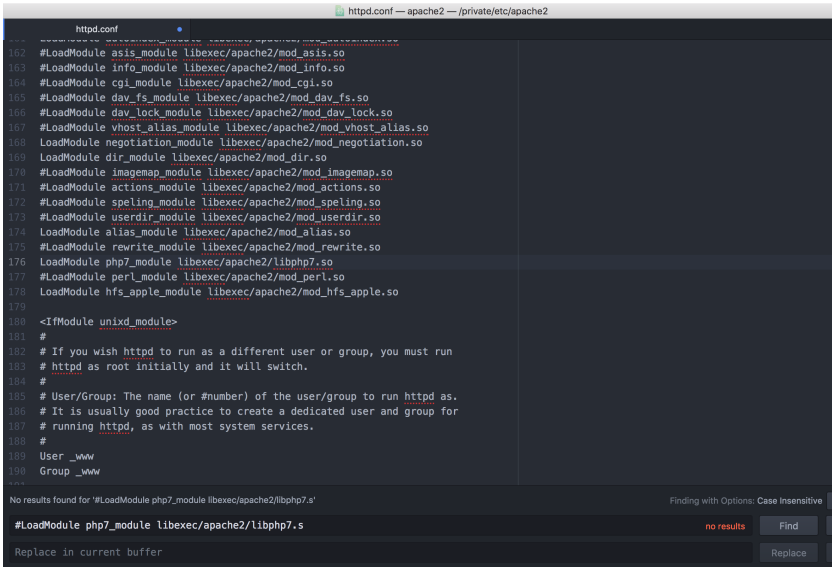

Figure 5.13: This is what it should look like after you delete the # on the appropriate line

9. **Press "Command"+"f" and then search for the text in the line below:**Once you find the line, **delete the hashtag (#) at the beginning of the line to complete this step**

```
#LoadModule rewrite_module libexec/apache2/mod_rewrite.so
```

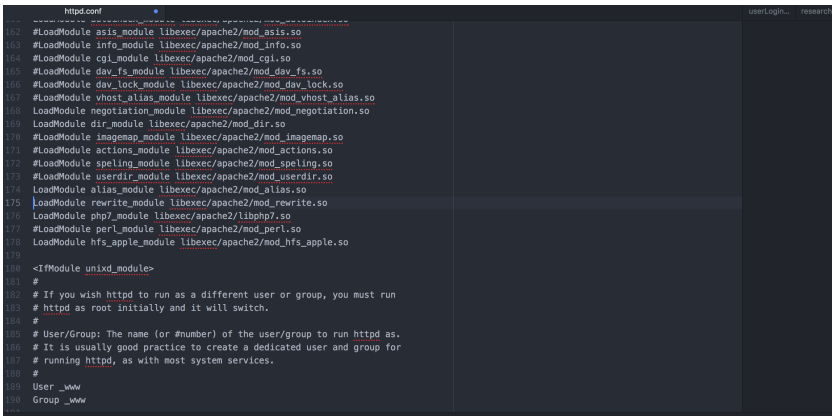

Figure 5.14: This is what it should look like after you delete the # on the appropriate line

10. **Open a new terminal window and type in the command below**
Copy or remember whatever the output of this command was

for the next step. For your reference in the images, my mine is tzubatiy.

```
whoami
```

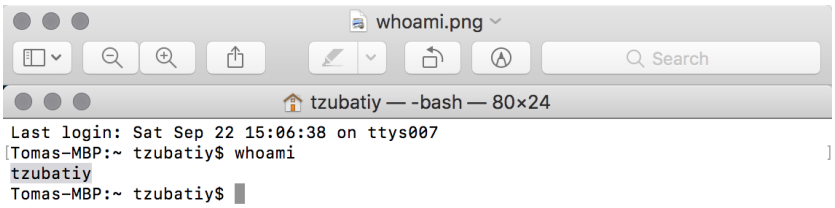

Figure 5.15: This is what your Terminal window will look like when you use the whoami command. The output is highlighted. Your output will be your username, not tzubatiy.

11. **Go back into your Atom window, press "command"+"f" and then search and then search for the text in the line below:** when you find it, **replace the _www with the ouptput of step 9**

```
User _www
```

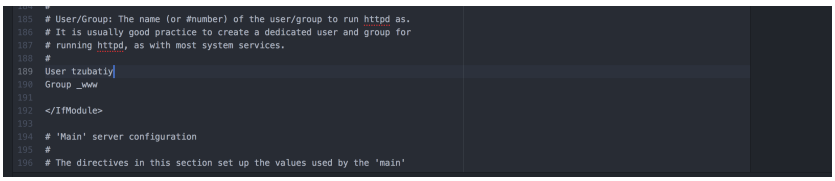

Figure 5.16: This is what your Atom window should look like when have added your name to the user line, in this case this is my name.

12. **On the line after "Group _www", copy in the following code:**

```
   <VirtualHost *:80>
      ServerName localhost
      DocumentRoot "home/downloads/OSP/embedded\ Webserver/public"
      <Directory "/home/linaro/OpenSpeechPlatform-UCSD/embeddedwebserver/public">
            AllowOverride All
Options Indexes MultiViews FollowSymLinks
      Require all granted
</Directory>
</VirtualHost>
```

Figure 5.17: This is what your Atom window should look like when have added these lines of code in the right place.

13. **In your Atom window press the buttons "Command"+"s"** to save the changes you have made, and close the file.

14. **Type the following line of code into a new Terminal window and press Enter:**

```
sudo apachectl restart
```

15. **Type the following line of code into a new Terminal window and press Enter:**

```
sudo chgrp -R www-data __PATH__/embeddedwebserver/
```

16. **Type the following line of code into a new Terminal window and press Enter:**

```
sudo chmod -R 775 __PATH__/embeddedwebserver/storage/
```

17. **Type the following line of code into a new Terminal window and press Enter:**

```
sudo chmod -R 775 __PATH__/embeddedwebserver/bootstrap/
```

18. **Type the following line of code into a new Terminal window and press Enter:**

```
sudo apachectl restart
```

19. **To connect to the OSP and control the RT-MHA REMOTELY, use the extension -t**

    This starts the application in TCP daemon mode. In this mode, the application is waiting for a connection on port 8001 from an OSP client. (See section 3.2) **Once a client connects, RT-MHA can be controlled remotely, such as the EWS, controlled using the web apps described in section 4.**

    To launch OSP in daemon mode, type the line below into your terminal and press Enter:

```
osp -t -T2 -d-40 -a -r
```

    Notice how in the terminal window it now says:"Using TCP in daemon mode" and "waiting for connection from client..."

20. **Open a new terminal window, copy in the following line of code and press Enter:**

```
cd
```

21. **Type the following line of code into the Terminal window from step 20 and press Enter:**

```
cd Downloads/OSP/Embedded\ Webserver
```

22. **Type the following line of code into this same Terminal window and press Enter:**

```
php artisan socket:start
```

23. **Confirm that you are connected to the same wireless network on this Mac, as on the device you are trying to connect with**

24. **Open a browser window on the device you are trying to connect with, and enter your IP address as the URL to start**

Verifying the EWS is Working

For verifying that the system is working properly, we recommend using Apple earbuds. Put them into your computer before adjusting volume in step 2 below.

6.1 Researcher Page

1. **If you still have OSP running, press "command"+"c" Otherwise, proceed to the next step**
2. **Open a new terminal window, copy in the following line of code and press Enter**

```
osp -t -T2 -d-40 -a -r
```

3. **Copy the following line of code in the same Terminal Window in which you ended the previous step, and press Enter**

```
php artisan serve
```

4. **Open a new Terminal Window, copy in the line below and press Enter:**

```
cd Downloads/OSP/Embedded\ Webserver
```

5. **Type in the following line of code into the Terminal window from the previous step (4.1.4) and press Enter:**

```
php artisan socket:start
```

6. **Open your browser, and go the following URL**

```
http://localhost:8000/researcherpage
```

7. **Increase the volume on your computer to the maximum**
8. **Add at least 5db to each cell in the the G65 row**

9. **Press transmit** and listen to the sound that plays
10. **Make another change, let's say removing those 5db and listen again**
11. **As soon as you have heard a difference between two parameter states, you can be sure that the software is working as intended.**

6.2 4AFC

1. **If you still have OSP running, press "command"+"c" Otherwise, proceed to the next step**
2. **Open a new terminal window, copy in the following line of code and press Enter**

```
osp -t -T2 -d-40 -a -r
```

3. **Copy the following line of code in the same Terminal Window in which you ended the previous step, and press Enter**

```
php artisan serve
```

4. **Open a new Terminal Window, copy in the line below and press Enter:**

```
cd Downloads/OSP/Embedded\ Webserver
```

5. **Type in the following line of code into the Terminal window from the previous step (4.2.4) and press Enter:**

```
php artisan socket:start
```

6. **Open your browser, and go the following URL**

```
http://localhost:8000/4afc
```

7. **Click on the four choices and the play symbol** and make sure you are hearing the appropriate word
8. **Once you hear the words playing on click**, you know that the 4AFC app is working properly

Bibliography

1. Harinath Garudadri, Arthur Boothroyd, Ching-Hua Lee, Swaroop Gadiyaram, Justyn Bell, Dhiman Sengupta, Sean Hamilton, Krishna Chaithanya Vastare, Rajesh Gupta, and Bhaskar D. Rao, "A Realtime, Open-Source Speech-Processing Platform for Research in Hearing Loss Compensation", In Signals, Systems and Computers, 2017 51st Asilomar Conference, IEEE, 2017.
2. James M. Kates, "Principles of digital dynamic-range compression," Trends in Amplification, vol. 9, no.2, pp. 45-76, 2005.
3. Shilpi Banerjee, The compression handbook: An overview of the characteristics and applications of compression amplification, 4th Edition, Starkey Laboratories, 2017.
4. James M. Kates, Digital hearing aids, Plural publishing, 2008.
5. Toon van Waterschoot and Marc Moonen, "Fifty years of acoustic feedback control: State of the art and future challenges," Proceedings of IEEE, vol. 99, no. 2, pp. 288-327, 2011.
6. Johan Hellgren, "Analysis of feedback cancellation in hearing aids with Filtered-x LMS and the direct method of closed loop identification," IEEE Transactions on Speech and Audio Processing, vol. 10, no. 2, pp. 119-131, 2002.
7. Hsiang-Feng Chi, Shawn X. Gao, Sigfrid D. Soli, and Abeer Alwan, "Band-limited feedback cancellation with a modified filtered-X LMS algorithm for hearing aids," Speech Communication, vol. 39, no. 1-2, pp. 147-161, 2003.
8. Donald L. Duttweiler, "Proportionate normalized least-mean-squares adaptation in echo cancelers," IEEE Transactions on Speech and Audio Processing, vol. 8, no. 5, pp. 508-518, 2000.
9. Jacob Benesty and Steven L. Gay, âĂIJAn improved PNLMS algorithm,âĂİ in Proceedings of IEEE International Conference on Acoustic, Speech, and Signal Processing (ICASSP), pp. 1881-1884, 2002.

10. Constantin Paleologu, Jacob Benesty, and Silviu CiochinÄČ, "An improved proportionate NLMS algorithm based on the lo norm," in Proceedings of IEEE International Conference on Acoustic, Speech, and Signal Processing (ICASSP), pp. 309-312, 2010.

11. Arslan, Levent, Alan McCree, and Vishu Viswanathan. "New methods for adaptive noise suppression." Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on. Vol. 1. IEEE, 1995.

12. Hirsch, Hans-GÃijnter, and Christoph Ehrlicher. "Noise estimation techniques for robust speech recognition." Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on. Vol. 1. IEEE, 1995.

13. Cohen, Israel, and Baruch Berdugo. "Noise estimation by minima controlled recursive averaging for robust speech enhancement." IEEE signal processing letters 9.1 (2002): 12-15.

14. Lee, Ching-Hua, Bhaskar D. Rao, and Harinath Garudadri. "Sparsity promoting LMS for adaptive feedback cancellation". In Signal Processing Conference (EUSIPCO), 2017 25th European, pp. 226-230. IEEE, 2017.