audiometry_trainer

Release 0.0.2

Samuele Carcagno

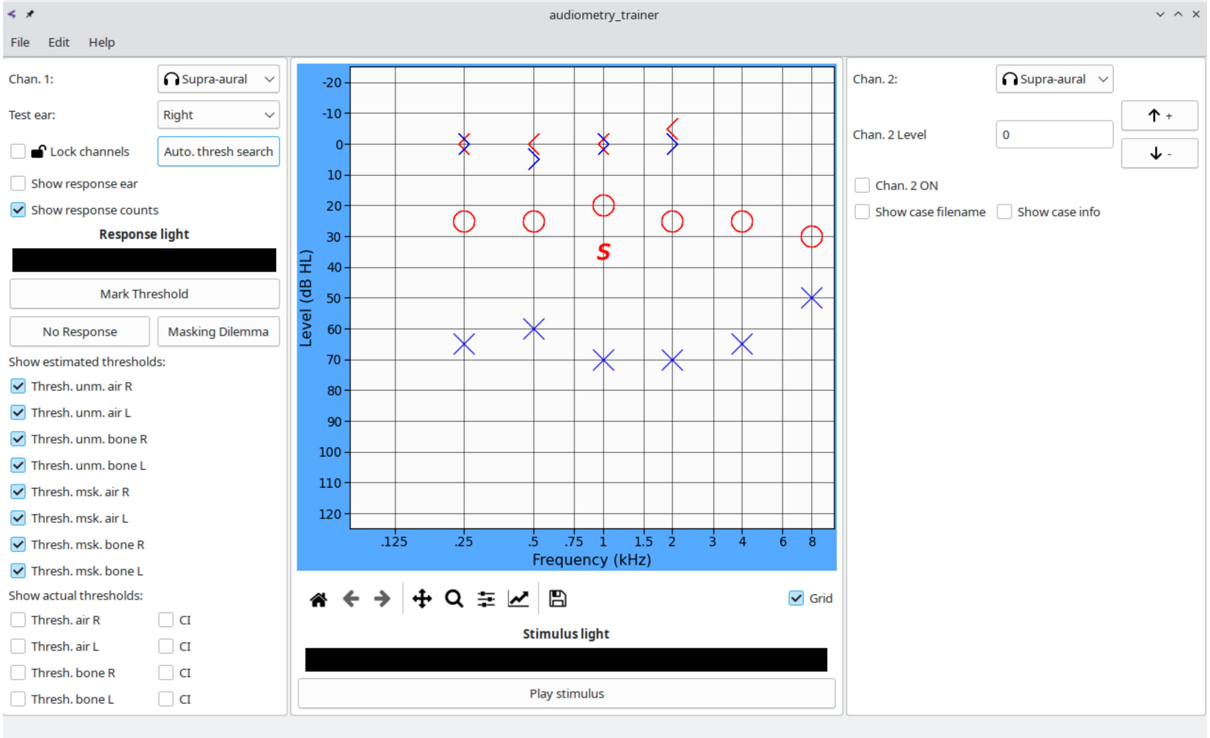
Apr 07, 2024

Contents:

CHAPTER 1

Introduction

`audiometry_trainer` is a software for practicing clinical audiometry. The software simulates patient's responses loaded from case files, and allows practicing essential aspects of the procedure including air/bone conduction threshold search and clinical masking. Figure *Screenshot of the audiometry_trainer interface* shows the main window of `audiometry_trainer`.



Chan. 1: Supra-aural
Test ear: Right
☐ Lock channels
☐ Show response ear
☑ Show response counts

Show estimated thresholds:
☑ Thresh. unm. air R
☑ Thresh. unm. air L
☑ Thresh. unm. bone R
☑ Thresh. unm. bone L
☑ Thresh. msk. air R
☑ Thresh. msk. air L
☑ Thresh. msk. bone R
☑ Thresh. msk. bone L

Show actual thresholds:
☐ Thresh. air R ☐ CI
☐ Thresh. air L ☐ CI
☐ Thresh. bone R ☐ CI
☐ Thresh. bone L ☐ CI

☑ Grid

Chan. 2: Supra-aural
Chan. 2 Level: 0
☐ Chan. 2 ON
☐ Show case filename
☐ Show case info

Fig. 1: Screenshot of the `audiometry_trainer` interface

At startup `audiometry_trainer` selects a random virtual patient case file (you can load other case files from the `File menu`). The S marker in the audiogram window shows the current stimulus level. The threshold search is more conveniently conducted using the keyboard rather than via mouse button presses. Press the space bar to play the stimulus (the `Stimulus light` at the bottom will light up). If the virtual patient heard the stimulus the `Response light` on the left side will light up. Use the up/down arrows to change the stimulus level. Once you've found the threshold, press the `T` key to mark it on the audiogram plot. Use the right/left arrow keys to move to another frequency.

You can compare the thresholds that you've estimated with the actual expected thresholds by using the `Show actual thresholds` checkboxes at the bottom of the left panel. Virtual patients are modeled via psychometric functions and the "actual" thresholds are computed via Monte Carlo simulations: they are the median thresholds obtained over a large number of simulations; 95% confidence intervals can be shown by checking the `CI` checkboxes.

You can move from right to left era and from air to bone conduction stimulation by selecting the desired options in the drop-down menus for `Channel 1` at the top of the left panel. `Channel 2` is used to deliver masking noise. To turn on `Channel 2` check the `Chan. 2 ON` check box on the right panel, a `M` marker on the audiogram window will indicate the masking noise level. You can use the `Chan. 2 level` box and the up/down arrows on its right to set the masker level. You can also `lock` the `Channel 1` and `Channel 2` levels using the `Lock Channels` checkbox on the left panel; when the channels are locked increasing/decreasing the stimulus level by a given amount will automatically change the masker level by the the same amount.

A detailed description of all the features of `audiometry_trainer` will be given in the next sections. Section *Installation* tells how to obtain and install `audiometry_trainer`. Sections *Threshold search* and *Masking* explain how to perform a threshold search and how to use masking, respectively.

`audiometry_trainer` uses psychometric functions to model virtual patient responses. Responses are not deterministic (always yes/no above/below a given stimulus level) but probabilistic, with the probability of a response increasing with the stimulus level. This adds realism to the software because in real life the responses of patients are not deterministic. Section *Virtual listener* details the psychometric function model used by `audiometry_trainer`.

`audiometry_trainer` allows you to create and use your own case files. This functionality can be accessed via the `Generate case` button under the `File menu` and will be described in detail in Section *Generating case files*. Some knowledge of the psychometric function model used by `audiometry_trainer` will be needed to create your own case files.

Section *User interface* covers in details all the user interface. Section *Internals* describes some of the internals of the software and is mainly intended for developers rather than for end users.

CHAPTER 2

Installation

audiometry_trainer has been successfully installed and used on Linux and Windows. It should also work on Mac platforms, but this has not been tested. A dedicated installer is provided for Windows. On Linux and Mac (and Windows if you do not wish to use the dedicated installer) audiometry_trainer, which is written in Python, can be installed via the Python package installer pip:

```
pip install audiometry_trainer
```

audiometry_trainer depends on a few Python modules including:

- PyQt5
- numpy
- scipy
- matplotlib
- pandas

depending on your Python distribution you may want to install these dependecies before installing audiometry_trainer via pip (e.g. through conda if you're using the Anaconda Python distribution, or through your Linux distribution package manager if you're using the Python installation that comes with your Linux distribution), otherwise pip will attempt to automatically pull in and install these dependencies. If the program is successfully installed you should be able to start it from a bash/DOS terminal with the command:

```
audiometry_trainer
```

You need to ensure that the Python environment you're using when you call the above command matches the one you used when you installed the application.

CHAPTER 3

Threshold search

The S marker in the audiogram window shows the current stimulus level. You can use the up/down arrow keys on the keyboard to change stimulus level and the right/left arrow keys to change stimulus frequency. Pressing the spacebar “plays” the stimulus to the virtual listener (the `Stimulus light` at the bottom will light up). If the virtual listener heard the stimulus the `Response light` on the left side will light up. You can also use the `Play stimulus` button to play the stimulus, but using the keyboard is generally more efficient.

On the left panels there are buttons to either mark the current level as the threshold, mark a no response or a masking dilemma. Keyboard shortcuts for these three actions are `T` to mark the threshold, `N` for no response, and `X` for masking dilemma. Pressing the `Delete` or the `D` keys will delete the currently marked response.

`audiometry_trainer` keeps counts of acending responses to assist in determining threshold according to the Hughson-Westlake procedure. If you do not wish to see these response counts uncheck the `Show response counts` checkbox.

If the `Show response ear` checkbox is checked the response light will be red if the response is coming from the right ear, blue if it is coming from the left ear, and white if both ears respond to the tone. This is useful to illustrate cross-hearing. This can also simulate the situation in which the audiologist asks the listener whether they’re hearing the sound in their left or right ear to ensure, for example, that masking is effectively silencing the non-test ear. This latter usage should not be abused while training to learn clinical audiometry because while `audiometry_trainer` will always indicate the correct answer regarding the response ear, in real life listeners may not always be reliable in indicating the response ear, therefore you should not rely on this.

The `Auto. thresh. search` button performs an automatic threshold search using the Hughson-Westlake procedure for all the frequencies using the currently selected transducer for channel 1. The automatic threshold search is available only for the measurement of unmasked thresholds. This feature was added because the most interesting aspect of audiometry is the measurement of masked thresholds (if needed), so one can quickly get the unmasked thresholds using the automatic thresholds to then move on to determine if masking is needed, and if so measure the masked thresholds.

The `Show estimated thresholds` checkboxes can be used to show/hide the thresholds that have been measured. By default all the measured thresholds are shown on the audiogram, but sometimes this generates clutter (e.g. when masked thresholds are obtained after unmasked ones) and it may be useful to hide some of the estimated thresholds.

To compare the thresholds that you've estimated with the actual expected thresholds by using the `Show actual thresholds` checkboxes at the bottom of the left panel. Virtual patients are modeled via psychometric functions (see Section *Virtual listener*) and the "actual" thresholds are computed via Monte Carlo simulations (see Section *Generating case files*), therefore, exact matches to the estimated thresholds should not always be expected, but they should generally be very close.They "actual" thresholds are the median thresholds obtained over a large number of simulations; 95% confidence intervals for these thresholds can be shown by checking the `CI` checkboxes.

CHAPTER 4

Masking

In order to mask the non-test ear (NTE) during air-conduction audiometry set the `Chan. 2 level` to the desired value and make sure the `Chan. 2 ON` checkbox is checked.

For masking during bone conduction audiometry, additionnally, the `Non-test ear status` has to be set to `Earphone on` (otherwise the software considers that the noise is being played through the earphone, but the earphone is not on the listener's ear, so no noise actually arrives at the listener's ear). `audiometry_trainer` automatically sets the `Non-test ear status` to `Earphone on` when you check the `Chan. 2 ON` checkbox. This small complication for masking in bone conduction audiometry is due to the fact that `audiometry_trainer` lets you test bone conduction thresholds either with the NTE uncovered, or with the NTE covered by the earphone while the earphone is not playing any noise. This is useful to measure the size of the occlusion effect (see *Measuring the occlusion effect*).

The `Lock channels` checkbox can be used to "lock" the test ear (TE) and NTE channels so that changing the signal level sent to the TE changes the noise level sent to the NTE by the same amount.

4.1 Interaural attenuation

Interaural attenuation (IA) values for supra-aural and insert earphones are drawn randomly for each case from a uniform distribution. The upper and lower limits of the distribution are frequency specific and are based on the max and min IA values reported by [MunroAndAgnew1999]. For frquencies that were not tested in [MunroAndAgnew1999] the values of an adjacent frequency are used.

IA values for bone conduction (BC) are drawn for each case from a uniform distribution. The lower and upper values for the uniform distribution as a function of frequency is shown in Table *table-IA_bone*. These values were chosen to reflect the observation that IA for bone conducted stimuli using a mastoid placement is essentially zero at 250 Hz and increases up to ~15 dB at 4000 Hz ([Studebaker1967]; [Gelfand2016]). More nuanced data on IA for BC are available [Stenfelt2012] and may be integrated to future versions of `audiometry_trainer`.

Table 1: Lower and upper limits of the uniform distribution for IA for bone conduction.

Freq.	Low	High
125 Hz	0	0
250 Hz	0	0
500 Hz	0	2
750 Hz	0	3
1000 Hz	0	4
1500 Hz	0	5
2000 Hz	0	7
3000 Hz	0	10
4000 Hz	0	15
6000 Hz	0	15
8000 Hz	0	15

4.2 Occlusion effect

For cases without a conductive hearing loss the size of the occlusion effect (OE) is drawn randomly for each case and frequency from a uniform distribution. The lower and upper values for the uniform distribution as a function of frequency is shown in Table *table-OE_supra* for supra-aural headphones and Table *table-OE_insert* for insert earphones. These values were chosen to reflect approximately the range of values found in various studies and summarized in [Gelfand2016].

Table 2: Lower and upper limits of the uniform distribution for the OE for supra-aural earphones. For frequencies > 1000 Hz the OE is always zero.

Freq.	Low	High
125 Hz	15	30
125 Hz	15	30
500 Hz	8	26
750 Hz	8	26
1000 Hz	4	12

Table 3: Lower and upper limits of the uniform distribution for the OE for insert earphones. For frequencies > 1000 Hz the OE is always zero.

Freq.	Low	High
125 Hz	2	10
125 Hz	2	10
500 Hz	2	10
750 Hz	2	10
1000 Hz	2	10

The OE is absent when a conductive/mixed hearing with an air-bone gap (ABG) >= 20 dB is present [MartinEtAl1974]. For this reason, if there is an ABG >= 20 dB the OE is set to zero. For ABGs between 0 and 20 dB the size of the OE is scaled by the ABG using the following equation:

$$OE_{out} = OE_{inp} - OE_{inp}(ABG/20) \tag{4.1}$$

where OE_{inp} is the OE before the scaling and OE_{out} is the OE after the scaling. This means that the size of the OE is progressively reduced as the ABG increases, reaching zero for an ABG of 20 dB.

4.2.1 Measuring the occlusion effect

Some authors (e.g. [Gelfand2016]) recommend measuring the OE in individual listeners to determine the value to be used in masking formulas. This can be done by covering the NTE with the earphone and measuring BC thresholds, without presenting any masking noise to the NTE. The thresholds obtained in this "earphones ON" condition can then be compared to those obtained without covering the NTE to estimate the magnitude of the occlusion effect [MartinEtAl1974]. To obtain BC thresholds with the NTE covered by an earphone, but without delivery of masking noise, in `audiometry_trainer` you need to set the `Non-test ear status` to `Earphones ON` and make sure that the `Chan. 2 ON` checkbox is uncheked. The size of the OE measured in this way will depend on the earphones used for `Chan. 2`, so make sure that you select the same earphone type that will be later used to deliver the masking noise.

A limitation of this technique is that in some cases it can underestimate the size of the OE ([FagelsonAndMartin1994]; [Yacullo1997]). This may happen in the case of a listener with NTE BC thresholds close to the lower output limits of the BC vibrator. For example, if the listener has a BC threshold of 0 dB HL and the BC vibrator has a lower output level limit of -20 dB HL, then OEs larger than 20 dB would be underestimated. This may also happen if the TE BC threshold is lower than the NTE BC threshold. Suppose for example that the TE BC threshold is 20 dB HL, the NTE BC threshold is 30 dB HL, and the NTE OE is 30 dB. The threshold recorded in the unoccluded condition will be 20 dB HL and when the NTE is occluded the threshold recorded will be 0 dB HL; the difference between the two, the estimated OE, will be 20 dB rather than the actual NTE OE of 30 dB. If masking is used only to confirm that the unmasked TE BC threshold is genuine, underestimating the OE will be of no consequence; the TE had the lowest threshold to start with, so undermasking will simply confirm, "in the wrong way", a correct decision that that threshold was coming from the TE. If the estimated OE is used for a full-blown masked threshold search, the masked TE BC threshold may be lower than the unmasked one because the NTE will be undermasked.

CHAPTER 5

Virtual listener

The responses of the virtual listener are determined by a psychometric function (PF) that represents the proportion of “Yes” (I heard a sound) responses as a function of stimulus level. A logistic function ([KussEtAl2005]; [KingdomAndPrins2016]) is used:

$$\psi(x; \alpha, \beta, \gamma, \lambda) = \gamma + (1 - \gamma - \lambda) \left(\frac{1}{1 + e^{\beta(\alpha - x)}} \right) \tag{5.1}$$

where α is the midpoint, β the slope, γ the lower asymptote, and λ the upper asymptote of the PF; x is the stimulus level at which the function is evaluated.

The midpoint is the stimulus level at which the PF reaches half of its amplitude, so for a virtual listener with $\gamma = 0$ and $\lambda = 0$, the midpoint α is the stimulus level at which the listener gives (on the long run) 50% of “Yes” responses. The Hughson-Westlake procedure declares the threshold as the lowest point at which the listener gives at least 50% “Yes” responses (over at least 3 trials). Therefore, the threshold (as defined with the Hughson-Westlake) procedure will be generally found at or above the midpoint α [MarshallAndJesteadt1986]. Obviously it is possible that on occasion the threshold will be found at a lower level because the proportion of “Yes” responses found over a small number of trials may differ from the proportion of correct responses expected over the long run.

The slope β controls the speed at which the PF goes from its lowest to its highest value as the stimulus level increases. Rather than reasoning in terms of slope, it is often more intuitive to reason in terms of “width” of the PF ([Alcalá-QuintanaAndGarcía-Pérez2004]; [KussEtAl2005]), which is inversely correlated with the slope. `audiometry_trainer` asks the user the desired 90% width of the PF in order to generate case files. For example, a width of 5 dB indicates that the function (for a virtual listener with $\gamma = 0$ and $\lambda = 0$) goes from 5% to 95% probability of a “Yes” response within a change in stimulus level of 5 dB.

The lower asymptote, γ, represents the propensity of the listener to respond “Yes”, even in the absence of a stimulus. This propensity is not formally measured in the clinical audiometry procedure [Barr-HamiltonEtAl1969] [MarshallAndJesteadt1986]. In a Yes/No task with temporally marked trials, in which a signal is presented or not, it would correspond to the false alarm rate. For example, a listener with a γ of 0.1 would respond “Yes” in 1 out of 10 trials in which no stimulus was presented. Under the perspective of signal detection theory this propensity of the listener to say “Yes” reflects an internal criterion which can be more or less conservative. Although the actual measurement of γ requires “blank” trials without a signal, the criterion determines the decision process also when a signal is present, hence the value of γ will affect the PF underlying the clinical audiometry procedure.

The upper asymptote, λ is the lapse rate, it represents the tendency of the listener to fail responding because of attentional lapses, even for a stimulus that is clearly audible.

When performing clinical masking with `audiometry_trainer`, the effect of the noise is to shift the midpoint of the PF, *if* the noise level arriving at the cochlea, in dB of effective masking (EM) units, exceeds the midpoint of the PF. The shift is equal to the level by which the noise exceeds the midpoint. For example, if the noise level is 5 dB above the unmasked midpoint, the masked midpoint will be 5 dB higher.

Because the effect of the noise varies between listeners, another variable `msk_diff` causes a further shift to the PF midpoint in the presence of noise. `msk_diff` is randomly drawn from a Normal distribution with a mean of 0 and a standard deviation of 3 for each frequency/ear combination when case files are generated. These values reflect the fact that interindividual differences in the effectiveness of the masking noise will be in the range of ~10 dB (this is the reason why a 10 dB safety pad is typically added to calculated noise levels in masking formulas). It should be noted that `msk_diff` can be either positive or negative. A positive `msk_diff` means that the masking effect is greater than average (the PF midpoint is shifted upward), while a negative `msk_diff` means that the masking effect is smaller than average (the PF midpoint is shifted downward).

CHAPTER 6

Generating case files

To generate a new case file click on the `File` menu and then on `Generate case`. The `generate case` window will open up. Cases are generated on the basis of psychometric function (PF) parameters for bone conduction (BC) thresholds and air-bone gaps (ABGs). PF parameters for air-conduction (AC) responses are internally generated by the software on the basis of the BC parameters and of the ABGs. For example, the AC PF midpoint is equal to the BC PF midpoint plus the ABG. The false alarm and the lapse rate for AC responses are assumed to be the same as those for BC responses. When you generate a case, virtual listener thresholds are simulated over a large number of iterations to obtain the median thresholds and the 95% confidence intervals that are shown by clicking the `Show actual threshold` checkboxes in the main `audiometry_trainer` window. Please see Section *Virtual listener* for further details on PFs.

The virtual listener parameters are specified using a spreadsheet-like interface in which each row represents a different frequency. Frequencies can be added/removed by clicking on the `Edit` menu and then on `Frequencies`. Only a discrete set of frequencies between 0.125 and 8 kHz can be used. Selecting with the mouse multiple cells and right clicking allows you to copy/cut/paste cell values as well as set the same value for all selected cells.

Some abbreviations are used in the column headers of the `generate case` window to save space: *mdp* for midpoint, *F.A.* for false alarm, *R* for right and *L* for left. To give some concrete usage examples, if you wanted to simulate a sensorineural hearing loss of 45 dB HL at a given frequency you could set the BC midpoint of the PF to around 43 dB HL, the PF width to around 8 dB, and the ABG to zero. The midpoint is the point at which the listener responds "Yes" 50% of the time (assuming F.A. and lapse rates close to zero). The threshold found with the Hughson-Westlake procedure is found at a %Yes responses between 50% and 100%. A width of 8 dB means that the PF will go from 5% to 95% correct in the span of 8 dB, therefore it will go from the midpoint (43 dB HL) to 95% "Yes", in the span of 4 dB (half the width). Overall, using these parameters the threshold will be located within a couple of dB of 45 dB HL. Given that a step size of 5 dB is used for the threshold search, the resulting threshold will almost certainly be found, on average, at 45 dB HL. This is simply an example and variations of each of the PF parameters (midpoint, width, F.A. rate and lapse rate) could be used to obtain a virtual listener with the same threshold but a different response behavior. Empirically, the region of uncertainty between inaudibility and certain detection in clinical audiometry has been found to be around 10 dB wide [Barr-HamiltonEtAl1969], therefore PF widths of around 10 dB should result in relatively realistic virtual listeners.

To simulate a conductive/mixed hearing loss simply set the ABG to the desired value of the conductive component. The AC PF midpoint will be shifted by the ABG with respect to the BC PF midpoint.

In addition to the BC and ABG PF parameters there are two columns, `Gain R` and `Gain L` that represent the gain of a hearing aid worn by the virtual listener. The values in these columns do not have any influence on the listener's responses to stimuli delivered via earphones or via the bone conduction transducer. They only influence free-field responses. Free-field responses are not currently implemented in `audiometry_trainer` but may be implemented in the future, so for now you can simply leave the `Gain R` and `Gain L` cells values to their default value of zero.

Once you're happy with the virtual listener parameters, you can input the desired number of simulations to run in the `No. simulations` input field and then click on the `Run simulation` button. You will be promted to select a destination file in which the results will be saved. To use the newly generated case file in `audiometry_trainer` simply load it through the main `audiometry_trainer` window by clicking on its `File` menu and then on `Load case`.

A random seed can be optionally set to obtain reproducible results. Additionaly, case information (e.g. type of hearing loss, source of the case, etc...) can be added to the `Case info` box. Markdown syntax can be used in this box and will be rendered appropriately when the case info is loaded into `audiometry_trainer`. Case information is saved to a case info file which is saved automatically when the simulation is run, even if the case info box is empty. The case info file will have the same root filename as the case file but will end with a `_info.md` suffix. In order for the case info file to be loaded by `audiometry_trainer` it has to be located in the same folder from which the case file is loaded.

The virtual listener parameters can be saved and later reloaded into the `generate case` window by clicking on the `Save parameters` and `Load parameters` actions found under the `File` menu. Saving a parameter file also saves any text found in the `Case info` box in a case info file. Loading a parameter files also loads case info from a case info file with the same root filename, provided that it is located within the same directory.

CHAPTER 7

User interface

To be written. Please refer to the other sections for the time being.

CHAPTER 8

Internals

Estimated thresholds are stored on a dict of the form ` self.est_aud['unmasked']['bone']['right'] `

matplotlib plotting points are stored in a parallel dict with the same structure: ` self.pnts['unmasked']['bone']['right'] `

another dictionary with a similar structure stores important parameters regarding the audiometric data and the plots: ` self.plot_prms['unmasked']['air']['left'] `

in particular the *thresh_status* key ` self.plot_prms['masked']['bone']['right']['thresh_status'] ` is used to differentiate between points marked as thresholds ('found' key), no responses at the maximum level ('NR' key), and masking dilemmas ('MD' key). Note that in each of these cases the stimulus level at which the cursor is located when marking the threshold will be stored in *self.est_aud*. The *thresh_status* key in the *self.plot_prms* dict allows to treat them differently according to the cases seen above.

Another key stored in *self.plot_prms* is used to indicate whether points for a given transducer and masking status should be displayed on the plot or hidden:

` self.plot_prms['masked']['bone']['right']['visible'] `

CHAPTER 9

References

CHAPTER 10

Indices and tables

- genindex
- modindex
- search

Bibliography

[Alcalá-QuintanaAndGarcía-Pérez2004] Alcalá-Quintana, R., & García-Pérez, M. A. (2004). The Role of Parametric Assumptions in Adaptive Bayesian Estimation. Psychological Methods, 9(2), 250–271. https://doi.org/10.1037/1082-989X.9.2.250

[Barr-HamiltonEtAl1969] Barr-Hamilton, R. M., Bryan, M. E., & Tempest, W. (1969). Applications of Signal Detection Theory to Audiometry. International Audiology, 8(1), 138–146. https://doi.org/10.3109/05384916909070201

[FagelsonAndMartin1994] Fagelson, M., & Martin, F. N. (1994). Sound pressure in the external auditory canal during bone-conduction testing. Journal of the American Academy of Audiology, 5(6), 379–383.

[Gelfand2016] Gelfand, S. A. (2016). Essentials of audiology (Fourth edition). Thieme.

[KingdomAndPrins2016] Kingdom, F. A. A., & Prins, N. (2016). Psychophysics: A practical introduction (Second edition). Elsevier/Academic Press.

[KussEtAl2005] Kuss, M., Jäkel, F., & Wichmann, F. A. (2005). Bayesian inference for psychometric functions. Journal of Vision, 5(5), 8. https://doi.org/10.1167/5.5.8

[MarshallAndJesteadt1986] Marshall, L., & Jesteadt, W. (1986). Comparison of Pure-Tone Audibility Thresholds Obtained with Audiological and Two-Interval Forced-Choice Procedures. Journal of Speech, Language, and Hearing Research, 29(1), 82–91. https://doi.org/10.1044/jshr.2901.82

[MartinEtAl1974] Martin, F. N., Butler, E. C., & Burns, P. (1974). Audiometric Bing Test for Determination of Minimum Masking Levels for Bone-Conduction Tests. Journal of Speech and Hearing Disorders, 39(2), 148–152. https://doi.org/10.1044/jshd.3902.148

[MunroAndAgnew1999] Munro, K. J., & Agnew, N. (1999). A comparison of inter-aural attenuation with the Etymotic ER-3A insert earphone and the Telephonies TDH-39 supra-aural earphone. British Journal of Audiology, 33(4), 259–262. https://doi.org/10.3109/03005369909090106

[Stenfelt2012] Stenfelt, S. (2012). Transcranial Attenuation of Bone-Conducted Sound When Stimulation Is at the Mastoid and at the Bone Conduction Hearing Aid Position. Otology & Neurotology, 33(2), 105–114. https://doi.org/10.1097/MAO.0b013e31823e28ab

[Studebaker1967] Studebaker, G. A. (1967). Clinical masking of the nontest ear. The Journal of Speech and Hearing Disorders, 32(4), 360–371. https://doi.org/10.1044/jshd.3204.360

[Yacullo1997] Yacullo, W. S. (1997). Clinical masking procedures. Needham Heights: Allyn and Bacon.