*multiThreshold* User's Manual



|  |  |
| --- | --- |
| http://14.media.tumblr.com/tumblr_kvo9zaxnmd1qz7tiao1_500.jpg | **Hearing Research Lab**  **University of Essex**  **2011**  **Volume 1, Issue 1** |

Contents

[1. Introduction 5](#_Toc311117421)

[1.1. Intended Use 5](#_Toc311117422)

[2. Hardware and software installation 7](#_Toc311117423)

[2.1. Download source 7](#_Toc311117424)

[2.2. Installation 7](#_Toc311117425)

[2.3. General setup 7](#_Toc311117426)

[2.4. Getting started 8](#_Toc311117427)

[2.5. General operating procedure 9](#_Toc311117428)

[2.6. Stopping the program midway 10](#_Toc311117429)

[2.7. Reporting bugs or software imperfections 11](#_Toc311117430)

[3. Using *multiThreshold* 12](#_Toc311117431)

[3.1. Experimenter graphical user interface (GUI) 12](#_Toc311117432)

[3.2. Stimulus parameters (left vertical panel) [1-13] 13](#_Toc311117433)

[3.2.1. Within-runs variable (WRV, red zone) [14-17] 15](#_Toc311117434)

[3.3. Paradigm settings (middle vertical panel) [18-19] 17](#_Toc311117435)

[3.3.1. Paradigm sub-panel [20-23] 17](#_Toc311117436)

[3.3.2. Paradigm [24-31] 19](#_Toc311117437)

[3.3.3. Paradigm [32-36] 22](#_Toc311117438)

[3.4. Graphics (right vertical panel) [37-40] 23](#_Toc311117439)

[4. Setting up an experiment 25](#_Toc311117440)

[4.1. Multiple runs in one session 25](#_Toc311117441)

[4.2. The task 27](#_Toc311117442)

[4.3. Saving the data 28](#_Toc311117443)

[5. Different paradigms 29](#_Toc311117444)

[5.1. Training 29](#_Toc311117445)

[5.2. Absolute thresholds 30](#_Toc311117446)

[5.2.1. '*absThreshold*' 30](#_Toc311117447)

[5.2.2. '*absThreshold\_16*' 31](#_Toc311117448)

[5.3. Forward masking paradigms (fixed target tone) 32](#_Toc311117449)

[5.3.1. '*training\_IFMC*' 32](#_Toc311117450)

[5.3.2. Iso Forward-masking contours (*'IFMC'*) 34](#_Toc311117451)

[5.3.3. Temporal masking curve (*'TMC'*) 36](#_Toc311117452)

[5.4. '*discomfort*' 38](#_Toc311117453)

[5.5. '*TENtest*' 38](#_Toc311117454)

[5.6. *'forwardMasking'* (fixed masker tone) 39](#_Toc311117455)

[5.7. Growth of masking (*'GOM'*) 39](#_Toc311117456)

[5.8. '*overShoot*' 39](#_Toc311117457)

[5.9. User defined paradigms 40](#_Toc311117458)

[6. Assessment methods 42](#_Toc311117459)

[6.1. Maximum likelihood 42](#_Toc311117460)

[6.2. Two-alternative forced choice (2I2AFC) 42](#_Toc311117461)

[7. Statistical modeling 44](#_Toc311117462)

[8. Bibliography 46](#_Toc311117463)

**Distribution**

This software is freely shared on the understanding that the recipient accepts full responsibility for how it is used. The author makes no guarantees of any kind by sharing this software. If you know someone who wants a copy, please refer them to the author who will send them the latest bug-free version of the software. Otherwise, old versions circulate to the detriment of our reputation!

**Acknowledgements**

While I am responsible for any bugs in the software, much of the development work associated with ***multiThreshold*** took place as part of the PhD studies of Wendy Lecluyse, Christine M. Tan and Manasa Panda. This software would have never happened without their help and the support of the charities ‘Deafness Research UK’ and ’Action on Hearing Loss’ and the Department of Psychology at the University of Essex.

# Introduction

***multiThreshold*** is a software tool for measuring a range of clinically relevant psychophysical thresholds quickly and painlessly. It has been designed around research activities in the Essex Hearing Research Laboratory using volunteers with a range of different hearing abilities.

A separate 'Quick Start' guide is intended as a useful primer and can be used before reading this manual if the reader already has a fair understanding of psychophysical tasks.

***multiThreshold*** is part of a much larger package (MAP1\_14) that includes auditory modelling software. The ***multiThreshold*** program has been designed so that it can evaluate a computer model of a subject's hearing as well as measure the subject's hearing itself. This document is concerned purely with measuring the hearing of a subject.

## Intended Use

The software can be used in many different ways in the laboratory for psychophysical measurements but the primary motivation for its development concerned the in-depth evaluation of impaired hearing. The aim is to go beyond the audiogram to evaluate other dimensions of hearing, particularly frequency selectivity and compression. The two figures (Fig. 1a and 1b) show profiles of two individuals; one with very good hearing (Fig. 1a) and one with impaired hearing (Fig. 1b). The continuous lines in the lower panel show the absolute thresholds for 250- and 16-ms tones at 250, 500, 1000, 2000, 4000 and 8000 Hz. The V-shaped functions indicate frequency selectivity and the upper row of panels are intended to reflect compression at the same frequencies. These profiles were measured using ***multiThreshold***.

|  |  |
| --- | --- |
|  |  |

Figure 1: These are examples of hearing profiles. Left: Hearing profile of normal hearing. Right: Hearing profile of impaired hearing.

Making these measurements using traditional laboratory techniques such as two-alternative forced choice procedures would be very time consuming. It may not even be possible with some elderly subjects who form the majority of people with a hearing impairment. To this end, we have developed the cued, single-interval measurement method which is the method used to illustrate this manual. In our experience, subjects are very tolerant of this technique and, at the time of writing, we have collected complete single-ear profiles from over 60 subjects.

A complete profile can be collected in about 2 hours of booth time. This is a short time by most laboratory standards but a long time by clinical standards. Complete profiles are therefore a luxury confined to the serious researcher of hearing impairment. Nevertheless, it is clear from the profiles above that much of the information in the profile is redundant and abbreviated profiles will be enough to get an insight into the nature of the subject's problems. The optimum strategy for trading measurement time against insight gained is currently under investigation.

A second motivation for the development of ***multiThreshold*** was the need for a measurement protocol that could be used to test computer simulations of hearing impairment using exactly the same procedures as those used to test the subjects. The modelling methodologies will be described in a separate document.

# Hardware and software installation

## Download source

The ***multiThreshold*** software can be downloaded directly from the author's website at **www.essex.ac.uk/psychology/department/people/Meddis.html**. The ***multiThreshold*** version that is published on this website is one that has been tried and tested. More serious users of the software who intend to make major modifications to the software should contact the author directly at **rmeddis@essex.ac.uk** so that they can be directed to a Git Hub webpage where they can have direct access, and regular updates to the latest version.

## Installation

The software package comes as a compressed folder called 'MAP1\_14' which can be saved in any location in the computer. Unzip the folder to make it ready for use.

## General setup

|  |  |
| --- | --- |
| C:\Users\ctan\Documents\My Dropbox\MultiThreshold_docs\images\IMG_3447.JPG | C:\Users\ctan\Documents\My Dropbox\MultiThreshold_docs\images\IMG_3442.JPG |

Figure 2: Examples of general setup. The experimenter (left picture) is seated in front of the main experimenter GUI whilst the subject (right) is sat facing the subject GUI on a secondary monitor. The subject is seated in the sound booth with the button box.

The software requires a MATLAB application (7.0 or later) and should work with any 24-bit sound card. It is assumed that the computer setup has extension monitors visible to a subject in a sound attenuated booth. It *can be made to work* with only one monitor but two is best (see ).

The listener in the booth should have control of a mouse. Alternatively, a Cedrus button box or even a touch pad can also be used to record responses. The listener should be supplied with headphones connected to the sound card. ***multiThreshold*** will need to be calibrated for the headphones and sound system used. This is the responsibility of the user. Before trying the software, make sure that the output is not painfully loud!

The software is optimised and has been tested with a Cedrus RB-834 model button box, Sennheiser HD600 headphones and M-Audio Audiophile 24-bit sound card. The software has defined the computer port COM2 as the port that should be used for the Cedrus button box. The computer port directory can be changed in the 'subjGUI\_MT.m' file.

## Getting started

1. Launch the MATLAB program.

2. Navigate to the 'MAP1\_14\multiThreshold 1.46' folder.

3. Run ***multiThreshold*** by typing 'run multiThreshold' in the command window. This should launch the experimenter's graphical user interface (GUI). Alternatively, right click on the 'multiThreshold.m' file and select 'Run' in the pop-up list.

4. To start measuring a threshold, type a subject's name into the 'name' box (top centre panel of the experimenter's GUI).

5. Click on the 'RUN' button.

6. This will now generate a second GUI for the subject.

## General operating procedure

|  |  |
| --- | --- |
| **Experimenter's GUI** | **Subject's GUI** |

Figure 3: Screenshots of the experimenter's GUI (left) and the subject's GUI (right).

The subject's GUI will always appear on the *experimenter's monitor* at first. This must be dragged onto the subject's own monitor if you have a two-screen system. This arrangement has the benefit that the experimenter GUI is not visible to the subject until the experimenter moves the GUI onto his monitor. This reduces the likelihood of a premature start.

By default, the program starts in 'training' mode. It is set up to make a simple pure-tone threshold measurement at 1000 Hz using a 100-ms tone stimulus.

The subject should be wearing headphones that are connected to the sound card of the computer. He initiates the measurements by clicking on the 'GO' button. One or two tones will be played at approximately 1 s later. When the tones have completed playing the subject GUI will change and will look like . The subject must then decide how many tones he hears and click on the appropriate button. It is important to wait for the buttons to appear on the screen before pressing any external hardware.

|  |
| --- |
|  |

Figure 4: Screenshot of subject's GUI after stimuli are played through the headphones.

The next stimulus will be presented approximately 1 s later and the series will continue until the threshold measurement is complete. The subject will hear a 'tada!' sound when the run is at an end and the measurement is complete. A 'Thank-you' message will appear on the subject's GUI.

## Stopping the program midway

If, for any reason the experimenter wants to restart the experiment prematurely, he can do so by clicking on the 'run' button again. This resets everything and all previous data from the current session are abandoned.

At the end of the session, summary information will be printed in the MATLAB command window. A great deal of information about the measurement procedure is printed. In fact, every variable and all results related to the measurement are printed out. Much of this can be ignored in everyday use but it can be vital when evaluating data retrospectively because it is a complete account of what happened. The most important information, e.g. thresholds, is printed both at the beginning and the end of the command window printed out.

The print out is tab delimited and is particularly suited to pasting in Excel ready for summary figures to be produced.

At the head of the printout, you will find something like this:

\*\*\*\*\*\*\*\* multiThreshold version 1.46

Name: Meddis

paradigm: training

Ear: left

method: oneIntervalUpDown/ withCue

date: 10-Sep-2011 07\_43\_51

thresholds

targetFrequency/ targetDuration

0.1

1000 21.4713

## Reporting bugs or software imperfections

The ***multiThreshold*** software is used as a tool in numerous other research applications and as such, will undergo continual modifications and improvements to the programming code. Any bugs or software imperfections found should ideally be reported to the author of the software as soon as possible so that corrections can be made. The author can be contacted directly by email at **rmeddis@essex.ac.uk**.

# Using *multiThreshold*

## Experimenter graphical user interface (GUI)

|  |
| --- |
| [Graphics](#Position_37)  [Stimulus parameters](#Position_1)  [Paradigm](#Position_18)    [**35**](#Position_35)  [**36**](#Position_36)  [**32**](#Position_32)  [**34**](#Position_34)  [**33**](#Position_33)  [**30**](#Position_30)  [**29**](#Position_29)  [**28**](#Position_28)  [**27**](#Position_27)  [**26**](#Position_26)  [**25**](#Position_25)  [**24**](#Position_24)  [**31**](#Position_31)  [**16**](#Position_16)  [**15**](#Position_15)  [**14**](#Position_14)  [**11**](#Position_11)  [**12**](#Position_12)  [**13**](#Position_13)  [**8**](#Position_8)  [**40**](#Position_40)  [**39**](#Position_39)  [**38**](#Position_38)  [**37**](#Position_37)  [**23**](#Position_23)  [**22**](#Position_22)  [**21**](#Position_21)  [**20**](#Position_20)  [**19**](#Position_19)  [**18**](#Position_18)  [**17**](#Position_17)  [**10**](#Position_10)  [**9**](#Position_9)  [**7**](#Position_7)  [**6**](#Position_6)  [**5**](#Position_5)  [**4**](#Position_4)  [**3**](#Position_3)  [**2**](#Position_2)  [**1**](#Position_1) |

The experimenter GUI can be used to define the stimulus and the testing procedure. It is also used to report back results while the testing is in progress. The definition of each measurement procedure requires a great deal of information. However, this can be supplied automatically by selecting a 'paradigm' from the paradigm drop-down list in the central panel of the GUI. When a paradigm is selected, all of the boxes in the GUI are automatically filled. After selecting a paradigm, changes can be made to the values in the boxes.

The following account is aimed at users who wish to manipulate the details of a paradigm. It also explains the meaning of the graphics. The GUI is divided into three vertical panels. The meaning of their contents will be explained in turn. Some editable boxes are automatically rendered invisible when they are not appropriate to the current paradigm (e.g. when a masker is not used in the stimulus - see position label **[**[**3**](#_Experimenter_graphical_user)**]**).

Each element in the experimenter's GUI is number labeled to enable easy identification of the items described. In subsequent chapters, these items in the experimenter's GUI are referred back to the experimenter's GUI on the previous page, by bracketed numbers in bold, e.g. **[**[**5**](#_Experimenter_graphical_user)**]** refers to the *gapDuration* box. These individual elements are also linked to detailed descriptions of their usage in subsequent chapters. This means that readers can click on these numbered elements (e.g. **[**[**5**](#_Experimenter_graphical_user)**]**) to jump straight to the section that describes their use.

## Stimulus parameters (left vertical panel) [1-13]

The boxes to the left of the experimenter's GUI screen allow the user to change the stimulus parameters. The quantities in the boxes are in Hz, seconds or dB SPL, as appropriate. The descriptions below refer mainly to 'cued, single-interval up/ down' estimation procedures.

|  |
| --- |
| **Test stimulus**  **Cue**  **Gap**  **Target**  **Masker**  **Ramp**    **Stimulus delay**  **Background** |

Figure 5: Example of a stimulus used in a cued, single-interval up/ down paradigm, with background sound added in grey.

|  |  |  |
| --- | --- | --- |
| Position | Name | Explanation |
| **1** | [Masker type](#_Experimenter_graphical_user) | A range of various types of masking stimuli is offered in a drop-down list. The pure tone option is the default. |
| **2** | [Masker duration](#_Experimenter_graphical_user) | This is measured in seconds. |
| **3** | [Masker level](#_Experimenter_graphical_user) | The masker level is rendered invisible if it is what is being investigated. This is true for all Within Runs Variables. |
| **4** | [Masker Relative Frequency](#_Experimenter_graphical_user) | These are masker/ target frequency ratios. For instance, if the masker frequency is required to be the same as the target frequency, then a ratio value of 1 is used. |
| **5** | [Gap duration](#_Experimenter_graphical_user) | The time delay between the end of the masker and the beginning of the target. The gap can be negative begins before the end of the masker. |
| **6** | [Target type](#_Experimenter_graphical_user) | A range of optional types of target can be used. The pure tone option is the default. |
| **7** | [Target frequency](#_Experimenter_graphical_user) | This is measured in Hertz. |
| **8** | [Target duration](#_Experimenter_graphical_user) | This is measured in seconds. |
| **9** | [Target level](#_Experimenter_graphical_user) | This is measured in dB SPL. |
| **10** | [Background type](#_Experimenter_graphical_user) | A range of various types of background stimuli is offered in a drop-down list. |
| **11** | [Stimulus delay](#_Experimenter_graphical_user) | The delay between clicking on a response button and the onset of the next stimulus in the sequence. The delay can also be the time interval between the cue and the test stimulus. |
| **12** | [Ramp duration](#_Experimenter_graphical_user) | All tones have cosine-squared onset and offset ramps. |
| **13** | [Cue-test difference](#_Experimenter_graphical_user) | The adjustment applied to the test stimulus when designing the (more audible) cue stimulus; e.g. a more intense target or a less intense masker, depending on the paradigm. |

## Within-runs variable (WRV, red zone) [14-17]

|  |
| --- |
| **14**  **15**  **16** |

Figure 6: Screenshot of red zone in the stimulus parameters panel.

In any test run, there is one stimulus parameter that changes adaptively from trial to trial. This is called the 'within runs variable'. The WRV is defined by the current paradigm. In the case of absolute threshold measurements, the WRV is, obviously, the level of the target tone.

|  |  |  |  |
| --- | --- | --- | --- |
| **14** | [WRV start values](#_Experimenter_graphical_user) | The value here is defined by the type of paradigm selected and can refer to either the level of the target, masker, gap duration etc. The starting value is selected randomly from trial to trial within the range +/- half of the large step size given in the WRV steps box (see below). | |
| **15** | [WRV steps](#_Experimenter_graphical_user) | This box controls the step sizes. It is the size of the change of the WRV between trials. This typically has two values; the initial step size and the final step size. The initial step size applies before the first reversal and the final step size applies thereafter.  When the subject reports having detected the stimulus, the level of the WRV will be changed adaptively to make the stimulus less audible. The amount by which it is raised or lowered is given in the WRV steps box.  The default values for the training paradigm are, for example, '10 2'. This means that the level will be changed in steps of 10 dB until the first reversal when the subject first changes his judgment from 'yes' to 'no' or vice versa. After this 'first reversal' the level is set to the midpoint of the most recent 'yes' and 'no' responses. The procedure then continues with a reduced step size of 2 dB. | |
| **16** | [WRV limits](#_Experimenter_graphical_user) | The track is restricted to within these level limits. If the next step will take the track outside of these limits, the run is aborted and the measurement reported as 'NaN' (not a number). | |
| **17** | [Message panel](#_Experimenter_graphical_user) | A white panel to the bottom-left of the experimenter GUI displays all of the subject responses, levels and threshold estimates of the current run.  Row 1: the subject response (1=detected, 0 = not detected).  Row 2: the corresponding level of the stimulus  Row 3: the current estimate of the threshold.  Other messages might be displayed here including warnings and reports of illegal parameter specifications.  At the end of a run, this information is replaced by a summary of the thresholds obtained. | |
|  | | |  |

## Paradigm settings (middle vertical panel) [18-19]

The middle panel is used to set up the design characteristics of the measurements procedure.

|  |  |  |
| --- | --- | --- |
| **18** | [Name](#_Experimenter_graphical_user) | The experimenter must type the subject's name here. It will be used in all printouts. |
| **19** | [Ear](#_Experimenter_graphical_user) | The second drop-down list in the middle panel selects the ear to be tested. In addition to the left and right it is possible to select diotic presentations.  A dichotic option also exists but only applies to paradigms used as a masker. 'dichoticRight' signifies that the target will be presented to the right ear while the masker will be presented to the left ear. 'dichoticLeft' is the converse.  The other options are related to modelling work. The auditory modelling (MAPmodel) facilities are not explained in this manual but the statistical modelling facilities (statsModelLogistic, statsModeRareEvent) are described below. |

## Paradigm sub-panel [20-23]

|  |
| --- |
| **23**  **22**  **21**  **20** |

Figure 7: Screenshot of paradigm sub-panel.

There are four drop-down list menus (grouped together) used to define the kind of experiment required. These are described in more detail below.

|  |  |  |
| --- | --- | --- |
| **20** | [Paradigm](#_Experimenter_graphical_user) | This first menu is the most useful. When you select a paradigm the software will put appropriate values in all the boxes on the GUI for you. For regular testing, most users will often only be required to choose a paradigm from the list.  For clinical use, the paradigms of most interest are 'training', 'absThreshold', 'absThreshold\_16', 'TMC' (for assessing compression) and 'IFMC' (for assessing frequency selectivity). Other paradigms are intended for laboratory use and for evaluation of computer models of hearing (see Chapter ).  When a paradigm is selected, all of the boxes on the GUI are re-populated with new values that are the default parameters for that paradigm. These values are stored in paradigm files in the paradigms sub-folder of ***multiThreshold*** *1.46*. These can be modified as described below. The user can also create new paradigms that automatically fill the boxes. This facility is explained later. |
| **21, 22** | [Between runs variable](#_Experimenter_graphical_user) | Variables that change between runs (e.g. target frequency in a series of absolute threshold measurements) are called 'between-runs variables'. The system allows for the possibility of two between runs variables. These variables are identified in the two drop-down list immediately below the paradigm menu. |
| **23** | [Presentation order](#_Experimenter_graphical_user) | The order of presentation of the runs can be fixed or randomized. Use the drop-down list to select an option.  The 'fixed sequence' option follows the order prescribed in the relevant stimulus parameter boxes; the second between-runs variable is the most slowly changing.  The 'randomize within blocks' option varies the second between-runs variable slowly in a fixed sequence but randomizes the sequence of the first between-runs variable.  The 'randomize across blocks' option presents the whole sequence in a random order.  ***multiThreshold*** expects only one value in a parameter box unless it is selected as a 'between-runs variable'. Putting multiple values in a parameter box is a mistake and will cause an abort unless that parameter has been selected as a 'between-runs variable' using one of the two drop-down lists. |

## Paradigm [24-31]

|  |  |  |
| --- | --- | --- |
| **24** | [Assessment method](#_Experimenter_graphical_user) | Various assessment methods are available (single-interval up/ down, maximum likelihood, two-interval forced choice (2AFC++, 2AFC+++). These are explained later. For subject testing, we recommend the 'single-interval up/ down' method. The other options have more use in a laboratory setting (see Chapter ). |
| **25** | [Cue/ no cue](#_Experimenter_graphical_user) | This gives the option of whether or not to play a cue stimulus before the test stimulus. The use of the cue is recommended because it produces the most sensitive measure of audibility. The 'no cue' condition is mainly used for modeling work.  In the 'no cue' condition, the subject merely has to indicate 'yes' or 'no' whether the target has been heard. In the 'cued' condition the subject is asked to count the number of targets. A '2' response indicates that the test target was heard. This is equivalent to a 'yes' response. A response of '1' indicates that only the cue target was heard. This is equivalent to a 'no' response. |

|  |  |
| --- | --- |
| **First reversal**  **Run**  **Trial** | **Series** |

Figure 8: Screenshots of graphics explaining the terms described in the text below.

Trial: A single episode consisting of the presentation of a stimulus and the recording of a single subject response.

Run: A complete sequence of the trials used to estimate a threshold.

Series: A complete sequence of runs used to estimate a number of thresholds.

|  |  |  |
| --- | --- | --- |
| **26** | [Stop criteria\ max trials.](#_Experimenter_graphical_user) | Each stimulus presentation is called a 'trial'. A run continues for a fixed number of trials after the first reversal. The value in this box sets the desired number of trials after the first reversal. The default is set to 20 although in practice it will often be found that 10 trials give just as good a result. This is particularly true for subjects with impaired hearing where the psychometric function is very steep and for whom very accurate measurements are not required.  If, at the end of the requested number of trials, the track is assessed as not yet close to threshold, the software automatically begins a new run of trials. This occurs as the track is continuously rising or falling (without any sign of a reversal), suggesting that it is the wrong measurement region.  Guidance on the number of trials to use is given in the description of the single-interval, up/ down method in Lecluyse and Meddis (2009). It is recommended to use between 10 to 20 trials. A *pdf* version of this article can be found in MAP1\_14\multiThreshold 1.46\documentation. |
| **27** | [Catch trial rate](#_Experimenter_graphical_user) | A fraction of the trials are ‘catch trials’. During a catch trial, no target is presented. If the subject responds ‘yes’ on a catch trial, the run is aborted and the subject must click on ‘Go’ to restart the run.  Subjects should be instructed to respond ‘yes’ only when they are sure that they have heard the target tone. We have experienced no difficulty with this conservative approach when using subjects and long error-free runs are normal. The matter is discussed in Lecluyse and Meddis (2009). Traditionalists often claim that this will lead to a biased estimate of the threshold but there is no experimental evidence to support this claim however self-evident it may seem to some.  The catch trial rate is set by two values in the ‘catch trial rate’ box. It may never be necessary to change this and the following is for information only. At the beginning of a run, the catch trial rate is the first number in the box. This is normally 0.2 and represents a 20% catch trial rate. If the subject is caught out, the rate is increased on the next run. The increase is in steps of 0.1 up to a maximum of 0.5. If he is not caught out on a run, the rate is reduced progressively to a minimum rate specified by the second value in the box (normally 0.1).  Catch trials never occur on the first trial of a run. Catch trials always occur on the second trial of a run. After that, they are distributed at random with no constraints. Catch trials serve to check that the subject understands the instructions. It is useful, therefore, that the first trial is always a regular trial and the second trial is always a catch trial. This gives early warning of a problem. However, the second catch trial gives the subject an exemple of a stimulus when no target is present and this can be a useful reference. The subject is not informed of this first/second trial rule. |
| **28** | [Music level](#_Experimenter_graphical_user) | The level (dB) of the musical signature (e.g. 'tada!') used to signal to the subject the end of a run or a series. This level is relative to a value clearly audible to subjects with normal hearing. The default value of '0' is adequate for listeners with normal hearing but may need to be increased for some subjects. |
| **29** | [Calibration](#_Experimenter_graphical_user) | A level of -28 dB SPL (on the GUI) will give an output of 1 at the sound card. If this generates 1 µPa at the tympanic membrane of the listener, your system is 'calibrated'. Otherwise, set a calibration value accordingly. Positive calibration will reduce the output from the sound card.  Negative calibration values can also be used. If a negative calibration value is used, this boosts the signal but the dynamic range is still reduced because the highest values cause the sound card to 'clip'. |
| **30** | [Sample rate](#_Experimenter_graphical_user) | The stimulus sample rate is fixed in the paradigm file and displayed on the experimenter GUI. The sample rate cannot be adjusted using the GUI but must be set using the paradigm file. |
| **31** | [Run](#_Experimenter_graphical_user) | Clicking on the red 'run' button initiates (or restarts) a measurement session using the parameters on the GUI at the time the button is clicked. You can hit 'run' at any time. It should abort the current run (if any) and start with the new set of parameters. |

## Paradigm [32-36]

|  |  |  |
| --- | --- | --- |
| **32** | [Msg font](#_Experimenter_graphical_user) | This sets the font size of the instructions displayed in the subject's GUI. |
| **33** | [Subject font](#_Experimenter_graphical_user) | This sets the font size of the numbers on the buttons in the subject's GUI. |
| **34** | [Phase](#_Experimenter_graphical_user) | This sets the phase of pure tones for both the masker and target. |
| **35** | [Save data](#_Experimenter_graphical_user) | This saves the output results of the completed task in an external Matlab file. Input '1' to enable saving or '0' to disable the function. |
| **36** | [Print track](#_Experimenter_graphical_user) | This prints the results of the complete level/ track responses in the command window. Input '1' to enable this function or '0' to disable it. |

## Graphics (right vertical panel) [37-40]

|  |  |  |
| --- | --- | --- |
| **37** | [Stimulus display](#_Experimenter_graphical_user) | This is a graphical representation of the most recent stimulus (half-wave rectified and plotted on a vertical log scale). This allows the experimenter to see what is happening. The representation is a half-wave rectified version of the stimulus presented on a decibel scale. |
| **38** | [Within-run variable track](#_Experimenter_graphical_user) | This panel shows the track, i.e. the sequence of the stimulus levels presented so far to the subject. The red line shows the running estimate of the threshold. The threshold estimated is updated after each trial. The threshold is the mean of the best-fit psychometric function to the data (see Lecluyse and Meddis, 2009). |
| **39** | [Previous thresholds](#_Experimenter_graphical_user) | This panel shows all the previous threshold estimates since hitting the red 'run' button. |
| **40** | [Psychometric function](#_Experimenter_graphical_user) | In this panel, the red line is the best-fit (logistic) psychometric function to the data on this run. The value k is the slope of the logistic function (g and A are part of the rare event function). The small dots at the top and bottom of the graph represents 'yes' and 'no' responses. The circles represent the proportion of 'yes' responses in a single 1-dB wide histogram bin. The size of the circle represents the proportion of responses included in the calculation. If a black line is also shown, this is the fit of a rival ('rare event') function of local interest to Essex Hearing Lab researchers. |

# Setting up an experiment

Every experiment is different and many parameters need to be established at the beginning of each session. ***multiThreshold*** takes some of the labour out of this by setting default parameters for a range of stimulus paradigms. The first step in setting up an experiment should involve the selection of a paradigm from the paradigm drop-down list in the central panel **[**[**20-23**](#_Experimenter_graphical_user)**]**. This will automatically change all parameters to a reasonable starting value. These can then be changed to suit individual needs before beginning a session. The characteristics of each paradigm are described later (Section ).

New paradigms can be created by the user. The MATLAB file should be created and saved in the paradigm folder.

The terminology used to describe an experiment has already been detailed in the previous chapter. Here are a few key reminders.

## Multiple runs in one session

A single threshold measurement is called a 'run'. Multiple runs are possible and, indeed, normal with ***multiThreshold***. To illustrate this process, select the 'absolute threshold' paradigm from the paradigm drop-down list in the central panel **[**[**20**](#_Experimenter_graphical_user)**]**. The target frequency box **[**[**7**](#_Experimenter_graphical_user)**]** will now contain a number of different frequency values:

250 500 1000 2000 4000 8000

Clicking on the 'run' button **[**[**31**](#_Experimenter_graphical_user)**]** will now initiate 7 threshold measurements in sequence. At the end of each measurement, the subject will be invited to click on the 'Go' button (on the subject GUI) to initiate the next run.

Two variables are always specified even though one may not be used. This is because the second value is used in printouts. Normally, the user should ignore these menus **[**[**21-22**](#_Experimenter_graphical_user)**]** are they are set automatically when a paradigm is selected from the paradigm drop-down list. They can be useful, however, for certain advanced applications. The value of the second, unused, 'between-runs variable' **[**[**22**](#_Experimenter_graphical_user)**]** is shown on the end of series printout in the MATLAB command window. This can sometimes be useful for record-keeping. In this example the printout shows that the duration of the target tone was 0.250 s. This is important information because the absolute threshold varies with signal level.

|  |
| --- |
| Name: Ray Meddis  Ear: left  date: 09-Apr-2008 07\_28\_09  paradigm: absThreshold  thresholds  targetFrequency/ targetDuration 0.25  250 15.8862  500 8.1767  1000 8.2452  1001 6.9469  2000 4.0175  4000 13.3904  8000 12.0761 |

**When two between-runs variables are specified, the second set of values will be changed more slowly.** For example, if duration is the second between-runs variable, we might chose to use two durations 0.016 and 0.25 s. In this case, all measurements at 0.016 s will be made before those at 0.25 s. If both variables are used in this way, the final report will look like this.

|  |
| --- |
| Name: Meddis  Ear: left  date: 09-Apr-2008 07\_38\_51  paradigm: absThreshold  thresholds  targetFrequency/ targetDuration  0.008 0.5  250 39.3717 17.2477  500 26.0641 9.6303  Target frequency is selected as between runs variables (1)  1000 23.3528 6.86  1001 22.357 9.173  2000 19.3514 4.5804  4000 21.7742 10.7419  8000 29.5017 12.6421 |
|  |
| Target duration is selected as between runs variables (2) |

## The task

The subject's task is to indicate whether or not he heard the target tone. This approach is commonly used in clinical situations. However, the method used to ask the question takes two forms depending on whether or not a cue is used.

|  |  |
| --- | --- |
| **Cued condition**  **Cue**  **Test Stimulus**    *Slower, easier task, more sensitive (most suitable for inexperienced observers)* | **No Cue condition**    *Faster, less sensitive and can be confusing for inexperienced users* |

Figure 9: Subject GUI for cued (left) and uncued (right) tasks

In the cued condition, either one or two target tones are presented and the listener is asked to count the number of tones he heard. The subject GUI for this task is shown on the left. In the *noCue* condition, either no target or one target is presented and the subject simply indicates 'yes' or 'no'.

In the *cued* condition, the presence of the cue on every trial simplifies the task and lowers thresholds by orientating the subject to the timing and nature of the target. In this condition, the subject is asked to count the number of targets heard ('1' or '2'). Moreover, subjects have little difficulty in grasping the concept of a counting task and this is recommended for clinical use.

The *noCue* task is quicker and may appear to be simpler but it is more confusing for an untrained user (i.e. a typical subject) because of uncertainties that arise when no target is presented as in a catch trial or when the target is below threshold.

It should be stressed to the subject that a '1', '2' or a 'yes' response should only be made when the judgement is certain. Listeners who try too hard may be responding positively when no target has been presented. This is clearly a poor basis for assessing hearing. Such responses will quickly be identified by the catch trials and the run will be aborted requiring the user to restart by pressing on ‘GO’ resulting in a slowdown of the measurements and a drop in subject morale. Encourage the subject to relax and treat this as a simple ‘did I definitely hear it?’ problem. This produces the best results and the shortest measurement times.

## Saving the data

At the end of each series a summary report of all thresholds is shown in the message window and a complete report is printed in the command window. The data for that session is also saved on disk in the ‘MAP1\_14\multiThreshold 1.46\savedData’ folder in a file called ‘mostRecentResults’.

A yellow ‘SAVE’ button appears at the end of each run. Click on this to move the data to a folder inside the savedData folder with the user’s name as the folder name. The data will be saved inside this folder with the data and time as part of the (.mat) file name. For example:

‘MAP1\_14\multiThreshold 1.46\savedData\Meddis\Meddis 26-Sep-2011 08\_46\_01.mat’

This button must be clicked *immediately* following the publication of the results otherwise the ‘mostRecentResults’ file will be overwritten be the next series of measurements.

# Different paradigms

The paradigm drop-down list can be used to select preset configurations of parameters in order to carry out frequently used measurements.

## Training

This paradigm is intended as an introduction to the testing situation. It is configured as a pure tone threshold estimation using the cued single-interval, up/ down method of assessment. The frequency is set to 1000 Hz and the duration is set to 0.1 s. The subject's task is to count the number of 'beeps' heard. The number of trials after the first reversal is set to 10. Only one run is requested. This paradigm is useful as an introduction to the procedure. After this, it is normal to move to the absolute threshold paradigm.

|  |  |
| --- | --- |
|  |  |

Figure 10: Screenshots of the *'training'* paradigm for the experimenter's GUI (left) and the subject's GUI (right).

The starting value for the target tone is at 30 dB SPL by default. This is ideal for normal hearing because it is clearly above threshold. For impaired listeners this value should be set at a much higher level. It is important not to start with a target that is effectively silent to the subject.

## Absolute thresholds

## '*absThreshold*'

|  |
| --- |
| **Test stimulus**  **Cue**    time |

This assesses the pure-tone threshold at a number of different frequencies using sued tones. The target duration is 0.5 s. The list of frequencies ranges from 250 - 8000 Hz on an octave scale. The starting level is at 30 dB SPL. This must be raised if a hearing loss is suspected.

|  |  |
| --- | --- |
|  |  |

Figure 11: Screenshots of the *'absThreshold'* paradigm for the experimenter's GUI (left) and the subject's GUI (right).

## '*absThreshold\_16*'

|  |
| --- |
| **Test stimulus**  **Cue**    time |

The 16-ms absolute threshold is often required before measuring Iso-forward masking curves (IFMCs) or temporal masking curves (TMCs). These paradigms require a 16-ms target to be set at a fixed level above its threshold. A special feature of this paradigm is that it does not specify the target frequencies. Whatever values were in the targetFrequency box **[**[**7**](#_Experimenter_graphical_user)**]** are left in place. Take care to check this box and change the target frequencies if necessary.

|  |  |
| --- | --- |
|  |  |

Figure 12: Screenshots of the *'absThreshold\_16'* paradigm for the experimenter's GUI (left) and the subject's GUI (right).

## Forward masking paradigms (fixed target tone)

## '*training\_IFMC*'

Forward masking tasks are more difficult than single-tone threshold tasks. Some guidance is needed to help the listener understand the task. The *trainingIFMC* paradigm **[**[**20**](#_Experimenter_graphical_user)**]** is intended to facilitate this process. First, it is important to explain to the subject what he is about to hear. A diagram may be helpful to some but often a simple verbal description will help.

**Cue** (easy to hear)

**Test stimulus** (hard to hear)

|  |
| --- |
| time |

Figure 13: Schematic of forward masking paradigm with fixed target tone.

In the cued condition (recommended) the task is to count the number of target tones and to ignore the masker beeps. This paradigm offers favourable conditions for carrying out this task.

The parameters of the paradigm consist of a 16-ms, 2-kHz target tone preceded by a 100-ms masker tone at a range of frequencies (0.5, 0.9, 0.7, and 1.3 relative to the probe tone) with a gap of 30 ms. To minimize confusion, the masker frequency is deliberately always different from the probe tone. To further simplify discrimination between the masker and the probe, the gap between the masker and probe is 30 ms (longer than for the regular IFMC condition). The initial level of the masker is deliberately set to below threshold so that the listener can hear the probe tones at the outset of the run and can identify what exactly it is that he has to count.

|  |  |
| --- | --- |
|  |  |

Figure 14: Screenshots of the *'training\_IFMC'* paradigm for the experimenter's GUI (left) and the subject's GUI (right).

Initial confusion is normal and it is down to the clinician’s skill to guide the user’s attention to the critical stimulus feature by using such phrases as ‘listen for the click/ roughness at the end of the beep’ or ‘ignore the beeps and count the clicks’. The care taken at this stage will be amply rewarded later with smooth error free runs and rich insights into the subject’s hearing.

## Iso Forward-masking contours (*'IFMC'*)

|  |
| --- |
| **Test stimulus**  **Cue**    **fp**  **fp**  **fm**  **fm**  time |

IFMCs are used to measure frequency selectivity by varying the frequency of a masker **(fm)** while tracking the audibility of a fixed-level, fixed-frequency target tone **(fp)**. The masker precedes the target by a fixed time interval of 0.01 s. It is the frequency of the masker that changes between runs. The expectation in a normal listener is that the required masker level will be greater, the bigger the difference in frequency between the target and the target. This gives rise to a V-shaped or U-shaped contour for listeners with normal hearing.

The default configuration is for a 16-ms target tone at 1000 Hz. The (relative) masker frequencies vary between 0.5 and 1.3 times the target frequency. The level of the 16-ms target tone must be set to 10 dB above its threshold. The threshold for the 16-ms target can be obtained by first running the ‘abs threshold\_16’ paradigm (see above). The figure shows examples of IFMCs for different target frequencies for a normal subject and for a subject with impaired hearing.

|  |  |
| --- | --- |
|  |  |

Figure 15: Screenshots of the *'IFMC'* paradigm for the experimenter's GUI (left) and the subject's GUI (right).

These diagrams below illustrate the kind of IFMCs that might be expected from normal and impaired hearing.

|  |  |
| --- | --- |
|  |  |

Figure 16: Examples of IFMCs for a young, male, normal subject measured using *multiThreshold* (left) and a subject with high-frequency hearing loss (right).

As a guide, the following verbal instructions can be used, "We will start this task by measuring just clicks (probe tones) first. Then we will start the next task. In this task, you will hear the clicks again. They will be very quiet, but you should be able to hear them. After a while, you will hear beeps (masker tones) that will gradually get louder and louder to try and make it difficult for you to hear the clicks. Your job is to count only these clicks and ignore the beeps. So sometimes, you may hear the sequence 'beep-click, beep-click' for two clicks, and other times, you may hear 'beep-click, beep-' for only one click. It may be difficult sometimes to hear these clicks, but try your best and don't forget to count only the clicks that you're positively certain of."

## Temporal masking curve (*'TMC'*)

|  |
| --- |
| **Cue**  **Test stimulus**    time  **Gap** |

TMCs are used to estimate auditory compression using a forward masking paradigm. In this paradigm, a fixed-level target follows a masker tone and the audibility of the trailing target is manipulated by changing the level of the preceding masker. The final threshold value is the lowest level of the masker that consistently masks the fixed-level target.

After paradigm selection the targetLevel box **[**[**9**](#_Experimenter_graphical_user)**]** is set to ‘NaN’ to remind the experimenter that it is his responsibility to supply this. The level of the target must be set by the experimenter prior to beginning the run. Typically, the target is set to be 10 dB above the subject’s 16-ms target threshold. The threshold for the 16-ms target should be obtained earlier by first running the ‘abs threshold\_16’ paradigm (see above).

The standard stimulus is a 16-ms, 10 dB SL target preceded by a 100-ms masker. The target and masker have the same frequency (default 1 kHz) and the gap between the masker and the target is variable between runs over a range of 10-100 ms.

The targetFrequency box **[**[**7**](#_Experimenter_graphical_user)**]** contains a range of frequencies to explore compression over a wide frequency region. When using the paradigm for the first time, it is recommended that only one targetFrequency is used (e.g. 1000 Hz).

|  |  |
| --- | --- |
|  |  |

Figure 17: Screenshots of the 'TMC' paradigm for the experimenter's GUI (left) and the subject's GUI (right).

For clinical purposes, it is recommended that only one masker frequency is used. This should be the same as the target frequency **[**[**7**](#_Experimenter_graphical_user)**]** and is indicated as ‘1’ in the maskerRelativeFrequency box **[**[**4**](#_Experimenter_graphical_user)**]**. The slope of the TMC can be compared with typical slopes obtained from normal listeners (30-40 dB rise in masker level as the gap increases from 10-90 ms). Pathological conditions are typically grossly different and in some cases are almost flat.

If a numerical assessment of the strength of compression is required, it may be necessary to measure two TMCs using two different masker frequencies; one at the target frequency and one below the target frequency. Repeat the series but adjust the maskerRelativeFrequency **[**[**4**](#_Experimenter_graphical_user)**]** accordingly.

For normal hearing, the slope of the TMC may be shallow at low signal levels and/or at high signal levels. When comparing with normal hearing use only the steepest portion of the slope. In the example below, the slope is low for gaps below 0.040 s and then becomes steeper. It is the steepest section of the function that gives most insight into the amount of residual compression.

|  |  |
| --- | --- |
|  | Thresholds  gapDuration/ targetFrequency  1000  0.01 17.0791  0.03 22.3628  0.05 27.7413  0.07 47.2053  0.09 58.6248 |

Figure 18: Result outputs of the 'TMC' paradigm.

## '*discomfort*'

The level at which a 500-ms pure tone becomes uncomfortable is measured using the 'discomfort' paradigm **[**[**20**](#_Experimenter_graphical_user)**]**. A range of different frequencies is specified as the between-runs variable. The subject's task is to rate each stimulus as 'comfortable', 'loud' or 'uncomfortable'. Only a single tone is presented on each trial (no cue). After each judgement, the level is raised by 3 dB. When the subject declares the tone to be uncomfortable, the trial is stopped and the next trial is initiated.

|  |
| --- |
|  |

Figure 19: Subject's GUI for the 'discomfort' paradigm.

## '*TENtest*'

Pure-tone thresholds for 5-ms tones at a range of frequencies are measured against a background of threshold equalized noise (TEN). The level of the noise can be set in the box alongside the background type drop-down list **[**[**10**](#_Experimenter_graphical_user)**]**. Unusually raised thresholds are taken to indicate possible dead regions.

|  |  |
| --- | --- |
|  |  |

Figure 20: Screenshots of the 'TENtest' for the experimenter GUI (left) and the subject's GUI (right).

## *'forwardMasking'* (fixed masker tone)

In this forward masking task, the target tone changes in level while the masker level is fixed. This paradigm is of interest mainly to psychophysicists. The default parameter set explores four different masker levels **[**[**3**](#_Experimenter_graphical_user)**]** and four different masker-target gaps **[**[**5**](#_Experimenter_graphical_user)**]**.

|  |  |
| --- | --- |
|  |  |

Figure 21: Screenshots of the 'forwardMasking' paradigm for the experimenter's GUI (left) and the subject's GUI (right).

## Growth of masking (*'GOM'*)

This paradigm measures the masker level just sufficient to mask a target at a number of different target levels. The maskerRelativeFrequency is set as a second between-runs variable **[**[**21**](#_Experimenter_graphical_user)**]** as it is traditional to measure the function for a second masker frequency that is one octave below the target frequency.

## '*overShoot*'

This paradigm measures the threshold of a target when presented simultaneously with a masking tone. The target is tested twice; first when the target begins immediately after the onset of the masker and, secondly, when the target is presented midway through the masker. Thresholds for the target are normally higher when the target is presented at the beginning of the masker. This is also known as the 'temporal effect'.

Notice that the two gapDurations **[**[**5**](#_Experimenter_graphical_user)**]** are negative. This is because the gap specifies the time between the end of the masker and the beginning of the target. If the target precedes the end of the masker, this implies a negative gap.

|  |
| --- |
|  |

Figure 22: Screenshot of the 'overShoot' paradigm for the experimenter's GUI.

This paradigm can be modified to explore a range of interesting paradigms including backward masking (gapDuration < maskerDuration), simultaneous masking (0 < gapDuration < maskerDuration) and forward masking (maskerDuration < gapDuration) all in the same session.

## User defined paradigms

A paradigm is defined by a MATLAB (*.m*) file in the paradigms folder:

MAP1\_14\multiThreshold 1.46\paradigms

To create a new paradigm, select an existing paradigm file that is most similar to the one you want to create. Save it under a new name in the paradigm folder and make the required changes to it. The name of the file must begin with ‘paradigm\_’. Otherwise it will be ignored.

The name of your new paradigm will automatically appear in the paradigm popup menu in the experimenter GUI after startup.

By convention, most paradigm files begin with a call to a file, *paradigmBase*. This gives a value to all of the variables that need to be defined in a paradigm. These are the same values as are used in paradigm\_training. This means that your new file need only contain variables that need to be changed (i.e. are different from the values in paradigmBase).

Stimulus sample rate is defined in one of the parameters in paradigmBase

stimulusParameters.subjectSampleRate=64000;

You can change the sample rate in any parameter file using this command.

# Assessment methods

***multiThreshold*** has been designed around the cued single-interval, up/ down procedure. However, two other methods are available; maximum likelihood (Green, 1993) and two-alternative forced choice. The assessment method can be chosen from the drop-down lost half way down the central panel **[**[**24**](#_Experimenter_graphical_user)**]**.

## Maximum likelihood

In many respects, this is similar to the single-interval, up/down procedure described above. The important difference is the method used to select the sequence of levels for the within-runs variable; see Green (1993) and Lecluyse and Meddis (2009) for more details. The maximum likelihood procedure converges more rapidly on the final threshold estimate but has a tendency to converge too quickly on erroneous estimates where it sticks whatever the responses of the listener. The matter is fully discussed in Lecluyse and Meddis. The inclusion of the procedure here is not a recommendation but an opportunity for comparison among traditional techniques.

## Two-alternative forced choice (2I2AFC)

The inclusion of this procedure is not recommended for profiling but is included here for the benefit of experimenters who are more familiar with this procedure and wish to compare its results with the recommended single-interval up/down procedure. Two varieties of the method are offered; ‘2I2AFC++’ and ‘2I2AFC+++’. In the first case, the tone or masker level is changed after two positive responses while, in the latter case, the change is made after three positive responses.

The measurement uses the large step size specified in WRVstepSize box until 2 reversals and then uses the small step size. The stopCriteria box contains three values; default is [75 3 5]. Here the first value is the upper limit of trials that will be allowed before the trial is brought to a conclusion. The second value is the minimum number of ‘peaks and troughs’ in the track (using small steps) to be used as the basis for a threshold estimate. The third value is standard deviation criterion used to reject a threshold estimate and continue for at least one more peak and trough in the track.

The subject GUI is slightly different for this procedure. The stimuli in the two intervals are played successively. When the first interval is presented, the button marked ‘1’ is illuminated. When the second interval is presented, the button marked ‘2’ is illuminated. The listener’s task is to choose which one contained the critical stimulus and click on the appropriate box.

|  |
| --- |
|  |

Figure 23: Screenshot of the 2I2AFC assessment method for the subject's GUI.

Lecluyse and Meddis (2009) compared this method with the single-interval up/down method and find that the latter is more efficient. Moreover, it was found using numerical simulations that the two-interval forced choice methods were biased low, i.e. they tend to underestimate the true threshold. This method also requires extensive training and is not suitable for clinical use. It is commonly claimed that forced-choice methods ‘control for criterion’ but there is little experimental evidence to support this view when dealing with absolute thresholds.

# Statistical modeling

The 'ear' drop-down list **[**[**19**](#_Experimenter_graphical_user)**]** supplies an option to study the statistical properties of all three assessment methods using numerical simulation. Select 'statsModelLogistic' to study the chosen assessment method on the assumption that the listener's responses are controlled by a logistic psychometric function. A new box will appear immediately below the 'ear' drop-down list that contains two values. The first is the threshold and the second is the associated slope of the function. These values can be edited by the experimenter. The default values are threshold = 15 and slope = 0.5.

When statistical modelling is used, the computer generates the responses that would normally be generated by the user. Click on 'run' **[**[**31**](#_Experimenter_graphical_user)**]** to see how accurately the procedure can estimate the 'true threshold' of 15. Note that the model is completely insensitive to the physical characteristics of the stimulus (frequency, duration...etc.). The response is purely dictated by a random selection from the logistic distribution.

All of the other controls on the experimenter GUI can be manipulated as usual. For example, the maximum number of trials can be varied to see how this improves accuracy.

When modelling in this way, it is normal to require many runs. This can be achieved by selecting the 'training' paradigm **[**[**20**](#_Experimenter_graphical_user)**]** and requesting a large number of frequencies **[**[**7**](#_Experimenter_graphical_user)**]**. For example, the entry '1000:1:1005' will generate 6 complete runs. Remember that the physical characteristics of the stimulus are always ignored in the statistical modelling and all runs will be equivalent.

The threshold estimates will appear in the message box and will also be printed out in the MATLAB command window. We might reasonably expect the method to produce threshold estimates close to 15 with no upward or downward bias.

|  |
| --- |
|  |
| thresholds  targetFrequency/ targetDuration  0.1  1000 14.2627  1001 14.3573  1002 15.6865  1003 14.7014  1004 14.7587  1005 15.3579 |

Figure 24: Screenshot of the statistical modeling (top) and example of the output (bottom).

# Bibliography

Lecluyse, W., and Meddis, R. (2009). “A simple single-interval adaptive procedure for estimating thresholds in normal and impaired listeners,” J. Acoust. Soc. Am. 126, 2570-2579.

Green, D. M. (1993). “A maximum-likelihood method for estimating thresholds in a yes-no task,” J. Acoust. Soc. Am. 93, 2096–2105.