WhisPER v1.8.0

User Documentation

This document was manufactured by Simon Ciba as a student at FG Audiokommunikation, TU Berlin, and completed in its first issue in November 17th, 2008. It is part of the software package WhisPER which has been developed by the author in collaboration with André Wlodarski. Additions to both this document and the software package WhisPER have been contributed by Matthias Herder, Andreas Rotter, Johannes Blickensdorff, Fabian Brinkmann, Martina Vrhovnik and Alexander Lindau.

Note: The author does not assume any liability for the correctness of this document's content and further takes no responsibility for any harm or damage resulting from its application.

Date of document version: Feb 4th, 2014.

Contents

List of abbreviations 5

1 Introduction **7**
1.1 What is WhisPER? 7
1.2 History and versions 7
1.3 License terms, conditions and availability 7
1.4 Warranty and safety information 7

2 A short characterization of the program's features **8**
2.1 Basic concept 8
2.2 Implemented types of test procedures 8
2.3 System and software requirements 10

3 Running the program **12**

4 The main window **13**
4.1 Menu bar 13
4.2 Desktop 14

5 Setting up a test series **16**
5.1 First step: Creation of a new test series 16
5.2 Second step: Definition of stimuli 16
5.3 Third step: Configuration of session setup 20
5.4 Fourth step: Specification of test sections 20
5.5 Fifth step: Setting up procedure properties 21

6 Performing a test run **22**
6.1 Initializing and starting a test run 22
6.2 The subject's GUI system 22

7 Test procedures – configuration and execution **25**
7.1 Adaptive psychophysical procedures (AP) 25
7.1.1 The test run 25
7.1.2 Setting up properties 26
7.1.2.1 Staircase 29
7.1.2.2 PEST 31
7.1.2.3 ML/Bayesian procedure 33
7.2 Repertory grid technique (RGT) 39
7.2.1 The test run 39
7.2.2 Setting up properties 42
7.3 Semantic differential (SD) 47
7.3.1 The test run 47
7.3.2 Setting up properties 48

7.4 ABX detection test (ABX) . . . 53
7.4.1 The test run . . . 53
7.4.2 Setting up properties . . . 53
7.5 Spatial Audio Quality Inventory (SAQI) . . . 57
7.5.1 The test run . . . 57
7.5.2 Setting up properties (Fig. 7.29) . . . 62
7.5.3 How to: Implementation of a new language . . . 65

8 Test procedures - default values' presets 67

9 Program set up 72
9.1 Basic audio setup . . . 72
9.2 Basic network setup . . . 72

10 Data handling 73
10.1 The test series folder . . . 73
10.1.1 The file 'testseries.info' . . . 73
10.1.2 The files 'TSD.mat' and 'TSP.mat' . . . 74
10.1.3 The subfolder 'audio' . . . 74
10.1.4 The subfolder 'export' . . . 75
10.1.5 The subfolder 'log files' . . . 82
10.1.6 The subfolder 'plots' . . . 82
10.2 The file 'PS.mat' ("program settings") . . . 82
10.3 The subfolder 'doc' . . . 82

Bibliography 84

List of Figures 87

List of abbreviations

AP adaptive psychophysical procedure

DW dialogue window (used in a generalized meaning, cp. annot. 5 on p. 13)

GUI graphical user interface

ID identification

LAN local area network

ML maximum likelihood

OSC open sound control

PEST parameter estimation by sequential testing

QUEST quick estimation by sequential testing

RGT repertory grid technique

SD semantic differential

ZEST zippy estimation by sequential testing

SAQI Spatial Audio Quality Inventory

1 Introduction

1.1 What is WhisPER?

WhisPER is a package of scripts under MATLAB for performing listening tests in the field of auditory-perceptive research. However its appearance to the user is meant to be rather more like a compact software, as most of the handling can be done by a coherent graphical user interface (GUI) system. The aim of WhisPER is to provide the experimenter a tool that can be used to automatize the courses of listening tests with respect to interaction with both the subject and also the playback environment. The patterns of actions which are performed during this to achieve a controlled elicitation of empirical data, are mostly defined by certain test procedures, several of which are implemented in this program (see sec. 2.2).

1.2 History and versions

WhisPER was developed by Andr Wlodarski and Simon Ciba within both of their magister theses (Ciba [2008], Wlodarski [2008]) at the FG Audiokommunikation, TU Berlin, during the summer semester in 2008. The emerged prototype (*version 1.0*) is intended to be further developed by the audio communication group and its students. Future work might include evaluation as well as upgrades by new test methods. Information about the history of software updates and the latest version is stated in the file *release_notes.txt* (see sec. 10.3). For up-to-date information about the program's state of development one may also visit the project's website http://www.ak.tu-berlin.de/whisper.

1.3 License terms, conditions and availability

Conditions of distribution of WhisPER are stated at the project's website http://www.ak.tu-berlin.de/whisper. For license information see the file *license_notes.txt* (see sec. 10.3).

1.4 Warranty and safety information

The use of the program – or any of its components (including this document) – for whatever purpose, is under the complete and sole responsibility of the user. The authors of WhisPER *do not extend any kind of warranty.* Special attention is drawn to the fact that WhisPER does *not* comprise an automatic mechanism to prevent the delivery of too high sound pressure levels. It is the explicit responsibility of the user to make sure that the stimuli in use, in combination with any further playback equipment (sound card, amplifier, headphone or loudspeaker, etc.), will yield a sound pressure level, that will not cause any damage to anyones health, especially the hearing system.

2 A short characterization of the program's features

2.1 Basic concept

Usually the situation in auditory-perceptive research is that an experimenter wants to conduct a set of listening tests by applying the same test procedure to different subjects, with some of the runs possibly performed under partially different experimental conditions than others. In this case runs under different conditions need sets of slightly different specifications that have to be made before. These procedure settings should not have to be dealt with from the very beginning before each run as a lot of them are mostly the same. To simplify the handling process in this respect the user interface system of WhisPER is designed in a way that test runs that belong to one research issue can be summed up and centrally controlled from one point of the system. For this purpose the program provides the administration of all data, including the empirical ones elicited during the runs, within a virtual environment called *test series*, which is structurally represented by one conjoint data folder (see sec. 10.1). Exporting empirical data to e.g. SPSS and Excel is possible as the software automatically generates a suitably formatted text file (CSV). Import of grid data into specialized repertory grid packages like Idiogrid (http://www.idiogrid.com/) and Gridsuite (http://www.gridsuite.de/) is simplified by supporting their respective native formats GridXML and GRD (see sec. 10.1.4).

WhisPER allows performing more than one sub-tests in succession during one test run, with possibly some of them being attached to different test procedures. This feature tries to meet the individual requirements of specific issues in experimental research. As only one example to be mentioned, it is often desired to carry out a familiarization experiment immediately before starting the actual listening test.

2.2 Implemented types of test procedures

In the present state WhisPER provides the performance of three different types of test procedures, which will be listed in the following with each given a short (!) survey of basic implemented features and some essential and potentially helpful references on literature. Besides they are described in more detail within the magister thesis of Simon Ciba [Ciba, 2008, chap. 3].

adaptive psychophysical procedures: A group of psychophysical procedures that aim at supplying efficient estimates of a single point on a psychometric function that is called a psychophysical threshold. This is achieved by adapting the intensity of the stimulus to be presented to the history of preceding subject responses. By this *adaption mechanism* and the method for estimating the final output several procedures can be distinguished. The current implementation contains *simple* and *transformed staircase*

procedures with the adaption rules 1-down/1-up, 2-down/1-up and 3-down/1-up, bisections of step size after predefined reversals and a calculation of the threshold estimate by averaging stimulus intensities at a defined number of last reversals. Further more the *parameter estimation of sequential testing (PEST)*, a modified version of this by Findlay referred to as *More Virulent PEST* and a maximum likelihood or bayesian procedure respectively which is based on the *Best PEST* but includes some extensions (see subsec. 7.1.2.3) taken from several similar procedures, can be applied. Finally of those the *QUEST* and *ZEST* procedures have explicitly been made available, too. At present the software allows employing these procedures in detection and discrimination tasks by utilization of the *n alternative forced choice (nAFC)* paradigm for n=2,3 or 4. Randomly interleaving of an arbitrary number of tracks using the same response paradigm is possible. *references:*

- overview: [Gescheider, 1997, pp. 159], Leek [2001], Treutwein [1995]
- Staircase: Cornsweet [1962], Levitt [1971], Wetherill [1963]
- PEST: Taylor and Creelman [1967]
- More Virulent PEST: Findlay [1978]
- ML/Bayesian procedure: Lieberman and Pentland [1982], Pentland [1980], Treutwein [1995], Treutwein [1997]
- QUEST: Watson and Pelli [1983]
- ZEST: King-Smith et al. [1994]

repertory grid technique (RGT): A combination of a qualitative elicitation and a direct scaling procedure, whereas for the latter basically different possibilities (dichotomization, ranking, category rating) exist, out of which the rating method (cp. *semantic differential* below) was chosen. During the first part of the procedure stimuli (called *elements*) are combined to groups of three (called *triads*) and presented to the subject that is supposed to report on similarities and contrasts. The result of each triad comparison is a pair of verbal expressions that are likely to represent the end-point descriptors of a bipolar attribute referred to as a *construct*. In the second part the elements are rated on scales which are formed by the elicited constructs. Scales can either be discrete by using radio buttons or continuous by using sliders[1]. The formation of triads can be carried out manually by the experimenter or automatically by the program by creating a complete variation. Further more the order of presentation of triads and elements can either be defined by the user or generated at random. There is a chance as there may possibly be a need for editing the elicited constructs in between the two depicted parts of the procedure. *references:* [Bech and Zacharov, 2006, 54], Berg and Rumsey [1999], [Bortz and Döring, 2006, 187], Fransella and Bannister [1977], Kelly [1955]

semantic differential: A procedure of direct scaling, more precisely spoken of as a (category) rating procedure. Stimuli (called *objects*) are evaluated on a set of rating scales which can be either discrete by using radio buttons or continuous by using sliders[2]. In this implementation the order of presentation of the objects can be defined by the experimenter as well as be generated randomly. *references:* [Bortz and Döring, 2006, pp. 185], Osgood et al. [1967]

[1] Sadly, sliders have not been implemented yet, as this venture finally turned out to be more tricky and time-consuming than at first assumed.

[2] See annotation 1.

a/b/x test: A method of comparing two different sensory stimuli for detecting perceptible differences. Usually used for comparing amplifiers or other devices known for exhibiting very small perceptual differences. The two stimuli A and B are presented in combination with a stimulus X which is randomly chosen out of A or B. The subject's task is to identify whether X equates stimulus A or B. The number of trials (repetitions) may be chosen depending on a chosen significance level, effect size and test power. The procedure is implemented as double blind to prevent supervisors of having some unconscious influence. *references:* Clark [1991], Leventhal [1986], Burstein [1988]

Spatial Audio Quality Inventory: The Spatial Audio Quality Inventory is intended for a qualitatively differentiated, comparative auditory assessment of real, imagined and simulated acoustic scenes in order to to reveal specific shortcomings of a simulation under test and allow for a directed technical improvement. The SAQI comprises 48 verbal descriptors of perceptual qualities assumed to be of practical relevance when comparing virtual environments (VAEs) to real or imagined references or amongst each other. In WhisPER it is implemented as a widely customizable questionnaire-like listening test (i.e. using rating scales and multiple choice check boxes). It is further available in a German and an English version. *references:* Lindau et al. [2014a], Lindau et al. [2014b], Lindau [2014]

2.3 System and software requirements

Operating system WhisPER was developed to be executed on a 32-bit version of Windows Vista. It has also been tested on a 32-bit version of Windows XP. Usage of the software under different operating systems like Linux or OS X has generally been enabled although not extensively been tested, use at your own risk.

Running environment (MATLAB version) Implementation of the software package was done on MATLAB version 7.4.0.287 (Release 2007a), additions have been made using MATLAB version 7.9.0.529 (Release 2009b). Version 1.8.0 of whisPER was tested to run on MATLAB 2013a. It is possible that also former versions of MATLAB can be used to run the scripts, though it is known that it has to be at least one of the 7th series (v7.0 upwards). However WhisPER has not been tested on these.

Important note: Since vs.1.8.0 the OSC capabilities of WhisPER are realized by Mark Marijnssen's script ***oscsend.m*** (on-line at MATLAB Central) which depends on MATLAB's Instrument Control Toolbox.

This means, that - for complete OSC capabilities - WhisPER **will now require the Instrument Control Toolbox (included in MATLAB's student version) to be installed.**

Display resolution The GUI system requires a display resolution of at least 1280x800 pixels.

Playback environment Presentation of the acoustical stimuli can either be done by playing .wav files, using a sound card that supports the *Direct Sound* device.[3] The other possible way is through sending OSC (open sound control) commands by a LAN (local

[3] This applies to the Windows operating system. On other operating systems make sure MATLAB's *audioplayer*-object does work with your sound card.

area network) connection. Thus any type of playback periphery can be controlled, auditory and visual likewise, even more than one simultaneously, as there can be up to 6 OSC commands sent "at once"[4] or in sequence after specified delay times.

[4]Technically MATLAB executes all commands in sequence, even though this is normally done very quickly.

3 Running the program

To start WhisPER, there have to be the following three steps taken in given order:

1. Start MATLAB.
2. Define the path of the program folder 'whisper1.0' as the current (working) directory in MATLAB. This can either be done by using the *Current Directory Browser* or using the function cd (change directory) by typing *cd('PathofProgramFolder')* into the command window and afterwards pressing *return*.
3. Start WhisPER by typing *whisper* into the MATLAB command window and afterwards pressing *return*.

After the conduction of these steps the GUI system's main window will appear which is labeled as *WhisPER*. It will be described in the next section.

4 The main window

The starting point of any action is the GUI system's main window which is labeled as WHISPER and appears right after starting the program as described in chapter 3. All functionality is available directly or indirectly from this part of the user interface system. Some functions can be addressed by using the *menu bar*, some by using control elements placed on the main window's *desktop* and some in both ways. The following two subsections supply an overview and a short description of the functionality contained in the main window.

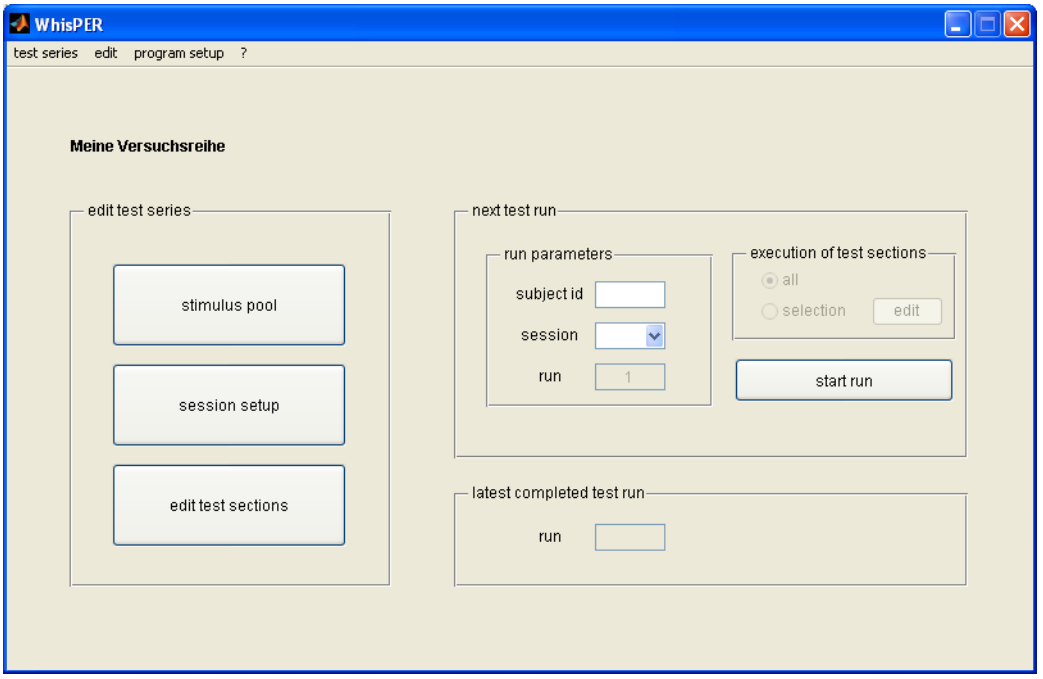

Figure 4.1: The *main window*.

4.1 Menu bar

In this subsection the items contained in the main window's menu bar are listed and described.

Main window (menu bar)

test series

new Opens a dialogue window (DW)[5] for creating a new *test series*. You have to specify name and location of a folder which will be created and later contain all the data

[5]This expression is used here in a generalized meaning, i.e. for any window of the GUI system that allows for the information flowing in both directions – from the user to the program and vice versa.

of *test runs* referring to a common research issue (cp. sec. 2.1). The inherent structure of this folder in terms of sub folders and files included will be described in sec. 10.1.

open Opens a DW for opening an existing *test series* by browsing for the belonging test series folder.

close Closes the current test series. *Note:* Every time one makes changes on settings, those will be automatically saved after clicking on the *OK* button on the respective window. That means that one does not have to save a test series' data before closing it.

export empirical data Manually creates the empirical data export file(s) (called *export sheets*) in the sub folder *export* (see sec. 10.1.4).

exit program Closes the program.

edit

stimulus pool Opens a DW for the definition of stimuli (see sec. 5.2).

session setup Opens a DW for making global specifications necessary for the use of multiple *sessions*. At the present state this is the number of sessions that is required for the current *test series*. By default this value is set to 1.

test sections Opens a DW for selecting the desired test procedures, editing test sections (see sec. 5.4) and setting up procedure parameters on a further DW (see sec. 5.5).

empirical data Opens a DW for deleting the empirical data of the latest completed test run.

procedure's defaults Opens a DW for configuring the default values of procedures as they are initialised and added to a *test series* (see sec. 8).

program setup

audio <disabled, cp. sec. 9.1>

network Opens a DW for adjusting basic settings of the network environment (see sec. 9.2).

plotting Opens a DW for adjusting settings referring to the data plots' file format.

?

help Opens a pdf-document (if acrobat reader is available) containing the user documentation at hand.

about Displays information about the version of WhisPER currently used.

4.2 Desktop

In this subsection the control elements placed on the main window's desktop are listed and described.

Main window (desktop)

-no test series loaded-/name_of_current_test_series (*dynamic superscription*) Indication of the test series currently loaded.

edit test series (*panel*)

stimulus pool (*button*) see sec. 4.1, menu item *edit\stimulus pool*

session setup (*button*) see sec. 4.1, menu item *edit\session setup*
edit procedures (*button*) see sec. 4.1, menu item *edit\test sections*

next test run (*panel*)

run parameters (*sub-panel*)

subject id (*input field*) Space for entering the ID of the subject that is going to take part in the next test run. The input is treated as a string variable.

session (*input field*) Space for entering the integer key that represents the session key number which should be assigned to the next run.

run (*output field*) Displays the count number (i.e. the value of the respective counter) of the next run.

execution of test sections (*sub-panel*)

all (*radio button*) If selected, all test sections (see sec. 5.4) will be performed during the next test run.

selection (*radio button*) If selected, only a specified subset (see button *edit*) of test sections will be performed during the next test run.

edit (*button*) Opens a DW for determining a subset of test sections to be performed during the next test run. This button is active only if the second of the preceding options has been selected.

start run (*button*) Starts the next test run and opens the GUI system for subject guidance.

latest completed test run (*panel*)

run (*output field*) Displays the count number of the latest completed test run.

5 Setting up a test series

In this chapter the approach to setting up a new test series from the very beginning will be described. It is also possible to open an existing test series which has been partially set up already and carry out only a few of these steps in order to address especially those settings that one wants to be adjusted. Conducted changes in setup will be saved instantaneously after confirmation by pressing the respective *OK* button. In most cases dialogue windows contain a *cancel* button that allows to discard changes (only those made on the current window) if necessary.

! Note that changes on the test series setup are only possible as long as the counter of the latest completed test run contains the value 0, respectively the counter of the next test run is 1. Otherwise, if you are still willing to make changes on the setup, you have to reset the counters by deleting the total empirical data set which is stored in the file *TSD.mat* contained in the current *test series* folder (see sec. 10.1). This may either be done by deleting (or removing) the whole file at once or by using the item *edit\empirical data* from the menu bar which allows for singularly deleting the data of the runs, i.e. one after another. Note that in the first case one has to restart the program so that the action will take effect on the program's current status. Additionally it is recommended to clear the contents of the sub folders *plot*, *export* and *logfiles*. Otherwise this could lead to confusion as old and new files will be mixed and old files might be gradually overwritten (not in general but in most cases if old and new files carrying the same labels). Do not forget to make a copy of the folders' contents if you want to preserve the data.

5.1 First step: Creation of a new test series

The creation of a new test series is done by selecting the corresponding item from the menu bar (see sec. 4.1). Then a DW will appear for specification of name and location of the new test series folder which will be automatically created including some sub folders and files. The folder's inherent structure will be described in sec. 10.1.

5.2 Second step: Definition of stimuli

Before stimuli can be chosen for application in certain test procedures, they have to be defined on a global level. For this purpose use the button *stimulus pool* (see sec. 4.2) or the item *edit\stimulus pool* from the menu bar (see sec. 4.1) to access a DW labelled as *stimulus pool* (see fig. 5.1), which contains a list providing an overview on the stimuli which have been defined already and their belonging playback properties. The supplied information for each stimulus consists of an automatically generated *ID*, a user-defined label (*name*), if applicable the name of a *.wav* file or a set of *OSC commands* or both.

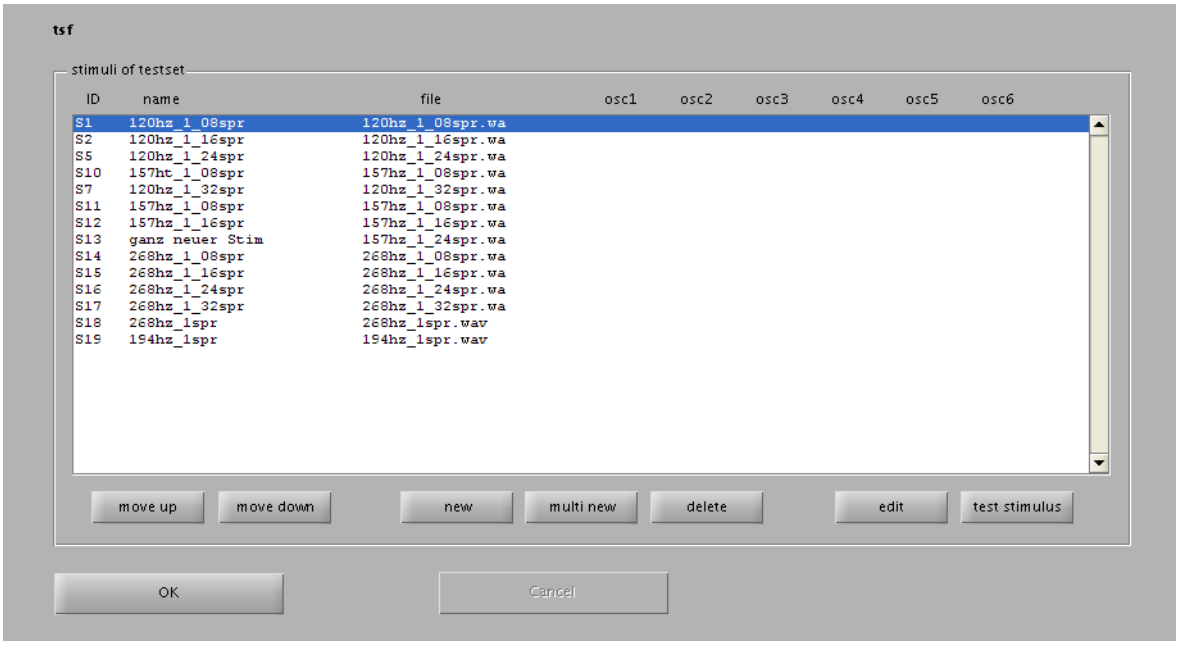

Figure 5.1: DW *stimulus pool*

New stimuli can be added by using the button *new* or *multi new*. While the former will only generate an empty stimulus in the list which later will have to be edited in order to be useful, the latter will open a file selection dialog window allowing multiple selection of .wav files. For each .wav file selected a separate stimulus associated with that .wav file will be added to the list.[6] Names will be auto-generated from the file names, but of course can be edited later.[7]

Unwanted stimuli can be deleted by using the *delete* button.[8]

Stimuli can be sorted using the *move up* and *move down* button, and the order the stimuli are in here will be the order the stimuli will be presented to you in all the other DWs when there are stimuli to be selected from the global stimulus pool.

If one wants to test if a selected stimulus' properties (see below) have been set well, one can use the button *test stimulus* for a presentation of it on a trial basis.

The editing of a stimulus' properties is performed by selecting the concerning position of the list and then using the *edit* button, which will open a suitable DW labeled as *edit stimulus* whose control elements will be described below. On this window definitions can be made regarding the stimulus' label (*name*), duration and content. The latter can be chosen either as a .wav file or a set of up to 6 OSC commands or even both at a time. That means that both kinds of playback environment, auditory and visual, can be addressed. If you choose a .wav file from any location of your data drive, it will be copied to the folder *audiofiles*, which is a sub folder of the *test series* folder (see sec. 10.1). The idea behind this behavior is to be able to easily transfer a test series' configuration data set between computers with a minimum amount of adjustments necessary. One restriction resulting from this is that the name of a

[6] For techniques of economically configuring multiple OSC stimuli instead please cf. sec. 8.

[7] Auto-generation of names from file names will be suspended however, if the default stimulus name has been changed, cf. sec. 8.

[8] While it is possible to delete global stimuli which have been assigned inside procedures (see sec. 7), you are highly encouraged to follow good practice and separate global stimulus setup from procedure stimulus assignment and first perform the one, afterwards the other.

chosen .wav file must be unique and not allocated to several files. When uploading a .wav file the stimulus' duration will be automatically determined as the file's length. The stimulus' name will also be auto-generated.[9] If the desired performance involves OSC commands, the stimulus' duration should be edited or must be manually entered respectively (the latter is the case if no .wav file is chosen). This is necessary because the software will receive no feedback information about the termination of playbacks controlled by OSC. As a consequence the entered duration period has to be long enough to include all presentations of the different contents, i.e. possibly a combination of a .wav file and several OSC commands to be executed at a time.

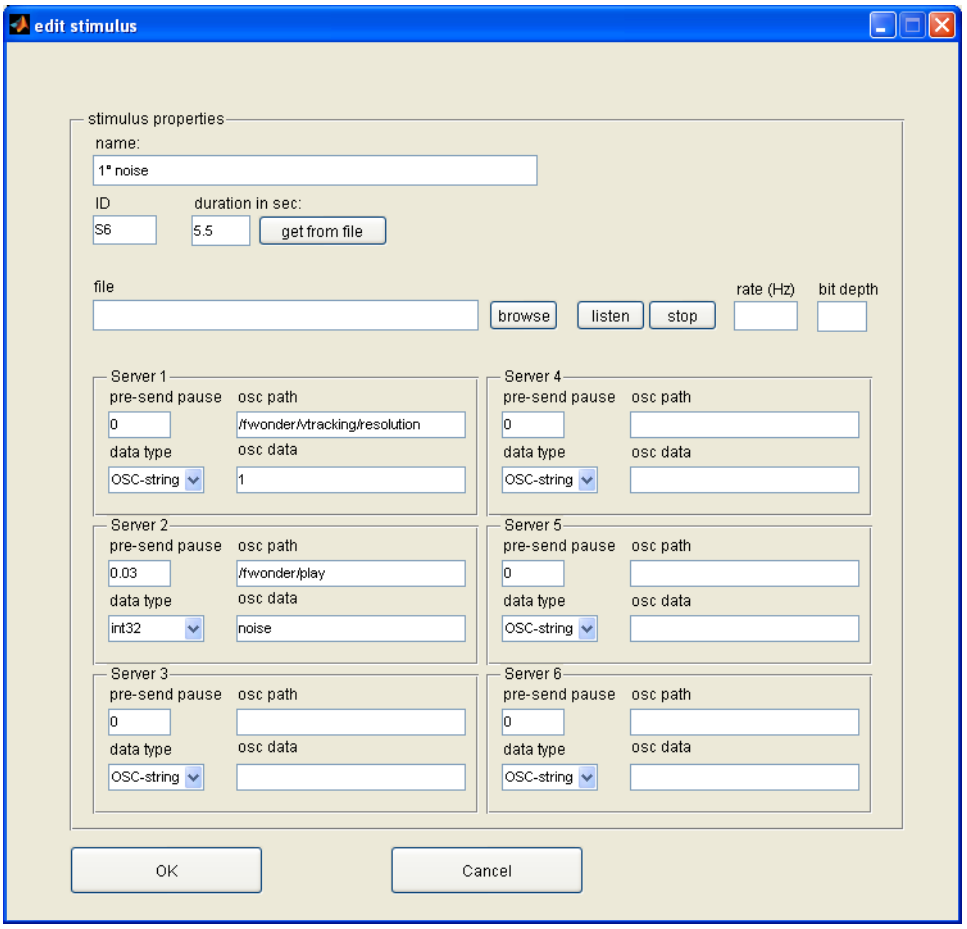

Figure 5.2: DW *edit stimulus*

[9] This does *not* apply if the name has previously been altered from the default! Cf. also sec. 8

DW *"edit stimulus"*

stimulus properties (*panel*)

name (*input field*) Space for entering the name or a short description of the stimulus respectively. The input will be treated as a string. This should release the user from distinguishing the selected stimulus from the remaining ones during the process of setting up test procedures. If this field has not yet been altered from the initial default and you select a .wav file by way of the appropriate dialog, this field will be auto-filled with a name generated from the .wav file minus its extension. Which name is displayed here upon stimulus creation as a default name can also be manipulated, cf. sec. 8.

ID (*output field*) Displays the stimulus' ID.

duration in sec (*input/output field*) In this field either the duration of the stimulus in [sec] can be manually entered or will displayed when received by using the *get from file* button.

get from file (*button*) Obtains the stimulus duration from a .wav file if there has been one already specified in the *file* field.

file (*input/output field*) In this field the original path of a .wav file which shall be played back directly from the sound card as part of the stimulus' audio content, either will be manually entered or displayed if received by using the file dialog which can be accessed by using the *browse* button.

browse (*button*) Opens a file dialog for selecting the path of a .wav file which shall be played back directly from the sound card as part of the stimulus' audio content. When leaving the current DW (*stimulus properties*) and confirming changes by pressing the *OK* button, the selected file will be copied to the folder *audiofiles* which is a subfolder of the *test series* folder (see sec. 10.1). After this the *file* field will no longer contain the path but only the file's name.

listen (*button*) Plays the .wav file if one has been specified. This function can be used for identification and calibration purposes.

stop (*button*) Stops the .wav file's playback.

rate [Hz] (*output field*) Displays the sampling rate in [Hz] if a .wav file has been specified.

bit depth (*output field*) Displays the resolution of the quantization process in [bit] if a .wav file has been specified.

Server 1–6 (*panel*) Here a set of up to 6 OSC commands can be determined with each addressing one of up to 6 servers which have to be specified globally (see sec. 9.2). The commands will be executed in sequence with the order given by their ascending label numbers.

pre-send pause (*input field*) Space for entering a time period in [sec] that lies between the previous and the current command's execution.

osc path (*input field*) Space for entering the path of the parameter one wants to address.

data type (*drop-down list*) Here one can choose among five types of data formats which are *OSC-string*[10], *int32*, *float32*, *True* and *False*. For a description of these the user is directed to e.g. http://opensoundcontrol.org/spec-1_0.

[10] A sequence of ASCII coded characters terminated by zero (C-String), filled up with zeros to a length of an integer multiple of four.

osc data (*input field*) Space for entering the data of the format which has been previously specified. This field will be disabled if either *True* or *False* are selected.

5.3 Third step: Configuration of session setup

In order to be able to perform and control different test runs under several experimental conditions within one test series (cp. sec. 2.1), a structural variable called *session* has been defined. Each subset of runs under one equally determined set of experimental conditions is thereby attached to a specific value of this variable. The variable's range starts at 1 and extends by integer steps to its maximum which is called *number of sessions*. Before the different experimental conditions can be specified for each test section, the number of sessions has to be determined as a global parameter. This can be done by using the button *session setup* placed on the main windows' s desktop (see. sec. 4.2) or selecting the item *edit\session setup* from the menu bar (see sec. 4.1). Thereupon a suitable DW will open. By default the number of sessions is set to 1.

5.4 Fourth step: Specification of test sections

As mentioned in sec. 2.1 WhisPER allows to sequentially perform several sub-tests within one test run. For this reason there may be several *test sections* assigned to a single test run. Moreover for each test section one must select an appropriate test procedure. Both tasks can be carried out by either using the button *edit test sections* which is placed on the main window's desktop or selecting the item *edit\test sections* from the menu bar. In both cases a DW appears which is labelled as *edit test sections* and whose control elements are described in the following.

DW *"edit test sections"*

available test procedures (*panel*)

<no label> (*list box*) Shows all test procedures available under WhisPER.

add to list below (*button*) Adds a selected test procedure to the bottom of the list of *test sections* below.

test sections (*panel*)

<no label> (*multi-columned list box*) Shows all test sections in an order defined by the user. The entries of the three columns represent the count number, an internal label which should support identification by the user (i.e. the experimenter) and a synonym which will be used for subject guidance during a test run.

<no label> (*output field* 1st col) Shows the count number of the selected test section.

<no label> (*input/output fields* 2nd & 3rd col) Space for displaying and editing of both the labels for the experimenter and the subject belonging to the selected test section.

apply (*button*) Applies changes in the contents of the input/output fields above to the selected test section.

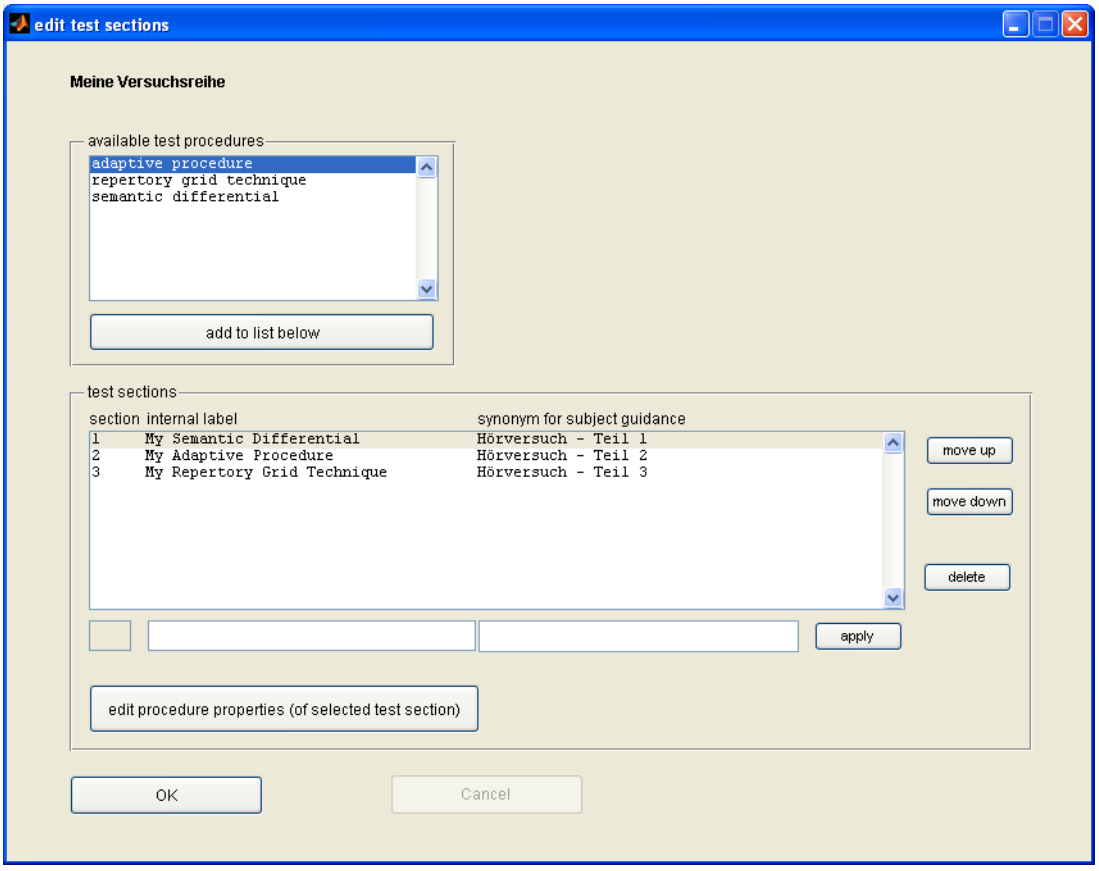

Figure 5.3: DW *edit test sections*.

delete (*button*) Deletes a selected test section.

move up/move down (*button*) Shifts the selected test section to a proximate list position above/below.

specify test properties (*button*) Opens a DW for determining the properties of the test procedure that is chosen for the selected test section (cp. sec. 5.5).

5.5 Fifth step: Setting up procedure properties

The process of determining a specific test procedure's properties is triggered by applying the button *specify test properties* from the DW *edit test sections* which has been described in sec. 5.4. Then a DW labeled as *<abbreviatory denotation of the type of test procedure> – main settings (<internal label of the current test section>)* will open which has different appearances in dependency on the type of test procedure that has been addressed to the currently selected test section. This window and the further approach to setting up the properties will be described in chapter 7, to be more precise in the respective subsection belonging to the regarding test procedure.

6 Performing a test run

6.1 Initializing and starting a test run

Right before a test run's performance only a few parameters have to be set by the experimenter. This is done on the panel *next test run* which is placed on the main window's desktop (see. sec. 4.2). The basic run parameter settings refer to the specification of the subject's ID and the input of the session key which is associated with certain experimental conditions (cp. sec. 5.3). Both actions have to be performed in the sub-panel *run parameters*. Furthermore one can choose only to perform a subset of test sections by making a selection using the sub-panel *execution of test sections*. The test run is started by actuating the button *Start run*. After this the GUI system for the subject's guidance and the collection of the data appears which will be described in the following section.

6.2 The subject's GUI system

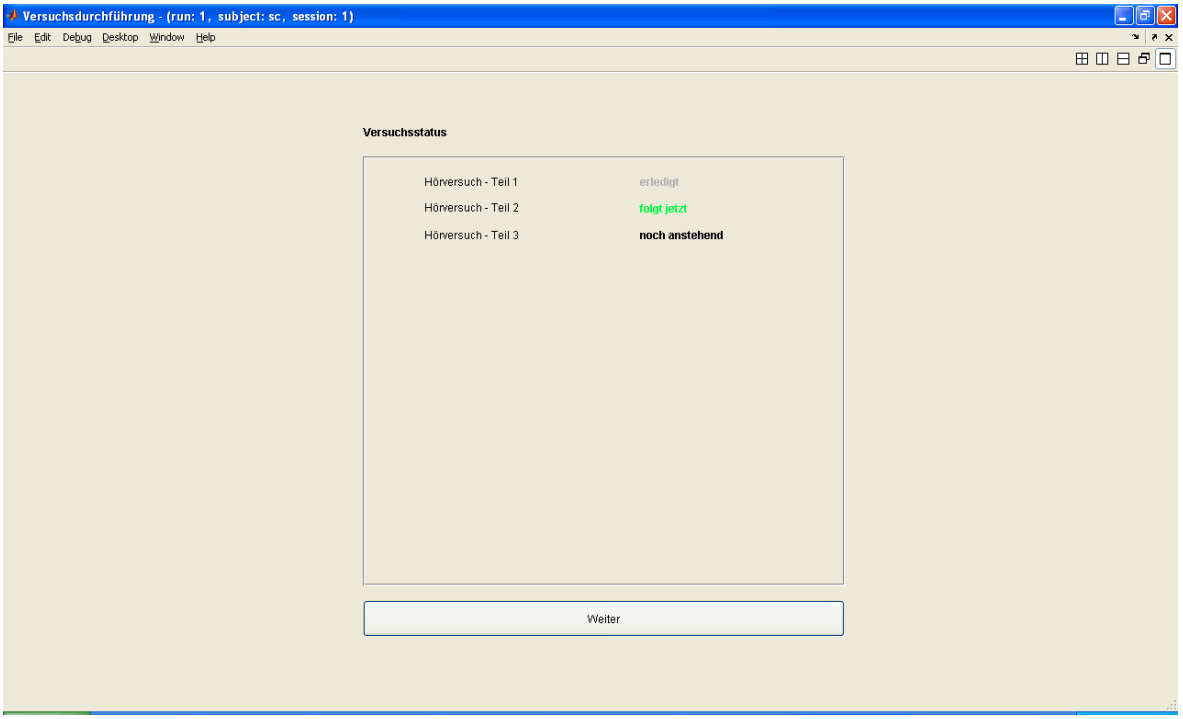

Figure 6.1: GUI system for subject guidance – basic screen showing information on the test run's status of progression.

The basis of the subject's GUI system is constituted by a window which is maximized to cover the whole screen and which stays in the background during the whole test run's performance.

This window is labeled as *Versuchsdurchfhrung*[11] and in the following it will be called the subject's *basic screen*. The two sorts of information alternately depicted on this window refer to the test run's current status of progression and to terms of instructions. The first is passed on to the subject in the beginning of a run and after each test section, by displaying a list of the synonyms which have been assigned to the single test sections by the experimenter before (see sec. 5.4). Those list items that represent former sections are printed in grey literals, the one belonging to the next section is printed in green and the remaining ones in black (cp. fig. 6.1). After the subject has clicked on the *Weiter* button, the information about the status of the test's progression is interchanged by the output of terms of initial instructions (see fig. 6.2) which have to be determined by the experimenter when specifying a procedure's properties (see chap. 7). After reading through the text the subject has to apply the *OK* button and then will arrive at a window which is specifically attributed to a certain test procedure, i.e. to the one that has been assigned to the next test section. These kinds of windows will be described in chapter 7.

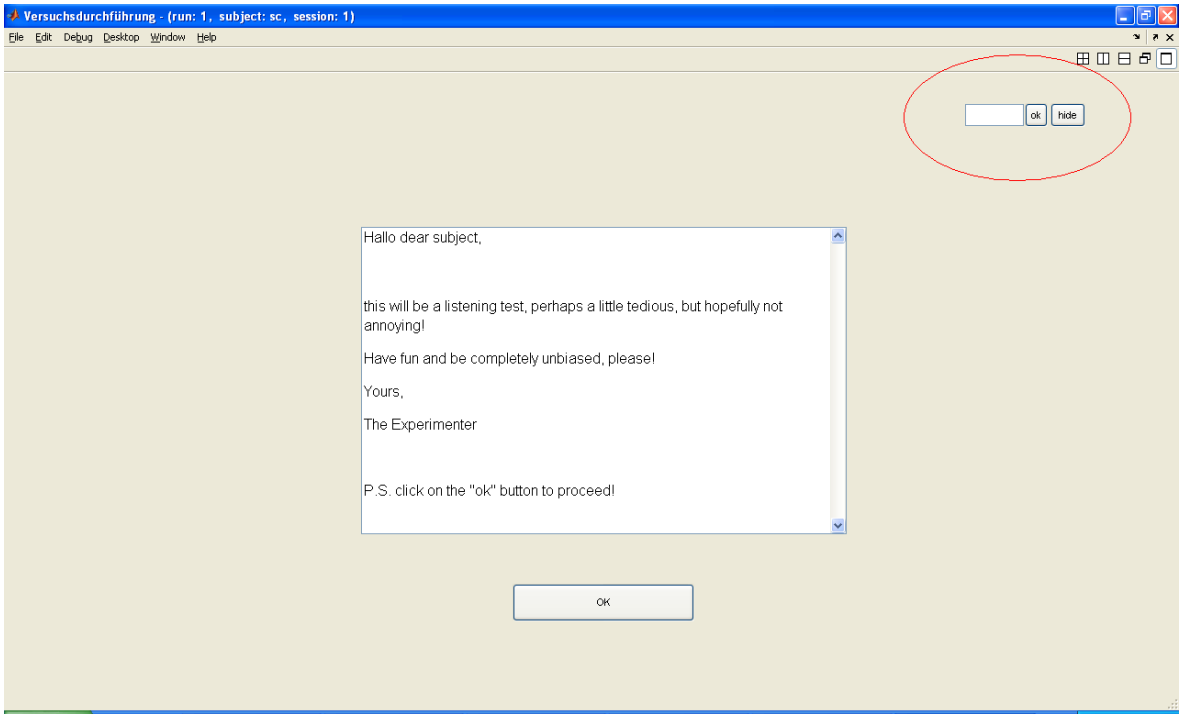

Figure 6.2: GUI system for subject guidance – *basic screen* showing the initial instructions. In the right corner on top (circled in red) one can see the normally hidden input field which appears if one actuates the window's close icon. When entering the word "break" and affirming by using the *ok* button one will break the test run and return to the program's main window.

Note: As already mentioned above, the subject's basic screen will stay maximized in the background during a test run. To avoid aborts caused by subjects performing faulty operations, the close icons of all windows belonging to the subject's GUI system have been cut off in their functionality. If it is required to break a run which already began, a controlled way is provided by taking the subject's basic window to the foreground and clicking on the standard *close* icon placed at the right corner on top. This will make an input field appear closely beneath (see fig. 6.2) for entering a command which is determined by the word "break".

[11] All information provided to the subject will be presented in German language as this software is primarily designed for local usage. If this needs to be changed, one has to modify the source code and is therefore directed to the technical documentation (see sec. 10.3).

Actuating the *ok* button will close the subject's GUI system and make the user return to the program's *main window*. In such a case an inclusion of the empirical data collected up to this point into the test series's data set sometimes might not be wanted. Deleting this data may be accomplished by using the menu item *edit\empirical data* (see sec. 4.1).

7 Test procedures – configuration and execution

7.1 Adaptive psychophysical procedures (AP)

7.1.1 The test run

Figure 7.1: The subject's GUI for the performance of an adaptive psychophysical procedure.

The design of the GUI for collecting the subject's responses depends on the selected response paradigm. At the program's current state one can use the nAFC-paradigm with $n = 2, 3$ *or* 4. Figure 7.1 shows the interface for a presentation of three stimulus intervals. It generally comprises n squared icons arranged on a horizontal line representing the stimulus intervals. The first stimulus' playback automatically starts in the beginning of each trial. To allow for a short delay the experimenter can determine a suitable time period (see sec. 7.1.2). A further period of silence can be defined for in between the presentations of the stimulus intervals. During the playback the respective icon which carries the current interval's number is highlighted to supply the subject visual information which should help assigning the current perception to the correct interval. After all stimuli of one trial have been played, according to the paradigm the subject now has to select one of the n intervals that is either guessed or supposed to contain the target stimulus. This is done by using the mouse pointer and clicking on the respective icon. At the same time this will make the procedure proceed with the next trial.

The placing of the stimulus intensities is controlled by the selected adaption mechanism. For the latter one can choose between several types of adaptive procedures which already have

been shortly described in section 2.2. A test run, i.e. more precisely a test section, can either contain one adaptive track or an arbitrary number of ones interleaved using the same response paradigm.

7.1.2 Setting up properties

The configuration of an adaptive procedure can be divided into several hierarchical "layers" which do sort of imply a stepwise processing. This is because some of the properties to be set on lower "layers" are affected by those defined on higher ones, but never the reversed case appears. The top "layer's" settings refer to the response paradigm, the time period between trials, terms of instructions and the laying of the tracks one wants to run. These settings are performed on the DW *AP – main settings (...)* (see fig. 7.2) which can be accessed as explained in subsection 5.5 and will be described below. If several tracks have been created one has the option either to randomly interleave all these tracks or to define subsets of tracks which are then assigned to different sessions. If a subset does comprise more than one track these will be randomly interleaved. By selecting a single track from the respective list-box and actuating the *edit* button, one gets to the DW *AP – track settings (...)* (see fig. 7.3) which represents the next lower "layer" which is for determination of the track's settings. These include specifying a label, setting the range of stimulus intensities, assigning the stimuli from the stimulus pool to the single intensity values and choosing the desired adaption and estimation mechanism out of five categories – Staircase, PEST, ZEST, QUEST and (manually configured) ML/Bayesian procedure. By clicking on the *settings* button belonging to the selected mechanism, one arrives at the third "layer", which refers to the mechanism's parameters. Setting up those will be described in the regarding paragraphs below. But before, there will be given a description of the DWs referring to the first two "layers".

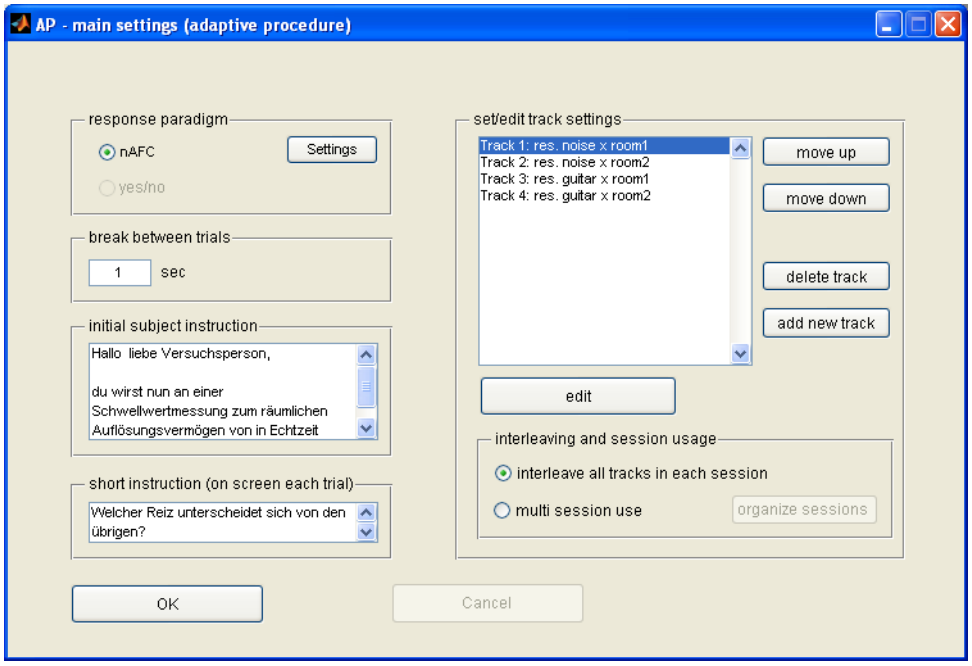

Figure 7.2: DW *AP – main settings (...)*

DW *"AP – main settings (...)"*

response paradigm (*panel*)

nAFC (*radio button*) If selected: selection of the nAFC paradigm.

yes/no (*radio button*) <this option is not available at the program's current state>

settings (*button*) Opens a window for specification of the response paradigm's settings. For the nAFC paradigm these are the number n of stimulus intervals, which can either be 2, 3 *or* 4 (default: 3) and the time period of silence between the intervals in seconds (default: 1 sec).

break between trials (*panel*)

<no label> (*input field*) Space for entering the time period in seconds that lies between the subject's response and the initiation of the next trial's first stimulus' playback (default: 2 sec). The input will be treated as a double value.

initial subject instruction (*panel*)

<no label> (*multi-lined input field*) Space for entering terms of instructions that will be displayed on the subject's *basic screen* (see sec. 6) before the current test section is started.

short instruction (on screen each trial) (*panel*)

<no label> (*input field*) Space for entering a short term of instruction that will be displayed on top of the subject's user interface in each trial.

set/edit track settings (*panel*)

<no label> (*multi-columned list box*) This list shows all the tracks that have been defined within the selected test section and across all sessions. The first column views the track number and the second its label name. Thereby a track's number is only important for its identification. Just for reasons of clearness there exists the possibility to shift a selected track to another list position by using the *move up/down* buttons (see below).

edit (*button*) Opens a DW labeled as *AP – track settings (...)* and described below, for the specification of properties like the track's label name, the definition of the stimulus intensity's range and the choice of an appropriate adaption and estimation mechanism.

move up/down (*button*) Shifts the selected item (i.e. track) to a proximate list position above/below.

delete track (*button*) Deletes the selected item (i.e. track).

add new track (*button*) Adds a new item (i.e. track) to the list's end.

interleaving and session usage (*sub-panel*)

interleave all tracks in each session (*radio button*) If selected, all defined tracks will be performed across all (globally) defined (cp. sec. 5.3) sessions in an randomly interleaved manner. *Note:* If the list consists of only one item (which is supposed to be a rather common scenario), this single track will be performed in the "normal" (i.e. non-interleaved) manner (across all defined sessions).

multi session use (*radio button*) If selected, in each run a subset of tracks which has been defined by the experimenter and assigned to a certain session key number, will be performed in a randomly interleaved manner (exclusively within the respective session). *Note:* If a subset consists only of one track, this

single track will be performed in the "normal" (i.e. non-interleaved) manner (within the belonging session).

organize sessions (*button*) If the respective option has been chosen, this button opens a DW that allows for defining subsets of tracks with each of them assigned to a certain session.

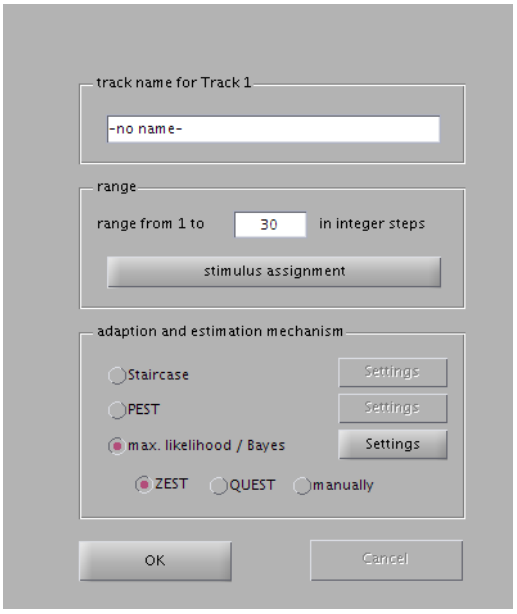

Figure 7.3: DW *AP – track settings (...)*

DW *"AP – track settings (...)"*

track name for Track <no. of selected track> (*panel*)

<no label> (*input field*) Space for determining a label name for the selected track.

range (*panel*)

<no label> (*input field*) Space for entering the maximum stimulus intensity[12] which has to be an integer value greater than 1. The stimulus intensity's range then is defined from 1 to this value in steps of 1. *Note:* Usually stimulus intensities are scaled in a specific unit and one wants to realize a different resolution than integer steps. In this (normal) case one has to apply a transformation to the physical domain outside the software with the discrete values kept in a look up table. Basically stimulus intensities should be "chosen along some scale that yields approximately equal sensory intervals" [Cornsweet, 1962]. At least for the application of parametric methods (e.g. in ML/Bayesian procedures) a certain shape of the psychometric function is assumed [Treutwein, 1995]. *Further note:* This parameter affects some properties of lower configuration layers, i.e. several of those that depend on the range, e.g. the signals assigned to the stimuli and properties of the selected adaption and estimation mechanism such as initial values, position and shape parameters, etc.. So if this parameter's value will be changed after the

[12] For simplification this parameter by itself sometimes is called *Range*. Note that this denotation will appear in some of the following statements and formulas.

mechanism's properties have already been set, some of the latter could be changed or overwritten by default values. It is therefore strongly recommended that in this case one repeatedly checks these properties to be on the safe side.

stimulus assignment (*button*) Opens a DW for assigning the stimulus signals from the stimulus pool to the discrete integer values of stimulus intensities. This includes a reference stimulus for which also an intensity value can be specified, which has rather to be seen as a label and won't be used by the program at the current state for any calculation, but might be helpful for documentation and evaluation purposes.

Adaption and estimation mechanism (*panel*)

Staircase (*radio button*) Selection of the *Staircase* procedure for placing stimuli and the threshold's estimation.

PEST (*radio button*) Selection of the *PEST* procedure for placing stimuli and the threshold's estimation.

max. likelihood/Bayes (*radio button*) Selection of *ML/Bayesian* procedures, based on the Best-PEST but including extensions taken from several similar procedures; selecting this enables the selection of one of the following three choices:

ZEST (*radio button*) Selection of the *ZEST* procedure for placing stimuli and the threshold's estimation; this preselects the appropriate options of the ML/Bayes based procedures for ZEST and disables the others.

QUEST (*radio button*) Selection of the *QUEST* procedure for placing stimuli and the threshold's estimation. This preselects the appropriate options of the ML/Bayes based procedures for QUEST and disables the others.

manually (*radio button*) Selection of *ML/Bayes* procedures for placing stimuli and the threshold's estimation; options from all extensions implemented in WhisPER are available to the user (yet no obvious guidance how to sensibly combine them).

settings (*button* (3x)) Opens a DW for specifications on the selected mechanism, labelled as *AP – edit <selected type of adaptive procedure> settings (...)* (see below).

7.1.2.1 Staircase

DW ***"AP – edit staircase settings (...)"***

stimulus domain (*panel*)

halve step size (*sub-panel*)

no (*radio button*) If selected, there will be no decrease of step size during a run.

after the following reversals (*radio button*) If selected, the step size will be halved after a set of reversals which has been determined by the experimenter.

<no label> (*input field*) Space for entering a set of numbers defining the track's reversals at which the step size should be halved. The single values should be integers greater than 1 and at most equal to the maximum number of reversals to be performed (see panel *termination criterion*). Each value must appear only once. The items have to be separated by space. Let's say one wants to achieve bisections after the 2., 4. and 6. reversal appearing during the run.

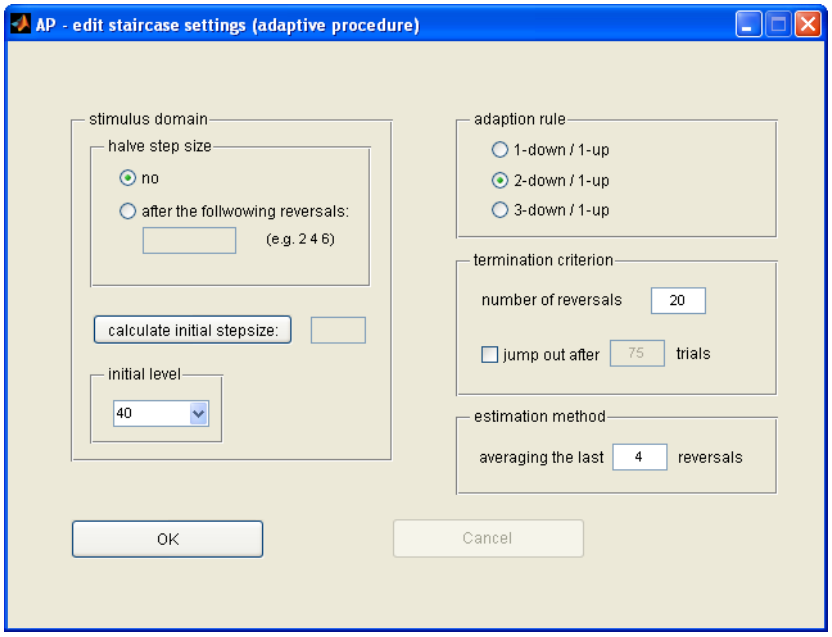

Figure 7.4: DW *AP – edit staircase settings (...)*

Then one has to enter "2 4 6". *Note:* Be careful not to insert a space after the last item!

calculate initial step size: (*button*) Calculates the size of the first step. This value directly depends on the number of bisections defined before. For example if one has chosen the reversals 2, 4, and 6 for halving the step size, the initial step size will be 8, as the minimum step size is always 1. *Note:* This function has no direct effect on the algorithm underlying the procedure. Its purpose only is to facilitate the configuration process. It should prevent the experimenter from defining too many bisections which might result in an absurd initial step size that is comparable to or even exceeds the range.

<no label> (*output field*) Space for viewing the initial step size's value which is calculated after actuating the respective button (see above).

initial level (*sub-panel*)

<no label> (*drop-down list*) Here the initial value of the stimulus intensity can be selected. By default this is set to the range's maximum value.

adaption rule (*panel*)

1-down/1-up (*radio button*) If selected, a simple 1-down/1-up adaption rule will be applied.

2-down/1-up (*radio button*) If selected, a transformed 2-down/1-up adaption rule will be applied.

3-down/1-up (*radio button*) If selected, a transformed 3-down/1-up adaption rule will be applied.

termination criterion (*panel*)

number of reversals (*input field*) Space for entering the maximum number of reversals which will be performed and after that the procedure will be terminated.

jump out after ... trials (*check box & input field*) If the box is checked, one may specify a maximum number of trials that will be performed and after that the procedure

will be terminated if the maximum number of reversals has not been reached yet. *Note:* This additional termination criterion was installed in order to avoid unreasonably long trial sequences. If this criterion catches and the first does not, the threshold's estimate will not be calculated as the estimation method (see panel below) then cannot be applied. In this case 0 is returned as the output. (Under certain conditions the calculation of a threshold estimate might be caught up manually outside the program by applying a slightly different scheme, e.g. using previous reversals for averaging stimulus intensities.)

estimation method (*panel*)

averaging the last ... reversals (*check box & input field*) Space for entering the number of last consecutive reversals whose associated stimulus intensities will be averaged for calculating the threshold's estimate. *Note:* This number must not exceed the track's maximum number of reversals specified before (see panel *termination criterion*). Otherwise the algorithm will not work properly. From a methodical point of view, i.e. in order to maximize accuracy and minimize bias, averaging should only cover an even number of reversals including only those that appear after the minimum step size has already been reached.

7.1.2.2 PEST

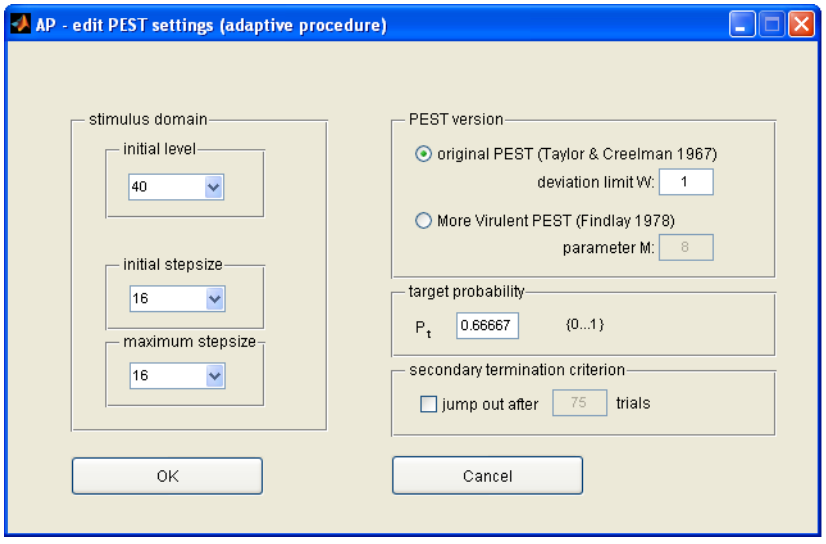

Figure 7.5: DW *AP – edit PEST settings (...)*

DW *"AP – edit PEST settings (...)"*

stimulus domain (*panel*)

initial level (*sub-panel*)

<no label> (*drop-down list*) Here the initial value of the stimulus intensity can be selected. By default this is set to the range's maximum value.

initial step size (*sub-panel*)

<no label> (*drop-down list*) Here the size of the first step can be selected. By default this is the second largest value that is possible in the case, i.e. the second largest power of two that fits in the range.

maximum step size (*sub-panel*)

<no label> (*drop-down list*) Here the maximum step size can be selected. This value will never be exceeded throughout the run no matter what value the rules call for. By default it is set to the second largest possible value, i.e. the second largest power of two that fits in the range. *Note:* As this parameter already affects the first step, its value should be at least as high as the initial step size.

PEST version (*panel*)

original PEST (Taylor & Creelman 1967) (*radio button*) If selected, the original PEST procedure described by Taylor & Creelman [Taylor and Creelman, 1967] will be performed.

deviation limit W (*input field*) Space for entering the value of the parameter called *deviation limit*. The authors of the PEST procedure suggest a value of $W = 1$ if one wants to yield a target probability of $P_t = 0.75$ and higher values like $W = 1.5$ or $W = 2$ for target probabilities closer to 0.5.

More Virulent PEST (*radio button*) If selected, a modified PEST procedure described by Findlay (1978) [Findlay, 1978] will be performed.

parameter M (*input field*) Space for entering the value of the parameter M. Sadly, the author gives no unambiguous recommendations on this parameter's choice. By default it is set to 8, as this is a moderate value used in the author's simulations.

target probability (*panel*)

P_t (*input field*) Space for entering the value of the target probability $P_t = \psi(\vartheta)$, i.e. the probability for a positive response at convergence. This value has to lie between 0 and 1 and should be chosen in dependency on the reponse paradigm. A reasonable choice is sometimes made by using Abott's formula (based on the classical threshold concept [cp. Gescheider, 1997, chap. 4]) to calculate the ordinate's value $\psi(x)$ of the psychometric function:

$$\psi(x) = \gamma + [1 - \gamma - \lambda] \cdot \psi^*(x) \tag{7.1}$$

Thereby $\psi^*(x)$ is the probability of sensory discrimination, λ is the lapsing rate and γ the guessing rate or false alarm rate. At convergence $\psi^*(\vartheta)$ is usually defined as 50%. In case of the nAFC paradigm there is $\gamma = \frac{1}{n}$. Hence by default the target probability is set to the respective value assuming $\lambda = 0$, i.e.

$$P_t = \frac{1}{n} + \left[1 - \frac{1}{n}\right] \cdot 0.5 \tag{7.2}$$

secondary termination criterion (*panel*)

jump out after ... trials (*check box & input field*) If the box is checked, here one may enter the maximum number of trials that will be performed and after that the procedure will be terminated if this has not been done by the PEST rules yet. (The procedure regularly terminates when the minimum step size of 1 is underrun.) *Note:* This additional termination criterion was installed in order to avoid unreasonably long trial sequences. If this criterion catches and the first does not,

the threshold's final estimate cannot be properly calculated and is therefore set to 0. In this case one might want to look for the stimulus intensity that would have been taken for the next placement. This value can either be found in the log file (see sec. 10.1.5) or in the track's plot (see sec. 10.1.6). Note that this value comprises a different accuracy, as the minimum step size has not been under-run and possibly not even been reached yet.

7.1.2.3 ML/Bayesian procedure

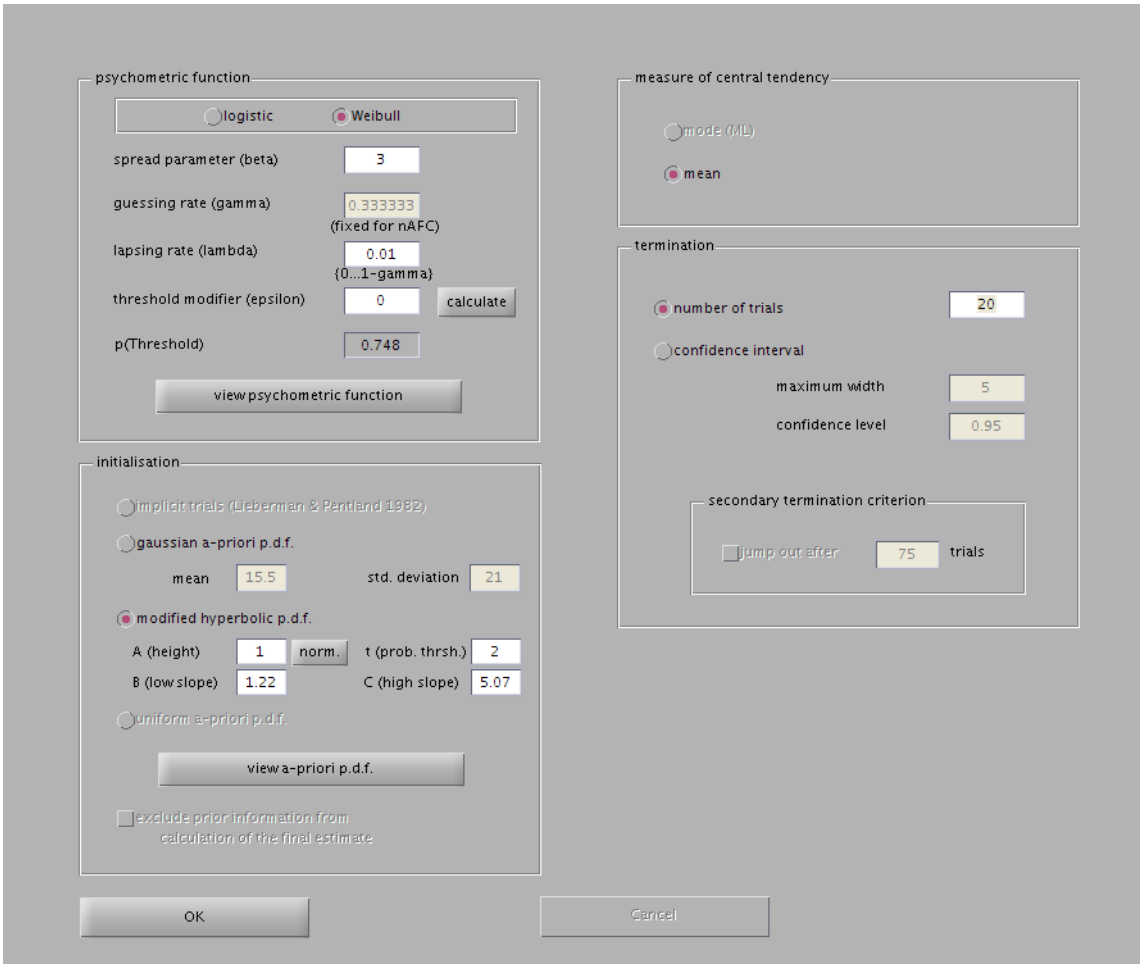

Figure 7.6: DW *AP – edit ZEST settings (...)*

DW *"AP – edit ZEST settings (...)"*
DW *"AP – edit QUEST settings (...)"*
DW *"AP – edit ML/Bayes settings (...)"*[13]

psychometric function (*panel*) The shape of the underlying psychometric (model) function $\hat{\psi}(x)$ is described by one of the following two equations, which can be derived from equation (7.1) on page 32 by assuming a certain function for $\psi^*(x)$.

[13] This DW's title depends on your previous choice of ML/Bayes based procedures, cf. sec. 7.1.2. The same goes for the availability of certain options of this DW.

logistic (*radio button*) The function used is the logistic function $l(x;\alpha,\beta) = \frac{1}{1+e^{\frac{-(x-\alpha)}{\beta}}}$. This yields:

$$\hat{\psi}(x) = \hat{\psi}(x;\,\theta,\beta) = \gamma + (1-\gamma-\lambda)\cdot\frac{1}{1+e^{\frac{-(x-\theta)}{\beta}}} \tag{7.3}$$

Thereby λ is the lapsing rate and γ the guessing rate or false alarm rate which is assumed to be $\gamma = 1/n$ for the nAFC paradigm. θ is the unknown position parameter (which in case of a logistic model function is equal to the threshold) and β the spread parameter (see further below).

Weibull (*radio button*) The function used is the Weibull function[14], yielding:

$$\hat{\psi}(x) = \hat{\psi}(x;\,\theta,\beta) = \gamma + (1-\gamma-\lambda)\cdot exp[-10^{\beta(x-\theta+\epsilon)}] \tag{7.4}$$

Again λ is the lapsing rate, γ the guessing rate (assumed to be $\gamma = 1/n$ for the nAFC paradigm), θ the threshold and β the spread parameter. ϵ gives the possibility of shifting what point of the psychometric function is used as threshold, thereby influencing the threshold criterium (see below). Fields for input of λ, γ and β are "shared" with the fields for the corresponding input for the logistic function (see below).

Please note that one of the mandatory requirements for psychometric functions is that for different conditions it doesn't change shape, but only its position along the x-axis, see e. g. Watson and Pelli [1983]. The Weibull function alas specifically meets this criterion only under the condition that the x-axis is scaled logarithmically, see e. g. [Treutwein, 1995]. Thus the Weibull function is best suited for use as psychometric function in those cases, where the relationship of the tested stimuli's intensities is logarithmic, too.

spread parameter (beta) (*input field*) Space for entering the value of the spread parameter β which should be chosen based on prior knowledge. Sometimes it might be helpful to consider its relationship to the psychometric function's slope S:

$$S = \frac{(1-\gamma-\lambda)}{4\beta}. \tag{7.5}$$

In other cases one might note the relationship to the spread σ_l of the logistic function l, which is approximately given by

$$\sigma_l \approx 1.81\cdot\beta \tag{7.6}$$

[Hastings and Peacock, 1975]. If one has prior knowledge about a psychometric function being described by a cumulative normal ogive then one might use the relationship

$$\sigma_N \approx 1.7\cdot\beta \tag{7.7}$$

with σ_N being the ogive's standard deviation, as for this the logistic function can be seen as a fairly good approximation to the cumulative normal [Treutwein, 1995, textbox 2]. If the spread parameter's true value is unknown the authors of Best PEST recommend rather to use a larger than a smaller value [Lieberman and Pentland, 1982]. By default it is set to $\beta = Range/10$ (cp. panel *termination*, option *confidence interval*). (This parameter is used by both the logistic as well as the Weibull function.)

[14] Used here in that of its many forms which is used by ZEST.

guessing rate (gamma) (*input field*) Space for entering the guessing rate (nAFC paradigm) or false alarm rate (yes/no paradigm). At the program's current state one only may use the nAFC paradigm and so this value is fixed to $1/n$ and therefore the respective field is not editable.(This parameter is used by both the logistic as well as the Weibull function.)

lapsing rate (lambda) (*input field*) Space for entering the lapsing rate λ. In the original procedure this parameter was not included in the psychometric model function, i.e. it was implicitly set to zero. As some scientists recommend to choose a value of about 1% [eg. Hall, 1981; Harvey, 1986; Klein, 2001], this one is selected for default setting. Although it should be stated that the true lapsing rate depends on the individual subject and therefore is unknown [cp. Treutwein, 1995]. *Note:* Due to the definition of the underlying psychometric model function (cp. eq. (7.3) on p. 34 and eq. (7.4) on p. 34) this value naturally has to be selected from the interval $[0,\ 1-\gamma]$. Otherwise the algorithm cannot work properly.(This parameter is used by both the logistic as well as the Weibull function.)

threshold modifier (epsilon) (*input field*) Space for entering the threshold modifier ϵ, only activated when the Weibull psychometric function is selected. This parameter shifts the psychometric function along the x-axis and therefore influences which point of the psychometric function is used for testing. This does not have an effect on the accuracy of obtainable threshold measurements' results, but it does have on the procedure's efficiency, see King-Smith et al. [1994]. King-Smith et al., 1994 suggest that for tests running between 8 and 16 trials long ϵ be set to 0, which is also the default value used in WhisPER. For tests lasting longer than 16 trials they assess the greater efficiency to be gained setting ϵ so that the testing occurs at the *ideal sweat factor* (see below), which is what Watson and Pelli, 1983 used.

calculate (*button*) Automatically calculates ϵ so that testing occurs at the *ideal sweat factor* of the psychometric function of the present values. The *sweat factor* is a measure of efficiency describing the relationship between the amount of "work" to obtain a certain result and the accuracy of said result. For the psychometric function it is calculated as the square of the ratio of the binomial standard deviation and the slope of the psychometric function at the point in question (Watson and Pelli [1983], Taylor [1971]):

$$K(X) = \frac{\Psi(X)[1-\Psi(X)]}{[d\Psi(X)/dX]^2} \qquad \text{with } X = x - \theta \tag{7.8}$$

The *ideal sweat factor* is represented by the minimum of $K(X)$. The calculation is done iteratively, the result is automatically written to the ϵ *input field*. This *button* is only activated when the Weibull function is chosen as psychometric function.

p(Threshold) (*output field*) Displays the answering probability of the psychometric function at the testing point. This can be useful for comparing probabilities derived from optimisation of ϵ (see above) with probabilities of more traditional threshold criteria.

view psychometric function (*button*) Opens a window containing a plot of the psychometric function's shape which has been calculated for the previously specified parameter settings. *Note:* This feature does not directly influence the algorithm underlying the procedure. Its purpose only is to facilitate the configuration as it might supply the experimenter a tool for a visual verification of the choices made on the parameters' values.

initialisation (*panel*) For a discussion of the different possibilities that exist to do the initialisation see for example [Pokorny, 1998].

implicit trials (*radio button*) If selected, the procedure will be initialized by defining a data set of stimulus intensities and corresponding (presumed) responses which is sometimes referred to as *implicit trials*. This is done in the way it is described by the authors of Best PEST, Lieberman and Pentland, 1982.

gaussian a priori p. d. f. (*radio button*) If selected, the procedure will be initialised with a (normalised) gaussian a priori probability density function $\mathcal{N}(x;\, \mu, \sigma)$, with μ being its mean and σ the standard deviation.

mean (*input field*) Space for entering the mean μ of the gaussian a priori probability density function. By default μ is set to the middle of the range, i.e. $\frac{(1+Range)}{2}$.

std. deviation (*input field*) Space for entering the standard deviation σ of the gaussian a priori probability density function. The experimenter should take care that this function's graph is relatively flat compared to the final p. d. f. so that the latter never will be dominated by the former [Treutwein, 1995; Martz and Waller, 1982, p. 184]. By default σ is set to $0.7 \cdot Range$ which is a value that has no theoretical justification but is intuitively chosen to yield such a flat curve.

modified hyperbolic p. d. f. (*radio button*) If selected a modified hyperbolic secant is used as a priori probability density function. Suggested by the authors of ZEST, King-Smith et al., 1994, they used this function to fit a p. d. f. to histograms of thresholds measured in earlier experiments. It takes the form of

$$q_0(\theta) = \frac{A}{B \cdot e^{-C(\theta - t)} + C \cdot e^{B(\theta - t)}} \tag{7.9}$$

with θ being (logarithmic) threshold, A being the overall height of the curve, B its lower and C its upper slope and t being the most probable threshold, the function's maximum (i. e. t shifts the function along the x-axis). The advantage of using a function like this is that in contrast to all other initialization options provided here it is capable of describing asymmetric threshold distributions, which – seeing that a modified hyperbolic p. d. f. has been reached fitting a curve to real data from earlier experiments – obviously do exist.

A (height) (*input field*) Space for inputting the overall height of the curve. As the height of the a priori p. d. f. is usually not relevant to the correct working of e. g. the ZEST or the QUEST procedure (c. f. King-Smith et al., 1994), this is simply set to 1 per default. As the function's overall height is heavily affected by both slope parameters, please see below for a remedy.

norm. (*button*) Pushing this button will calculate the height parameter for the curve to have a maximum of 1. As the height is usually really not important to the working of the procedures, this is simply for convenience, e. g. when viewing the generated curve via the *view a-priori p. d. f. button* this simply makes sure the whole graph is visible.

B (low slope) (*input field*) Space for entering the slope of the curve in the range of lower threshold values. For lack of different approaches this is simply initialized to 1.22, the value King-Smith et al., 1994 used. Please note that although they expressed their confidence in the possibility that their modeled a priori p. d. f. might be of universal nature, they warned at the same time about it

possibly not being usable in the same way for every possible psychometric measurement. "Wrong" values entered for the a priori p. d. f. should not harm the accuracy of the test, however, but only its efficiency (cf. King-Smith et al., 1994).

t (prob. thresh.) (*input field*) Space for entering the value for the value of the function that is to be the most probable threshold. As mentioned above this parameter slides the function along the x-axis. This, too, is simply initialized to the value used by King-Smith et al., 1994, 2. If not fitted to actual data of your own it should usually better be set to $\frac{Range}{2}$, for example.

C (high slope) (*input field*) Space for entering the slope of the curve in the range of lower threshold values. For lack of different approaches this is simply initialized to 5.07, the value King-Smith et al., 1994 used. Please note the warning about this further above.

uniform a-priori p. d. f. (*radio button*) If selected, no prior information will be used. The first stimulus' intensity will be chosen from the middle of the range, i.e. $round\left(\frac{1+Range}{2}\right)$.

view a-priori p. d. f. (*button*) Opens a window containing a plot of the a priori p. d. f. which has been calculated for the selected variant of initialization. *Note:* This feature does not directly influence the algorithm underlying the procedure. Its purpose only is to facilitate the configuration as it might supply the experimenter a tool for a visual verification of the selected initialization approach and the choice of the respective parameters' values.

exclude prior information from calculation of the final estimate (*check box*) If checked, the prior information will be excluded from calculation of the final estimate as this has been recommended by some scientists [e.g. Watson and Pelli, 1983]. This is internally performed in a way that is equivalent to dividing the final p. d. f. by the a priori p. d. f. before selecting/calculating the average (i.e. the mode or the mean, cp. below). Hence this option will affect the first three approaches to initialization described above.

measure of central tendency (*panel*)

mode (ML) (*radio button*) If selected, the maximum of the likelihood function will be taken for both placing stimuli and the threshold's final estimation. (A procedure involving this approach basically can be called a *maximum likelihood* (ML) procedure as it is implied amongst others by the *Best PEST* [Lieberman and Pentland, 1982; Pentland, 1980].)

mean (*radio button*) If selected, the mean of the a posteriori probability density function (i.e. the likelihood, normalized to the area enclosed by the graph) will be taken for both placing stimuli and the threshold's final estimation.

termination (*panel*)

number of trials (*radio button & input field*) If selected, the procedure will be terminated after a fixed number of trials has been carried out. The latter must be entered in the corresponding input field. As ZEST has been proven to already converge among the first 20 trials (cf. Otto [2008]), selecting ZEST will set this value to 20. By default this parameter is set to 40, though this has been a rather arbitrary choice.

confidence interval (*radio button*) This dynamic stopping criterion leads to a termi-

nation if the current confidence interval[15] (at a predefined confidence level) of the threshold's estimation falls below a preset limit (which is specified by the input field *maximum width*). *Note:* A true confidence interval (as an absolute measure) can only be calculated if the true spread of the psychometric function is known. As this usually is not the case the probability intervals calculated in the given context can only be regarded as comparative measures. *Further notes:* The number of trials which are carried out till this criterion catches strongly depends on the setting of the spread parameter (see above). Higher values lead to a greater number of trials and vice versa. As a rule of thumb it is suggested in Treutwein [1995] to adjust one of the three parameters, the spread, the maximum width of the confidence interval and the confidence level, until the procedure stops after the following number of trials on average in dependency on the response paradigm: yes/no, 20; 4-AFC, 30; 3-AFC, 37.5 and 2-AFC, 50. By default the maximum confidence interval is set to $round(0.15 \cdot Range)$ and the level to 0.95.[16]

maximum width (*input field*) Space for entering an upper limit for the total width of the confidence interval belonging to the final estimate. Due to the range's resolution and the internal calculation of the current confidence interval, only positive integer numbers constitute sensible choices (i.e. any value from the half-open interval extended by an integer number and the next one above which is not included, results in the same behavior of the procedure as the integer by itself).

confidence level (*input field*) Space for entering the value of the confidence level (also called *confidence coefficient*). The value must be from the interval $[0, 1]$.

secondary termination criterion (*sub-panel*)

jump out after ... trials (*check box & input field*) This option is only available if a dynamic stopping criterion has been selected. If in this case the box has been checked, one can enter a maximum number of trials that will be performed and after which the procedure will be forcefully terminated if the dynamic criterion has not caught yet. *Note:* This additional termination criterion was installed in order to avoid unreasonably long trial sequences. If this criterion catches and the first does not, the threshold's final estimate cannot be properly calculated and is therefore set to 0. In this case one might want to look for the stimulus intensity that would have been taken for the next placement. This value can either be found in the log file (see sec. 10.1.5) or in the track's plot (see sec. 10.1.6).

[15] The calculation of the current confidence interval is performed as described in [Treutwein, 1995, eq. 23] and as carried out in the *YAAP* implementation (published in Treutwein [1997]; seen version 3.0, 1993/12/1, by courtesy of B. Treutwein).

[16] These values have been encountered in simulations which yielded the following numbers of trials on average (stimulus range stated in brackets, resolved in steps of 1): 26,1 (30); 34,4 (50); 47,0 (100). The simulations have been performed for the 3AFC paradigm and a spread parameter of $\beta = Range/10$. Note that criteria of validity like accuracy and bias thereby have not been under consideration.

7.2 Repertory grid technique (RGT)

7.2.1 The test run

The performance of this test procedure is split in two parts. The first consists of the elicitation of constructs by triad comparisons and the second of the process of rating the elements on scales whose endpoint descriptors are represented by right these constructs. In between the two parts there is a break that can be used by the experimenter for editing the elicited constructs, which can be necessary as there might be ambiguous or in some respect unusable phrases to be cleared or revised. Moreover decisions can be made with respect to the polarity of the scales.

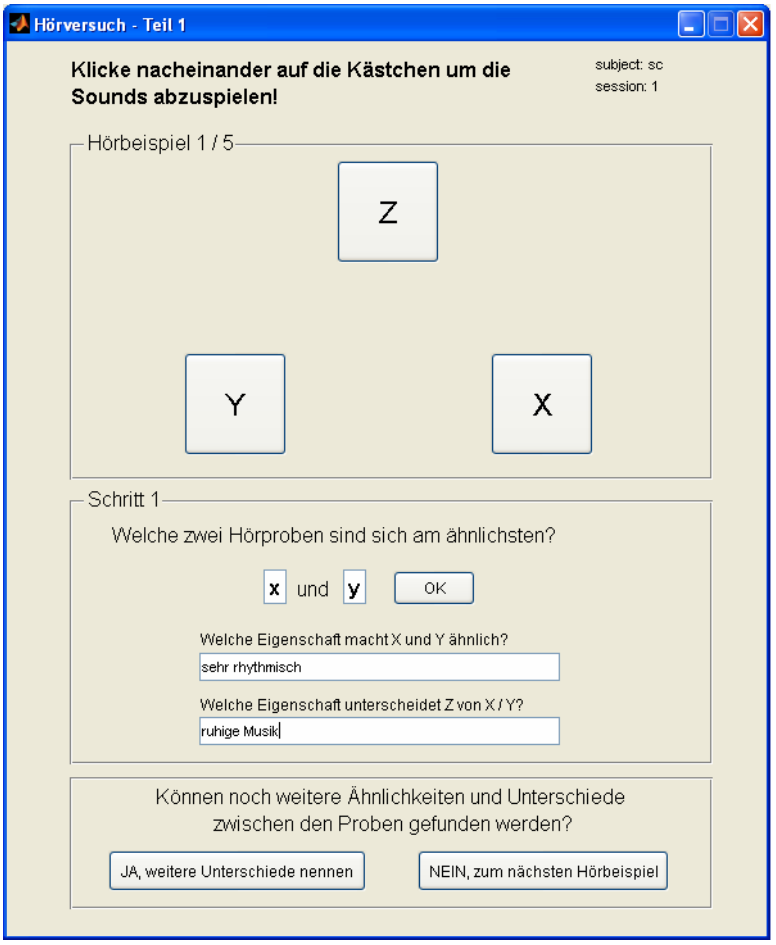

Figure 7.7: The subject's GUI for the performance of the RGT's first part, i.e. the triad comparisons.

Part I: elicitation of constructs The concerning GUI for the stimulus presentation and collection of the subject's responses is shown in figure 7.7. To play one of the triad's stimuli (called *elements* in case of RGT) the subject has to apply one of the three buttons denoted by *X*, *Y* and *Z*. The first task which is assigned to the subject is to allocate two stimuli which seem to be the most alike and to enter the labels of the corresponding buttons (i.e. X, Y or Z). The inputs must be confirmed by applying the *OK* button. This task was only designed to facilitate the formulation of the further two questions that properly matter. Those will be

presented only if each stimulus has been played for at least one time. The first of these questions requires a verbal description of the respect in that the very two stimuli which have been previously identified as the most similar ones are alike. The second asks for the difference between these two stimuli and the third.

After answering these questions, the subject has the choice between either reporting on further similarites and differences or to continue with the next triad's presentation. Note that progression in one of these ways is only possible if each stimulus has been played for at least one time and all questions have been answered, i.e. inputs have been made to the regarding fields. The first part of the procedure is ended by displaying a fixed standard term which instructs the subject to wait for the experimenter.

Break: Editing constructs After the experimenter applying the respective button (*nur vom Versuchsleiter zu bedienen*) a DW will appear which is labeled as *RGT – edit constructs* and described below. As the editing process is expected to be performed by the experimenter the language of the information displayed consistently is english – just like in case of any other dialogue which he is involved in. Possible modifications range from revising a construct's elicited verbal phrases, determining their assignments to either one of the rating scale's poles, deleting entire constructs, making decisions on the scale's polarity, i.e. its orientation when being posed during the rating process, and changing the order in that scales are posed one below the other. After editing work has been finished one has to apply the respectively named button to resume the current test run for the performance of the procedure's second part, i.e. the rating procedure.

DW *"RGT – edit constructs (...)"*

subject: ... (*text*) Displays the ID of the current subject.
session: ... (*text*) Displays the key number of the current session.
edit constructs for rating (*panel*)

<no label> (*multi-columned list box*) This list primarily shows the constructs arranged in the order of their elicitation events which is represented by the numbers placed in the first column. After modifications have been made with respect to the order of the list's entries (by using the *move up/move down* buttons, see below) these numbers then represent the order in which the constructs' scales will be posed one below the other during the rating process. The further columns view the poles' verbal descriptors and the scale's polarity, i.e. its orientation when being posed during the rating process. For the latter a "+" indicates that the low pole (i.e. the pole belonging to a minimum scale value) will be posed on the left side of the DW and the high pole (i.e. the pole belonging to a maximum scale value) on the right side. The reversed case is indicated by a "-". Note that at first poles may be not properly defined as the verbal descriptors' assignment either to the scale's minimum or maximum extension has been automatically done straight forward (i.e. the inital pole is always assigned to the low pole and vice versa). A correction then has to be performed manually by using the button *switch labels* which is described below. When all editing work has been completed the list must not contain more than 30 constructs. This restriction is due to the finite display resolution as a greater number would lead to problems with viewing all scales within one common window.

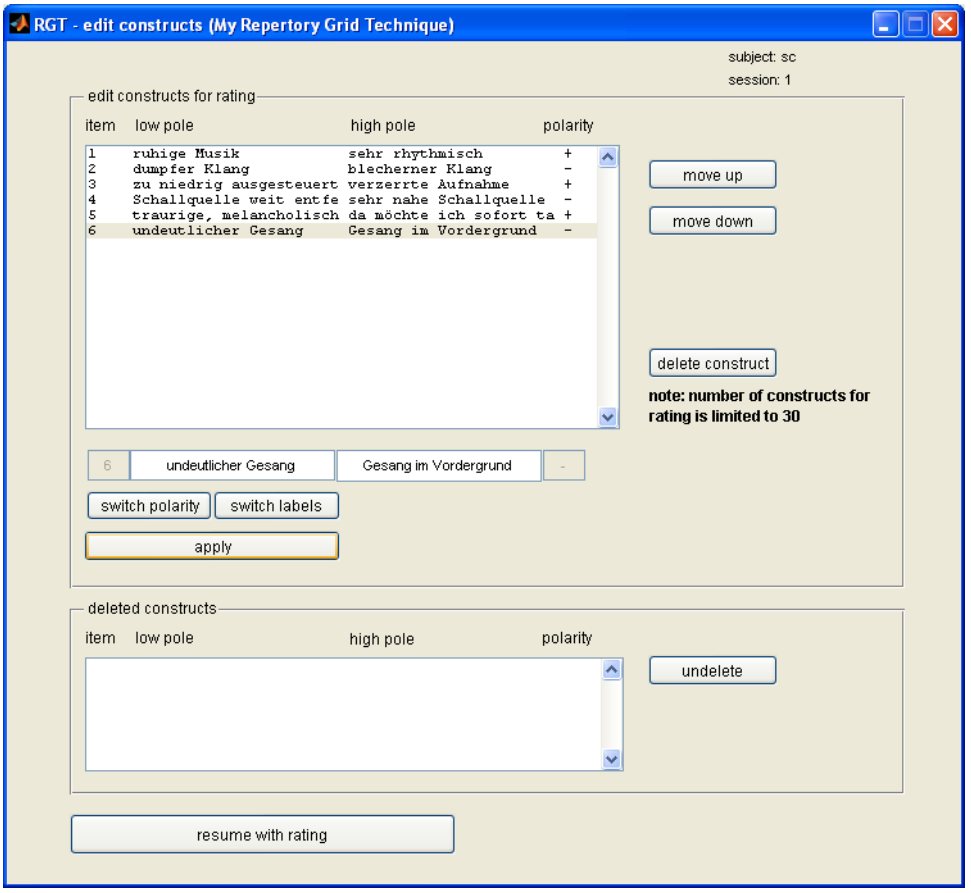

Figure 7.8: The DW *RGT – edit constructs (...)* offers the possibility to the experimenter for editing the constructs elicited in the first part.

<no label> (*output field* 1st col) Displays the number assigned to the selected construct.

<no label> (*input/output fields* 2nd & 3rd col) Spaces for displaying and editing the verbal descriptors of the scale's poles belonging to the selected construct.

<no label> (*output field* 4th col) Indicates the polarity of the scale belonging to the selected construct.

switch polarity (*button*) Switches polarity of the scale belonging to the selected construct.

switch labels (*button*) Switches verbal descriptors of the scale's poles belonging to the selected construct.

apply (*button*) Applies changes in the contents of the input/output fields above to the selected construct.

move up/move down (*button*) Shifts the selected construct to a proximate list position above/below.

delete construct (*button*) Deletes a selected construct which then is shifted to the list below.

deleted constructs (*panel*)

<no label> (*multi-columned list box*) This list views all constructs that have been deleted from the list above. The idea behind this is to prevent the experimenter from losing a construct by faulty operation, as the items contained in this list can

be reassigned to the list above by using the button *undelete* which is described below.

undelete (*button*) Reassigns the selected construct to the list above.

resume with rating (*button*) Resumption of the current test section by initiating the RGT's second part, i.e. the rating process. Note: make sure that all editing work has been completed before applying this button, as there will be no possibility to return to this DW afterwards.

Part II: Rating procedure Again this part is introduced by user-defined terms of instructions. After the subject actuating the *weiter* button the rating procedure begins.

The respective GUI is shown in figure 7.9. The stimulus' (i.e. *element's*) playback automatically starts in the beginning of each trial. To allow for a short delay the experimenter can specify a suitable time period (see sec. 7.2.2). The subject's ratings can be recorded from the beginning of a trial (even during the playback) either on discrete scales which are realized by radio buttons or on quasi-continuous scales using sliders[17]. Progression to the following trial is possible if the whole set of at most 30 constructs has been rated.

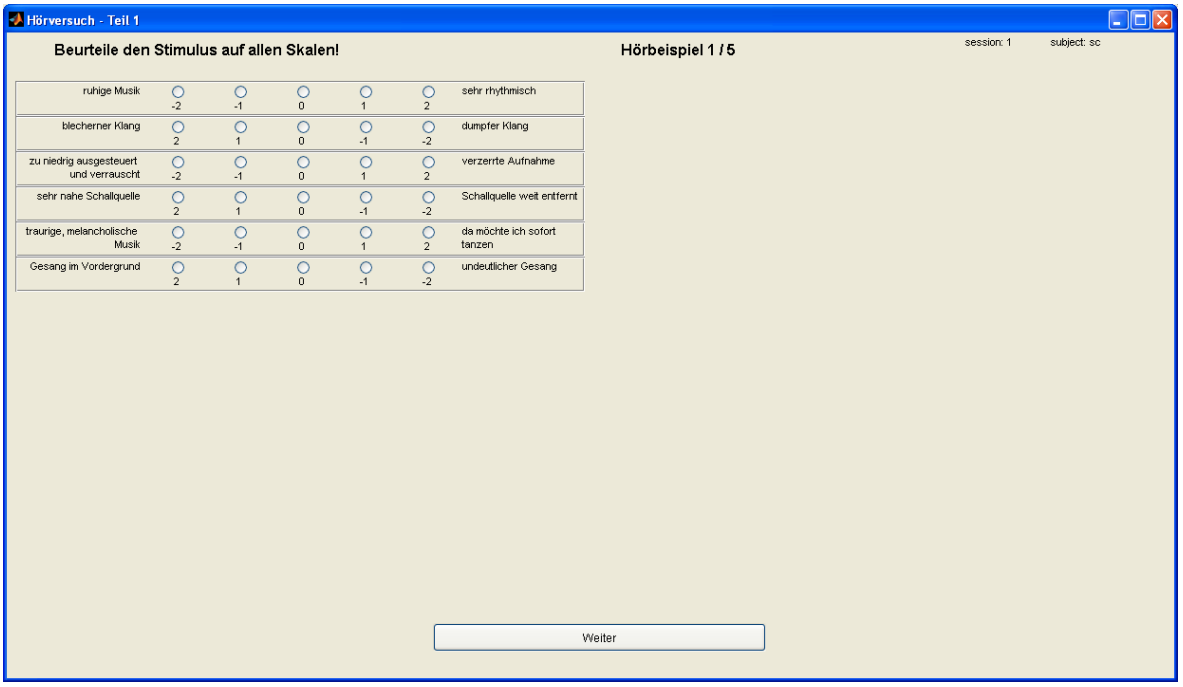

Figure 7.9: The subject's GUI for the performance of the RGT's second part, i.e. the rating of the elements on the constructs which have been elicited and edited before.

7.2.2 Setting up properties

Settings of the procedure's properties have to be determined on the DW *RGT – main settings (...)* which can be accessed as explained in section 5.5. Its control elements will be described below. The basis for the whole procedure is formed by the definition of elements which is done on the respective DW accessible by the button *element pool*. Thereby a set of stimuli

[17] See annotation 1 on page 9

from the stimulus pool can be selected and assigned to element key numbers, gathered in the *element pool*. Further settings are structured by the procedure's two parts and hence adjustable in two distinct panels.

If elements have been defined, the generation of the triads can be performed on the panel *Part I: construct elicitation* by applying the button *edit triads* which then opens a DW that allows either to manually assign elements to the triads' key numbers or to let the program automatically generate a complete variation of (at most 15) elements over triads which then may be edited manually. After the definition of the triads the order of their presentation to the subject has to be determined. This can be done in two basically different ways. Either one instructs the program to play all triads in a random order or one makes a selection of triads which is related to a certain session's key number and which can either be played in a determined sequence or again in a random succession. When choosing the second way one has to apply the regarding *edit* button to reach the respective DW labeled as *RGT – triad sequence setup (...)* which is described in more detail below. A further property that can be set is the maximum number of constructs the experimenter wants to emerge from one triad comparison (cp. sec 7.2.1). Moreover two different terms of instructions can be entered. One will be displayed in the beginning of the procedure's first part and the other one should only consist of a short phrase which will be placed on top of the subject's user interface in each trial.

Setting the properties of the second part has to be performed on the panel *Part II: rating procedure*. By applying the button *edit scales* one gets to the DW *RGT – edit scale format* which is described below and allows adjusting the scales' format settings such as the number of categories, the categories' numerical representatives, category labels if required, the choice between discrete and continuous scales and the scales' resolution in the latter case. Further settings on the performance of the rating process refer to the order of the elements' presentation and will be adjusted in a manner that is completely analogous to the approach already described above in the triads' context. Moreover the time period between the subject's actuation of the *weiter* button and the presentation of the next element can be fixed. Finally two terms of instructions can be entered. One will be displayed in the beginning of the second part and one should only consist of a short phrase which will be shown on the top of the subject's user interface in each trial.

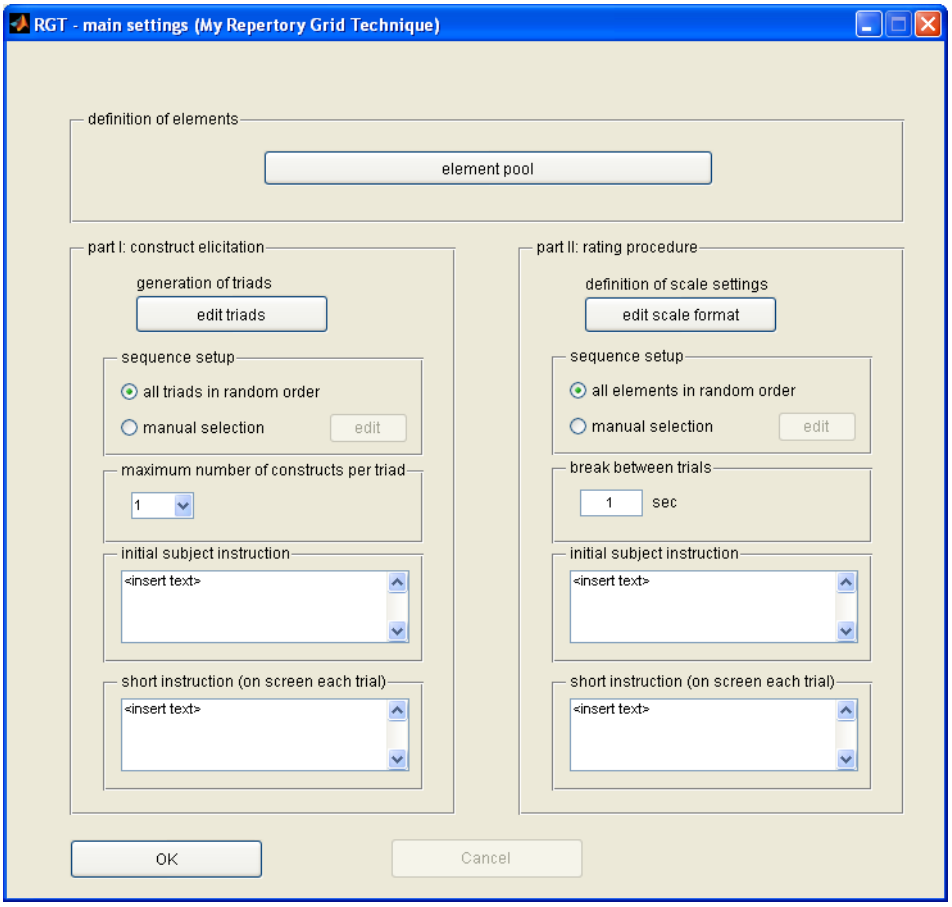

Figure 7.10: DW *RGT – main settings (...)*

DW *"RGT – main settings (...)"*

definition of elements (*panel*)

element pool (*button*) Opens a DW for definition of the elements by assigning stimuli from the stimulus pool to the elements' key numbers which are listed in the *element pool*.

Part I: construct elicitation (*panel*)

edit triads (*button*) Opens a DW for generation of triads. The latter can either be performed manually by assigning three elements respectively to each triad's key number or by automatically generating a complete variation. In the latter case it is possible to permute 15 elements at most yielding a list of 455 triads.

sequence setup (*sub-panel*)

all triads in random order (*radio button*) If selected, the complete list of triads generated before will be presented to each subject in a random order.

manual selection (*radio button*) If selected, only a selection of triads, i.e. a subset of the triad list, will be presented to each subject with sequences either being randomly generated or predefined (see button *edit* below).

edit (*button*) Opens a DW labeled as *RGT – triad sequence setup (...)* and described below, for selection of subsets of triads and determination of their orders of presentation.

maximum number of constructs per triad (*sub-panel*)

<no label> (*drop-down list*) Here you can choose the maximum number of constructs that may result from one triad comparison. Subjects will then have the opportunity to report on a respective quantity of attributes if they intend to.

initial subject instruction (*sub-panel*)

<no label> (*multi-lined input field*) Space for entering terms of instructions that will be displayed on the subject's *basic screen* (see sec. 6) before the test section (i.e. the first part of the RGT) is started. The input will be treated as a string.

short instruction (on screen each trial) (*sub-panel*)

<no label> (*multi-lined input field*) Space for entering a short term of instruction that will be displayed on top of the subject's user interface in each trial. The input will be treated as a string.

Part II: rating procedure (*panel*)

edit scales (*button*) Opens a DW labeled as *RGT – edit scale settings (...)* and described below, for editing the format settings of the rating scales.

sequence setup (*sub-panel*)

all elements in random order (*radio button*) If selected, all defined elements will be presented to each subject in a random order.

manual selection (*radio button*) If selected, only a selection of elements, i.e. a subset of the element pool, will be presented to each subject with sequences either being randomly generated or predefined (see button *edit* below).

edit (*button*) Opens a DW labeled as *RGT – element sequence setup (...)* and described below, for selection of subsets of elements and determination of their orders of presentation during the rating process.

break between trials (*sub-panel*)

<no label> (*input field*) Space for entering the time period in seconds that lies between the subject's activation of the *weiter* button after finishing rating on the current stimulus (i.e. element) and the initiation of presentation of its successor. The input will be treated as a double value.

initial subject instruction (*sub-panel*)

<no label> (*multi-lined input field*) Space for entering terms of instructions that will be displayed on the subject's *basic screen* (see sec. 6) before the second part of the RGT (i.e. the rating of the elements) is started. The input will be treated as a string.

short instruction (on screen each trial) (*sub-panel*)

<no label> (*multi-lined input field*) Space for entering a short term of instruction that will be displayed on top of the subject's user interface in each trial. The input will be treated as a string.

The following description of a DW refers to the manual selection and determination of sequences of both triads and elements.

DW *"RGT – triad/element sequence setup (...)"*

contents of sessions (*panel*)

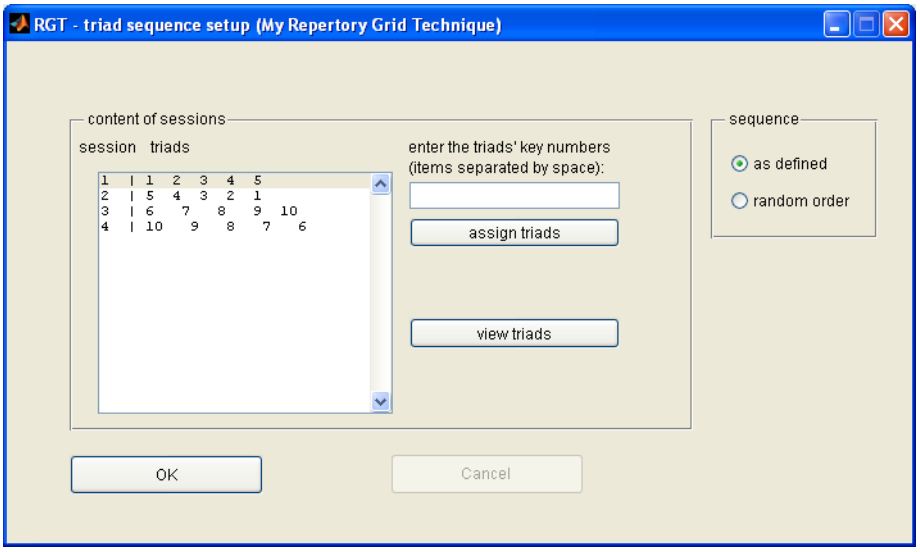

Figure 7.11: DW *RGT – triad/element sequence setup (...)*

<no label> (*multi-columned list box*) Shows all sessions that have been defined earlier (see sec. 5.3) and their belonging subsets of selected triads/elements.

separate items by space (*input field*) Space for entering the key numbers of the triads/elements one wants to be presented during the regarding session. Items have to be separated by space.

assign triads/elements (*button*) Assigns the content of the previously described input field to the session currently selected. Preexisting content will be overwritten.

view triads/elements (*button*) Shows the list of triads/elements defined earlier (see button *edit triads/element pool* above).

sequence (*panel*)

as defined (*radio button*) If selected, the sequence of presentation will be the same as the order of the items entered to the field previously described.

random order (*radio button*) If selected, the sequence of presentation will be randomly generated before each run.

DW *"RGT – edit scale format (...)"*

range settings (*panel*)

number of categories (*drop-down list*) Here you can choose a number of categories that is inbetween 2 and 11. By default it's set to 5.

minimum value (*input field*) Space for entering the minimum value of the scale's range. By default it's set to -2.

category width (*input field*) Space for entering the category width. By default it's set to 1.

apply (*button*) Calculates the categories' numerical representatives.

your category values are (*output field*) Displays the categories' numerical representatives if calculated before by using the *apply* button.

category labels (if required) (*input field*) Space for entering the category labels if required. The single inputs will be treated as strings that have to be separated

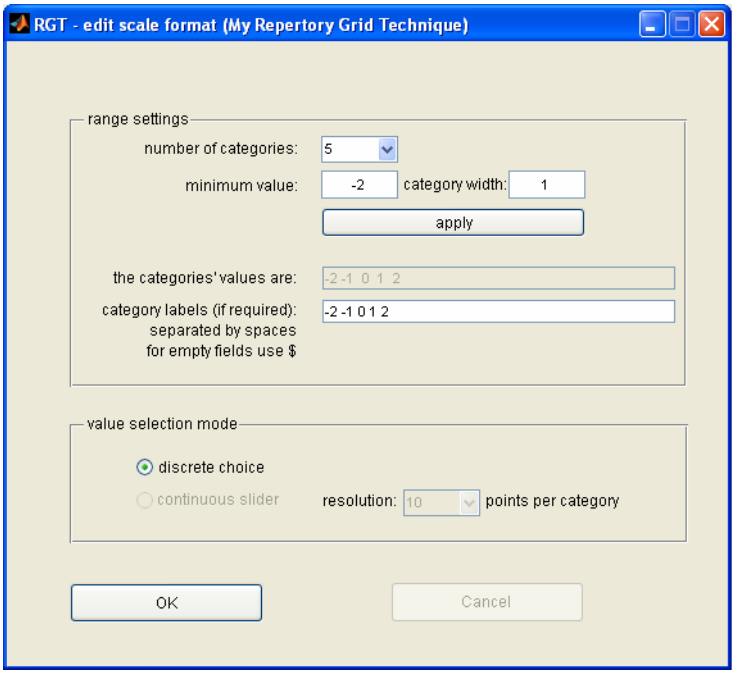

Figure 7.12: DW *RGT – edit scale format (...)*

by space and are assigned to the categories in ascending order of their numerical representatives. If you want to skip a category enter a $-sign.

value selection mode (*panel*)

discrete choice (*radio button*) If selected, the subject will only be able to choose a point on the scale that belongs to one of the categories' marks. Selection is made by using radio buttons.

continuous slider (*radio button*) If selected, the subject will be able to select one out of a specific number of points placed equidistantly spaced within a category interval. Selection is made by using a slider. <not available at the program's present state>

resolution (*drop-down list*) Here the number of points available for selection within each category when using a slider can be chosen from a set of pre-defined possible values (1, 2, 10 or 100). <not available at the program's present state>

7.3 Semantic differential (SD)

7.3.1 The test run

The procedure-specific user interface is depicted in figure 7.13. The stimulus' (i.e. the *object's*) playback is automatically commenced in the beginning of each trial. To allow for a short delay the experimenter can determine a suitable time period (see sec. 7.3.2). The subject's ratings can be recorded from the beginning of a trial (even during the playback) either on discrete scales which are realized by radio buttons or on quasi-continuous scales using sliders with a predefined resolution[18]. Progression to the following trial is possible if the whole set of at most 30 objects has been rated. The latter restriction is due to the finite display resolution

[18]See annotation 1 on page 9

as a greater number would lead to problems with viewing all scales and their belonging poles' labels within one common window.

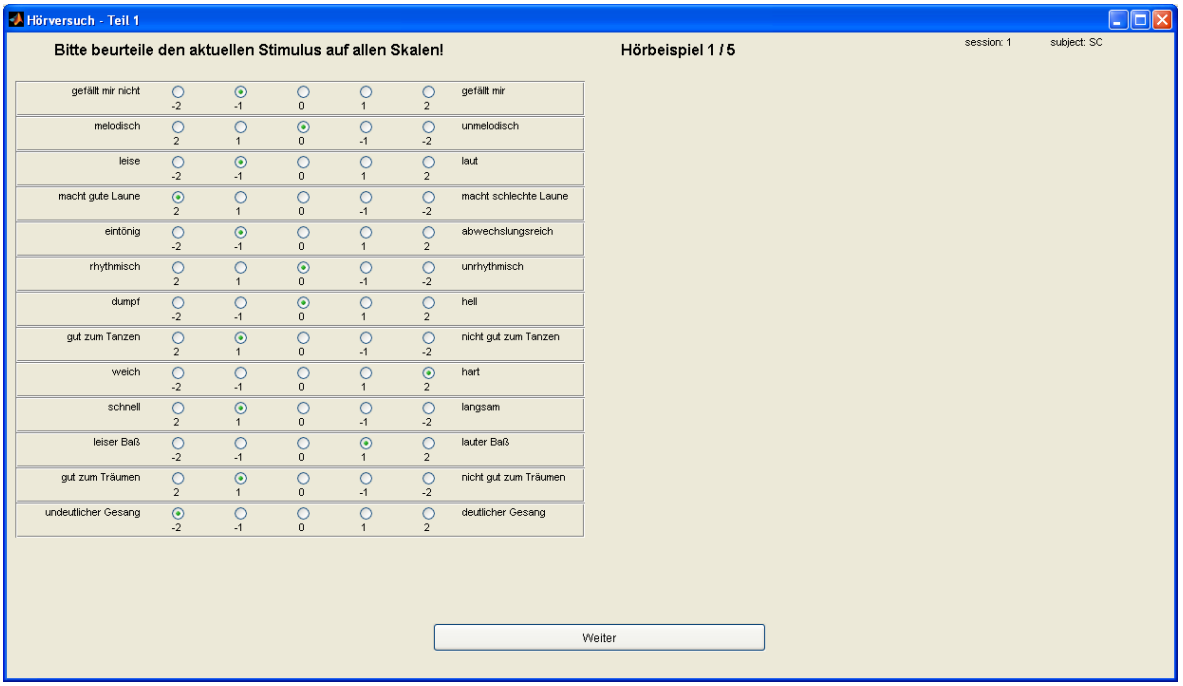

Figure 7.13: The subject's GUI for the performance of the SD.

7.3.2 Setting up properties

Determination of the procedure's settings has to be performed on the DW *SD – main settings (...)* which is depicted in figure 7.14 and can be accessed as explained in subsection 5.5. Its control elements will be described below. The basis of the whole procedure is formed by the definition of objects which is done on the respective DW *SD – object pool (...)* (see fig. 7.15 on p. 50) accessible by the button *object pool*. Thereby a set of stimuli from the stimulus pool can be selected and assigned to object key numbers, gathered in the *object pool*. By applying the button *edit scales* one gets to the DW *SD – edit scales* (see fig. 7.16 on p. 51) which is described below and allows on the one hand adjusting the scales' format settings such as the number of categories, the categories' numerical representatives, category labels if required, the choice between discrete and continuous scales and the scales' resolution in the latter case. On the other hand the verbal descriptors of the scales' endpoints can be defined. Further settings on the performance of the rating process refer to the order of the objects' presentation and can be adjusted in the panel *sequence setup*. There one can choose from either the generation of a random sequence or defining those manually. The latter action can be performed by applying the button *edit* that leads to the DW *SD – object sequence setup (...)* (see fig. 7.17 on p. 52) which is described below. Moreover the time period between the subject's actuation of the *weiter* button and the presentation of the next object can be fixed. Finally two terms of instructions can be entered. One will be displayed in the procedure's beginning and one should only consist of a short phrase which will be shown on the top of the subject's user interface in each trial.

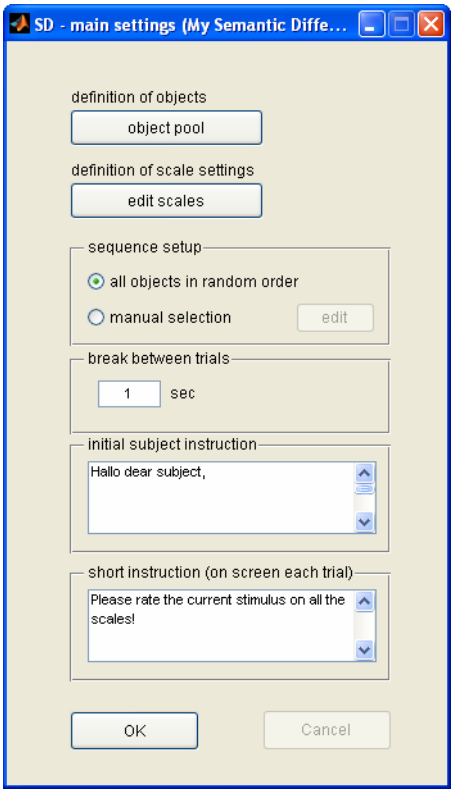

Figure 7.14: DW *SD – main settings (...)*

DW *"SD – main settings (...)"*

object pool (*button*) Opens a DW for definition of the objects by assigning stimuli from the stimulus pool to the objects' key numbers which are listed in the *object pool*.

edit scales (*button*) Opens a DW labeled as *SD – edit scales (...)* and described below, for editing the formal settings of the rating scales and defining the endpoints' verbal descriptors.

sequence setup (*panel*)

- **all objects in random order** (*radio button*) If selected, all objects will be presented to each subject in a random order.
- **manual selection** (*radio button*) If selected, only a selection of objects, i.e. a subset of the object pool, will be presented to each subject with sequences either being randomly generated or predefined.
- **edit** (*button*) Opens a DW labeled as *SD – object sequence setup (...)* and described below, for selection of subsets of objects and determination of their orders of presentation.

break between trials (*panel*)

- **<no label>** (*input field*) Space for entering the time period in [sec] that lies between the subject's actuation of the *weiter* button after finishing rating on the current stimulus (i.e. object) and the initiation of its successor's presentation. The input will be treated as a double value.

initial subject instruction (*panel*)

<no label> (*multi-lined input field*) Space for entering terms of instructions that will be displayed on the subject's *basic screen* (see sec. 6) before the test section is started.

short instruction (on screen each trial) (*panel*)

<no label> (*multi-lined input field*) Space for entering a short term of instruction that will be displayed on top of the subject's user interface in each trial.

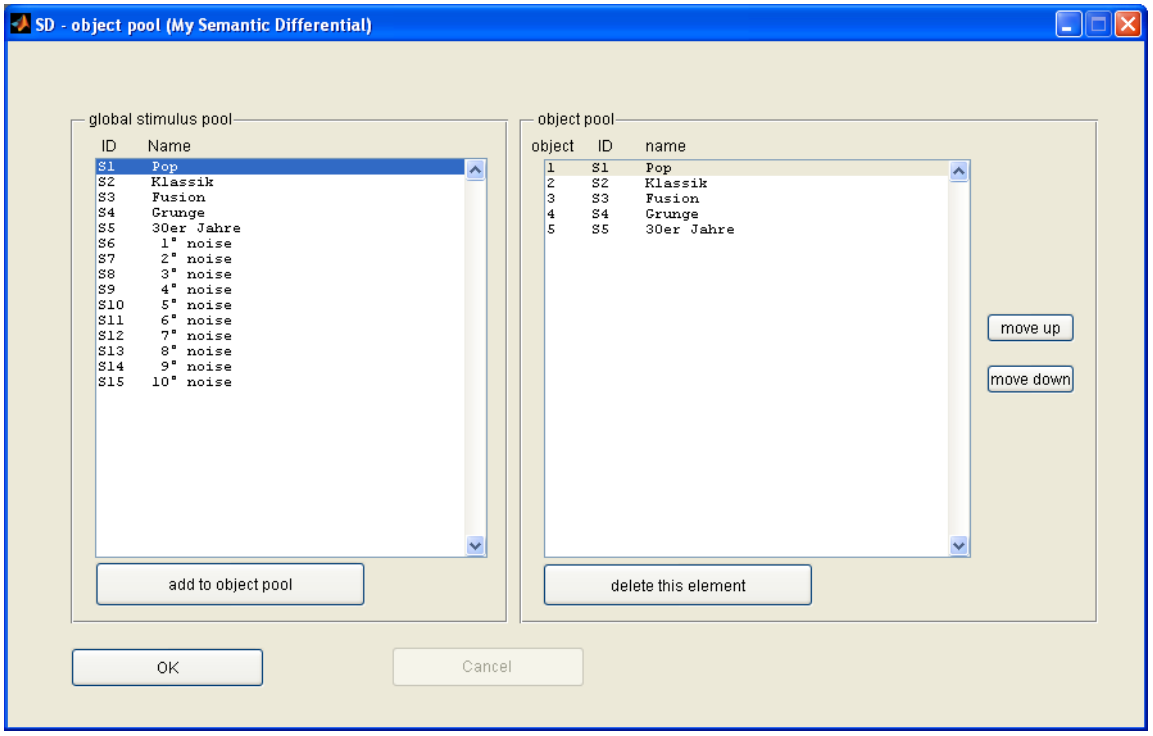

Figure 7.15: DW *SD – object pool (...)* (see text on p. 48)

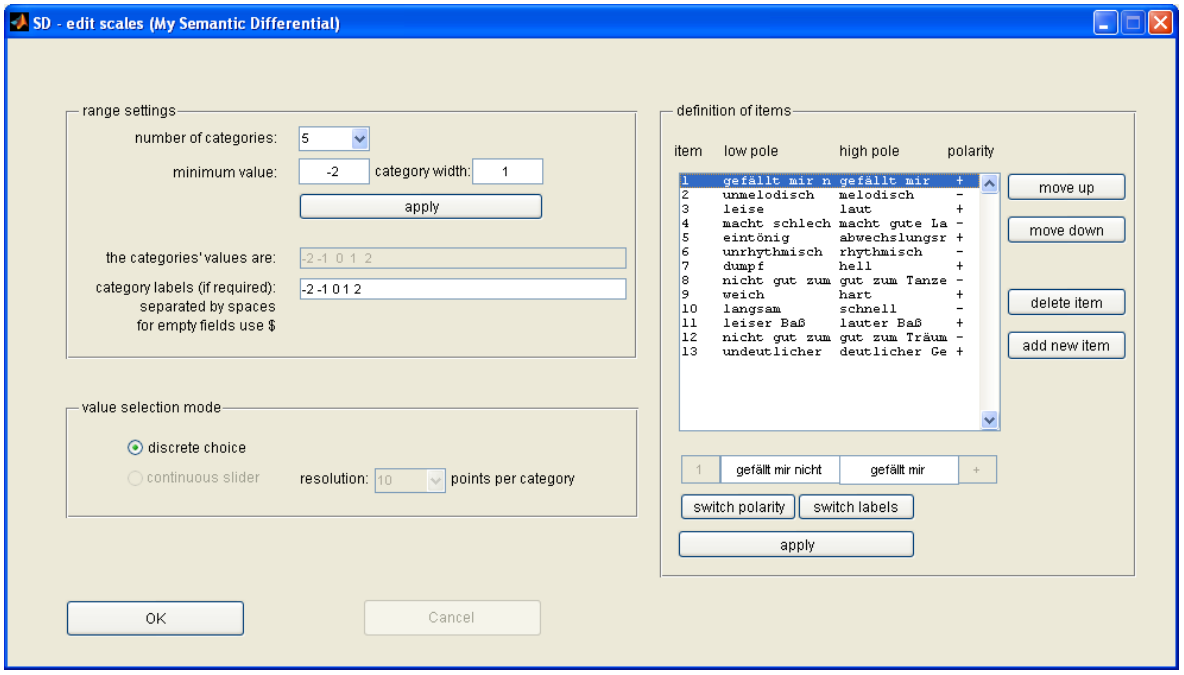

Figure 7.16: DW *SD – edit scales (...)*

DW *"SD – edit scales (...)"*

range settings (*panel*)

number of categories (*drop-down list*) Here you can choose a number of categories that is in between 2 and 11. By default it's set to 5.

minimum value (*input field*) Space for entering the minimum value of the scale's range. By default it's is set to -2.

category width (*input field*) Space for entering the category width. By default it's set to 1.

apply (*button*) Calculates the categories' numerical representatives.

the categories' values are (*output field*) Displays the categories' numerical representatives if calculated before by using the *apply* button.

category labels (if required) (*input field*) Space for entering the category labels if required. The single inputs will be treated as strings that have to be separated by space and are assigned to the categories in ascending order of their numerical representatives. If you want to skip a category enter a $-sign.

value selection mode (*panel*)

discrete choice (*radio button*) If selected, the subject will only be able to choose a point on the scale that belongs to one of the categories' marks. Selection is made by using radio buttons.

continuous slider (*radio button*) If selected, the subject will be able to select one out of a specific number of points placed equidistantly spaced within a category interval. Selection is made by using a slider. <not available at the program's present state>

resolution (*drop-down list*) Here the number of points available for selection within each category when using a slider can be chosen from a set of pre-defined possible values (1, 2, 10 or 100). <not available at the program's present state>

item definition (*panel*)

<no label> (*multi-columned list box*) This list shows the scales' endpoint descriptors representing the *items* under test, i.e. the attributes underlying the affective or perceptual measurement. During the test run the items and their belonging scales will be placed on the screen in the same order as they are created in the list. The further columns view the poles' verbal descriptors and the scale's polarity, i.e. its orientation when being posed during the test run. For the latter a "+" indicates that the low pole (i.e. the pole belonging to a minimum category value) will be posed on the left side of the DW and the high pole (i.e. the pole belonging to a maximum category value) on the right side. The reversed case is indicated by a "-". The list may not contain more than 30 items. This restriction is due to the finite display resolution as a greater number would lead to problems with viewing all scales and their labels within one common window.

<no label> (*output field* 1st col) Displays the selected item's number which also represents its position within the order of presentation during the test run.

<no label> (*input/output fields* 2nd & 3rd col) Spaces for displaying and editing the verbal descriptors of the scale's poles belonging to the selected item.

<no label> (*output field* 4th col) Indicates the polarity of the scale belonging to the selected item.

switch polarity (*button*) Switches polarity of the scale belonging to the selected item.

switch labels (*button*) Switches verbal descriptors of the scale's poles belonging to the selected item.

apply (*button*) Applies changes on the contents of the input/output fields described above to the selected item.

move up/move down (*button*) Shifts the selected item to a proximate list position above/below.

delete item (*button*) Deletes a selected item.

add new item (*button*) Adds a new item to the list after its last position.

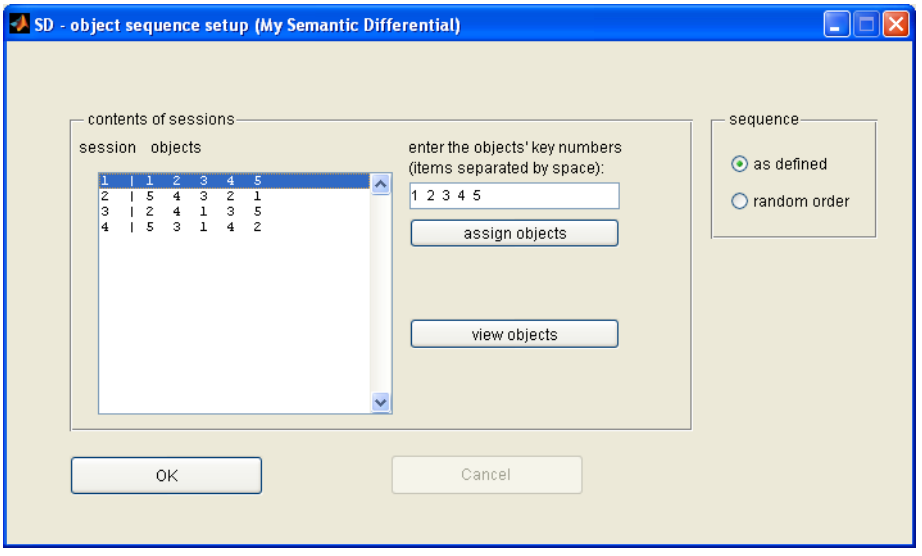

Figure 7.17: DW *SD – object sequence setup (...)*

DW ***"SD – object sequence setup (...)"***

contents of sessions (*panel*)

<no label> (*multi-columned list box*) Shows all sessions that have been defined earlier (cp. sec. 5.3) and their belonging subsets of selected objects.

enter the objects' key numbers (...) (*input field*) Space for entering the key numbers of the objects one wants to be presented during the regarding session. Items have to be separated by space.

assign objects (*button*) Assigns the content of the previously described input field to the session currently selected. Preexisting content will be overwritten.

view objects (*button*) Shows the list of objects defined earlier.

sequence (*panel*)

as defined (*radio button*) If selected, the sequence of presentation will be the same as the order of the items entered to the field previously described.

random order (*radio button*) If selected, the sequence of presentation will be randomly generated before each run.

7.4 ABX detection test (ABX)

7.4.1 The test run

The subject's GUI while performing an ABX test is divided into two parts, the upper one, where the subject can start/stop the three stimuli and the lower section for answering the question, which of the stimuli A or B is equal to X. The stimuli are always re-triggered, so while hitting the upper Button A or B or X, the stimuli always starts playing from the beginning. In the lower section of the test subjects's GUI (under the short question, definable in the ABX Main Settings) you will find two answer buttons A and B. After one of them is hit, the "Weiter"-Button will be enable, allowing the subject to go on to the next trial round. Only the last decision made by the subject counts, so it is possible, to change the given answer after an first decision has been made.

7.4.2 Setting up properties

To perform the ABX test, the investigator has to define:

1. Number of trials n
2. The Stimuli assignments for A and B
3. Text providing the necessary information for subjects

All other fields in the ABX main settings have only informational value and are describe below the picture. For further information about the parameters see Burstein, 1988, Clark, 1991,Leventhal, 1986.

DW ***"ABX – edit main settings (...)"***

initial subject instruction (*input field*)

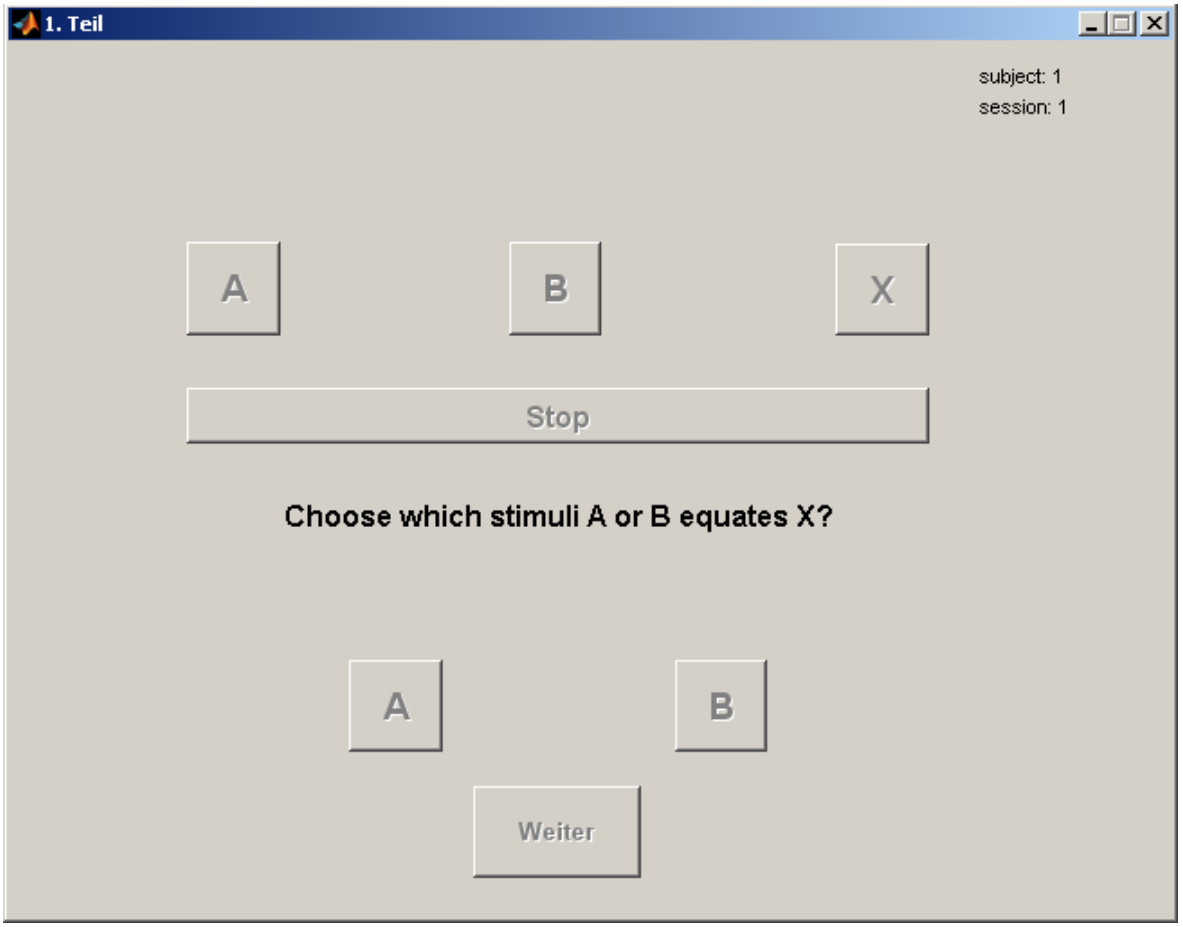

Figure 7.18: The subject's GUI through one A/B/X Trial.

The text shown to the subject before the ABX test starts

short instruction (*input field*)

The text shown in every trial on the subject's GUI

number of trials n (*input field*)

Number of repetitions, Value that defines the length of the ABX test

number of correct answers c (*input field*)

Setting this value and fill out the "number of trials"-field and press the "Calculate"-Button will result in the corresponding "Significance Level"

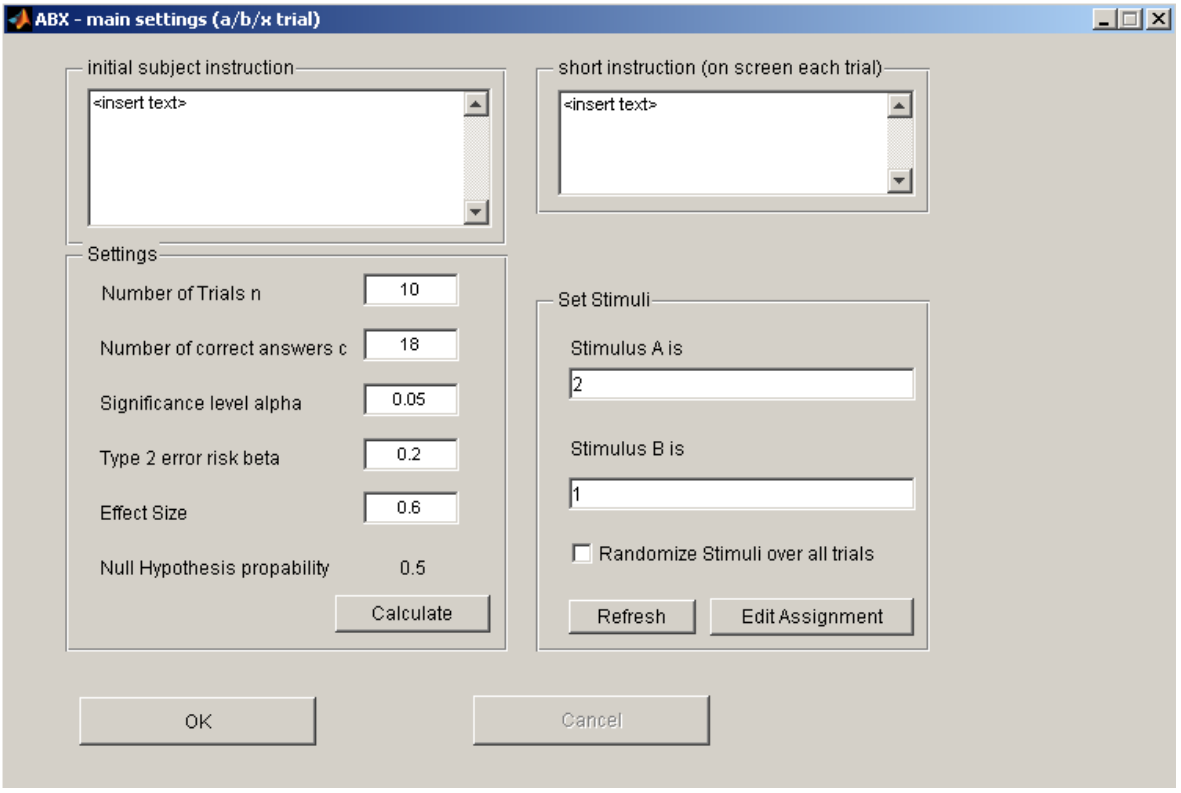

Figure 7.19: ABX Trial Main Settings

Significance level alpha (*input field*)

setting this value as well as the Type II error risk and the "Effect size" leaving empty the "number of trials"-field and the "number of correct answers"-field will result in the corresponding "number of trials" and "number of correct answers".

type 2 error risk beta (*input field*)

see "Significance level"

Effect size (*input field*)

see "Significance level"

Set Stimuli (*input field*)

by pressing "Edit Assignment" you enter the menu shown below, where you can define, which stimuli (you define them first in the main menu) you want to compare in this ABX test; by pressing the "Assign"-Button you say, which stimuli is rather Stimuli A in the trial and which one is B; by hitting the "refresh"-button in the main settings menu, the assigned stimuli will also show up in this menu

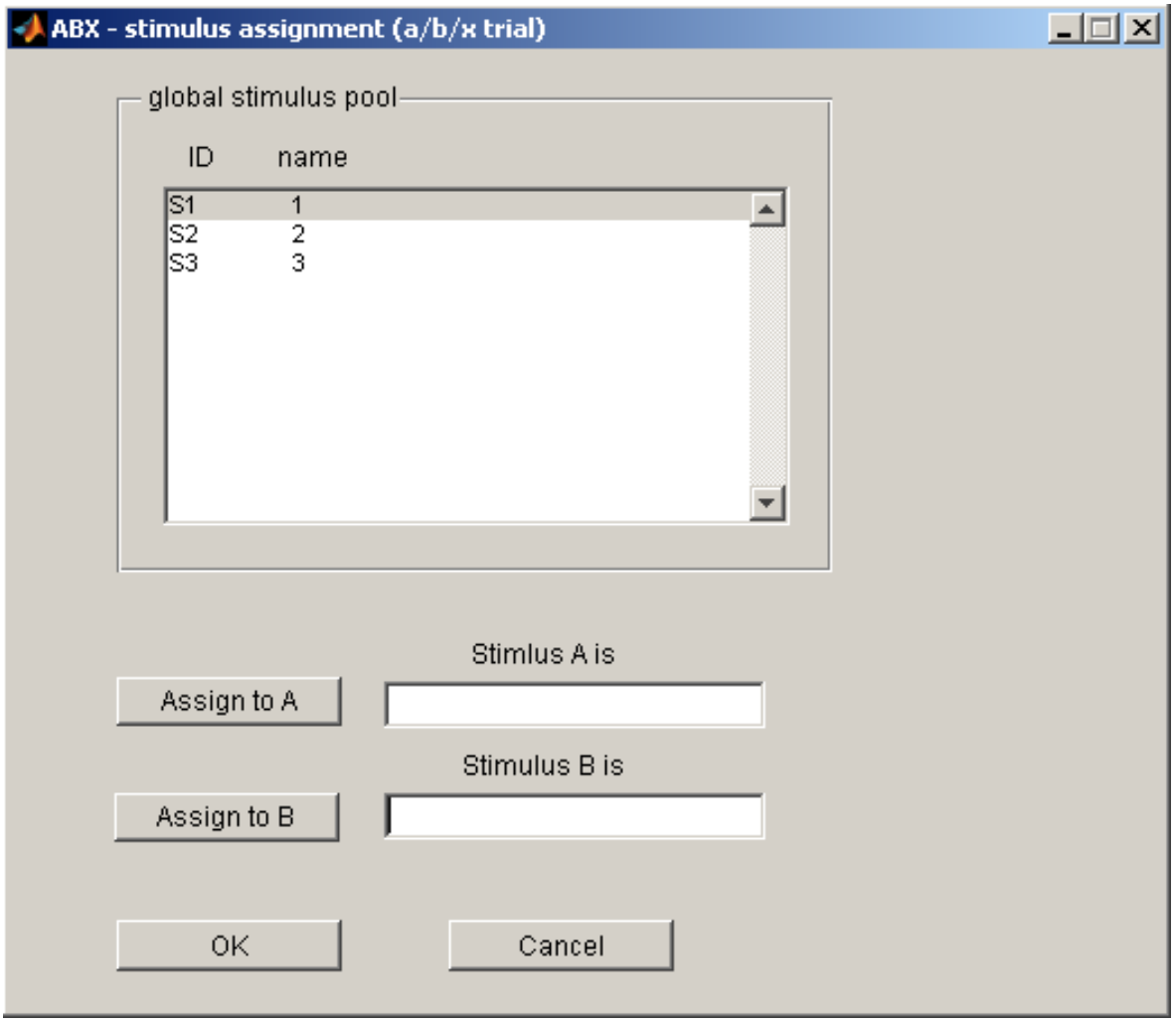

Figure 7.20: Assigning the stimuli to be judged in the current trial.

7.5 Spatial Audio Quality Inventory (SAQI)

The Spatial Audio Quality Inventory (SAQI) is intended for a qualitatively differentiated, comparative auditory assessment of real, imagined and simulated acoustic scenes in order to to reveal specific shortcomings of a simulation under test and allow for a directed technical improvement. The SAQI comprises 48 verbal descriptors of perceptual qualities assumed to be of practical relevance when comparing virtual environments (VAEs) to real or imagined references or amongst each other. It was generated by a Focus Group of 20 German experts for virtual acoustics. Five additional experts helped verifying the unambiguity of all descriptors and the related explanations. Moreover, an English translation was generated and verified by seven bilingual experts. Rationale and methodology pursued in constructing the English (SAQI-EN) and the German (SAQI-GER) vocabulary are described in more detail in Lindau et al. [2014a] and Lindau et al. [2014b].

The SAQI vocabulary in its entirety (including perceptual descriptors, circumscriptions, scale end label, and - if given - illustrative sound examples) is intended to enable experts in the field to train any laymen to use it for assessments of VAEs. An extensive Test Manual is freely available for download (Lindau [2014]). It is strongly recommended that you read it, in order to better understand the following descriptions of the SAQI test in WhisPER. Further, there is a SAQI project website at http://www.ak.tu-berlin.de/saqi.

7.5.1 The test run

The SAQI is a rather extensive test instrument, which can be widely customized to the user's specific needs. If one chooses to conduct a complete SAQI test, rating a singular perceptual difference quality will involve presenting five concurrent GUIs (Graphical User Interfaces) to the listener:

1. quality name and description
2. rating scale
3. assignment of modifications (2 GUIs)
4. choice of assessment entities

Whether this comprehensive way of testing will be actually required will of course depend on the researcher's aims. Whereas only the second GUI will be mandatory (quality name and rating scale) the other four may be optionally used in a test. As an example, in case of more exploratory research questions one might decide to collect as much information as possible, whereas for a more confirmatory study, the SAQI test may be reduced to cover only specific aspects. Hence, the user may conveniently chose from the test settings which assessments shall be conducted (i.e. which GUIs will be shown to the user) in a test run. In the following section it will be shown how a (complete) SAQI evolves from the view of the test subject. In section 7.5.2 it will be explained, how a SAQI test may be parameterized by the experimenter.

Description GUI *(Fig. 7.21)*
This GUI is intended for instructing test subjects to focus on a specific auditory quality to be assessed. For each quality it contains the quality's name and a closer circumscription. **Note:** As described in the SAQI Test Manual (Lindau [2014]) all subjects should be suitably trained to the full understanding of the SAQI **beforehand**. Therefore, presenting the qualities circumscriptions here is only thought to work as a short reminder.

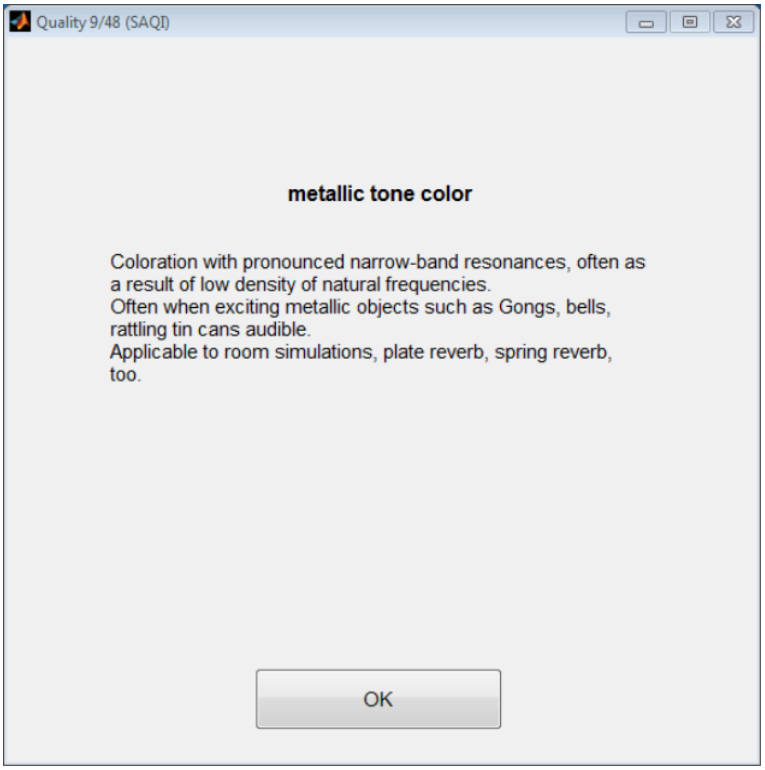

Figure 7.21: Example presenting a name and circumscription of a auditory quality

Rating GUI *(Fig. 7.22-7.25)*
This GUI is intended for rating the perceived amount of the (described) auditory difference when comparing a test stimulus to to a given or imagined reference. For each quality it presents its name, the rating scale and play buttons for playing back the reference and the comparison stimuli. The 'OK'-button saves the rating and guides the subject to the next GUI.
The design of this GUI will change slightly according to the type of rating scale that belongs to a quality. Most of the qualities in SAQI were found to demand bipolar rating scales represented by sliders, however, two qualities, the 'overall difference' and 'sequence of events', are understood as unipolar. To support a better comparability of ratings, unipolar sliders are displayed with half the length of the bipolar ones.
The two qualities 'horizontal direction' and 'vertical direction' are rated using an edit field allowing subjects to directly type in the perceived difference in degree instead of a slider. In addition, there are two radio buttons for indicating the perceived difference in direction (clockwise, or counter clockwise).
Finally, one quality has been found to be dichotomous (difference in 'front-back position'). This descriptor will be rated by using radio buttons: 'not confused' and 'confused'.

Note, that for rating the first quality ('overall difference') the user has to listen at least once to the stimuli (i.e. use the play button), otherwise the 'OK'-button won't be enabled.

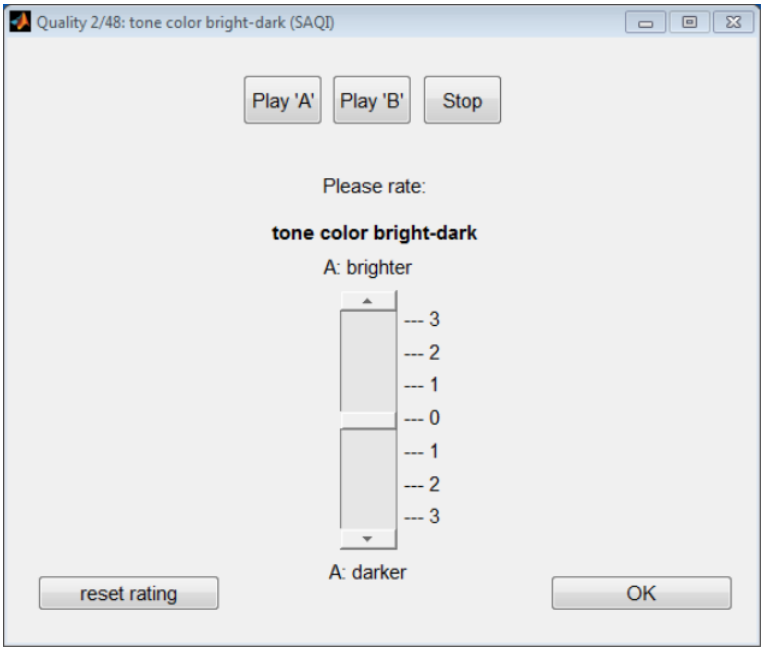

Figure 7.22: Example of a quality to be rated using a bipolar, open-ended rating scale

Figure 7.23: Example of a quality to be rated using a unipolar, closed/open-ended rating scale

Figure 7.24: Example of a quality to be rated using direct bipolar, closed-ended rating

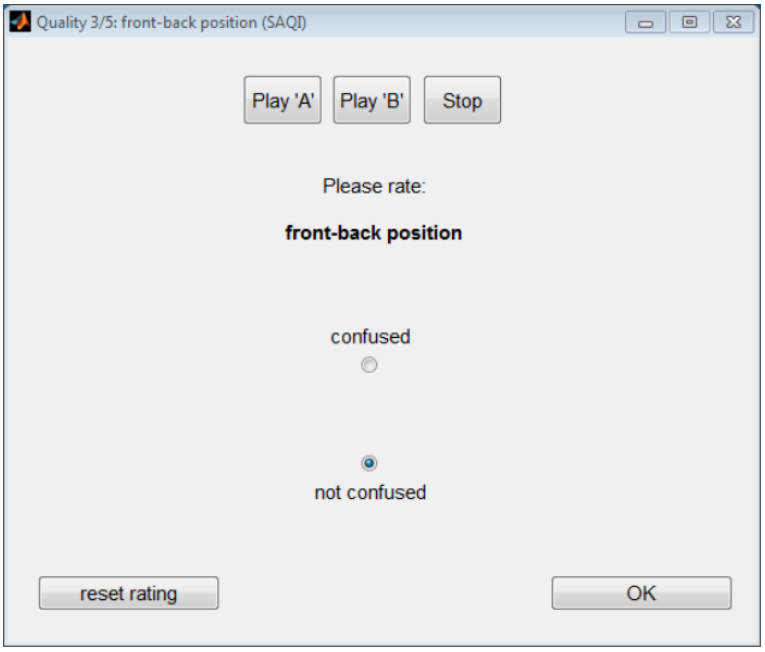

Figure 7.25: Example of a quality to be rated using a dichotomous scale

Optional GUIs *(Fig. 7.26-7.28)*
Since both, a closer description of the perceived qualities by indicating certain modifications, and the assignment of the perceived differences to certain assessment entities is optional, these GUIs will only appear if they were selected before in the test set up.
The GUIs contain questions referring to the modifications and assessment entities for each single descriptor and will be displayed right after the rating of an perceptual difference in the main GUI. More information about the meaning of modifications and assessment entities can be found in the SAQI Test Manual (Lindau [2014]); related information on the SAQI test setup can be found in sec. **??**.
Please note, that even if the modifications and/or entities are enabled for the test run, they are not going to be displayed if a quality was not rated or was rated with 0 (no difference).

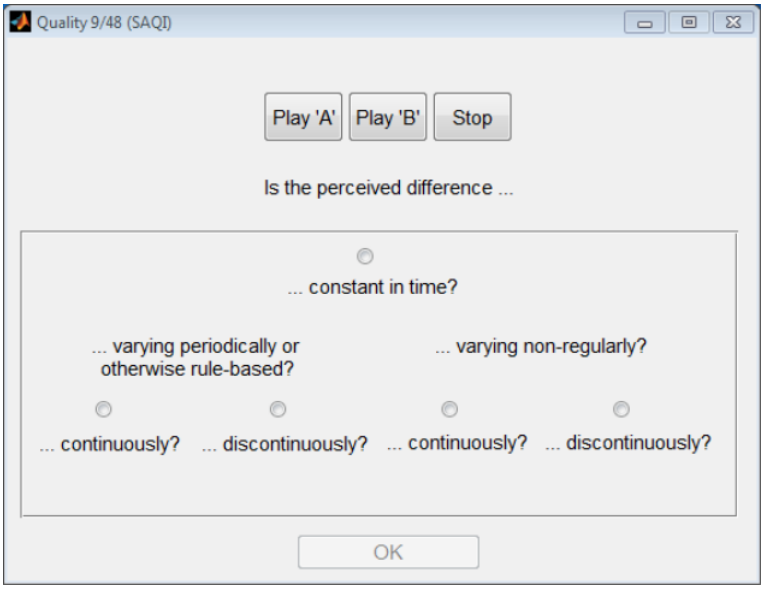

Figure 7.26: GUI for the rating the temporal variability of perceived differences

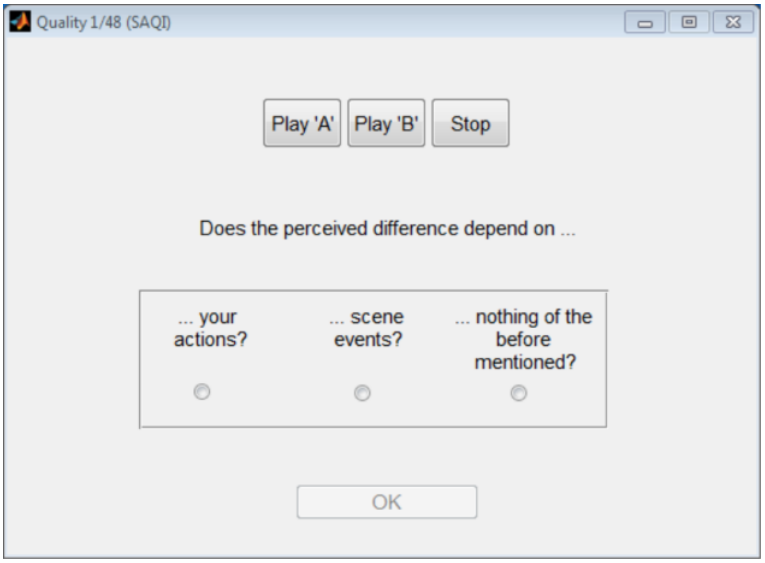

Figure 7.27: GUI for the rating the cause of perceived differences

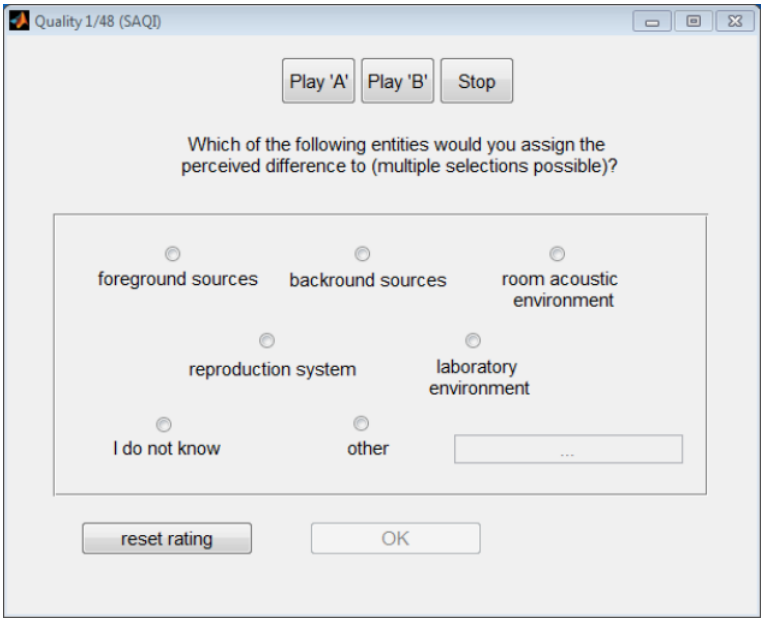

Figure 7.28: GUI for assigning assessment entities to perceived differences

7.5.2 Setting up properties (Fig. 7.29)

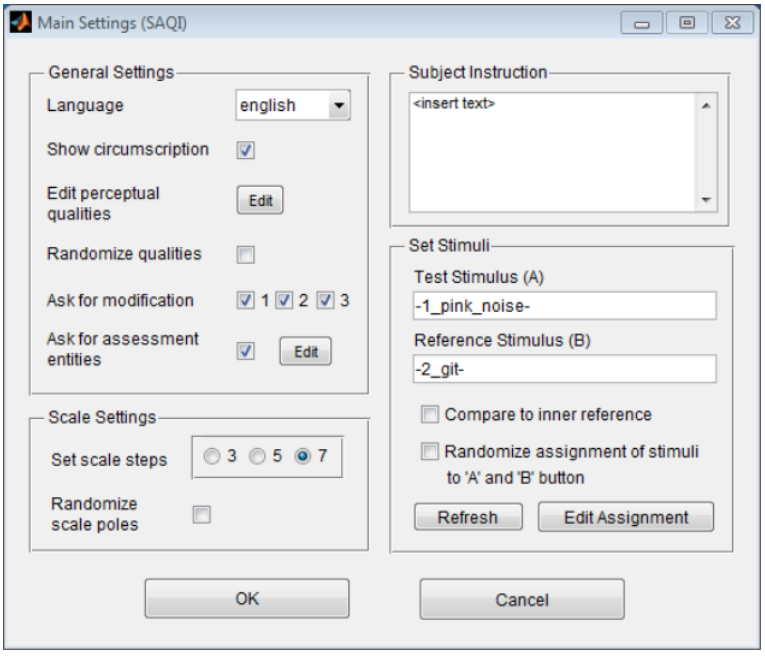

Figure 7.29: SAQI Main Settings

DW *"SAQI – edit General Settings (panel)"*

Language (*drop-down list*)
It is possible two choose between a German and an English test run by using the respective drop-down list. The default language is German. In programming the SAQI, possibilities

have been foreseen for adding new languages. In case you plan to add a new national version of SAQI, please check section 7.5.3.

Show circumscription (*check box*)
Check to show the written circumscriptions of the perceptual qualities in a separate window before assessing each quality. The default setting is 'on'.

Edit perceptual qualities (*push button*; Fig. 7.30)
Pushing the 'Edit' button opens another window for selecting the perceptual qualities. It contains all 48 original descriptors included in the SAQI. By default they are entirely selected. Furthermore, all qualities are subdivided in categories.
In order to customize a SAQI test, it is possible to either select singular qualities separately (left window pane) or to select entire categories (right window pane).

Please note, that the first ('overall difference') and the last quality ('other') **must** be selected, otherwise a warning window will pop up, telling you to do so. If qualities are chosen via whole 'categories' (right window pane), these two qualities are selected automatically. The 'De/Select all' button enables you to select or de-select all (also including both, the first and last) qualities.

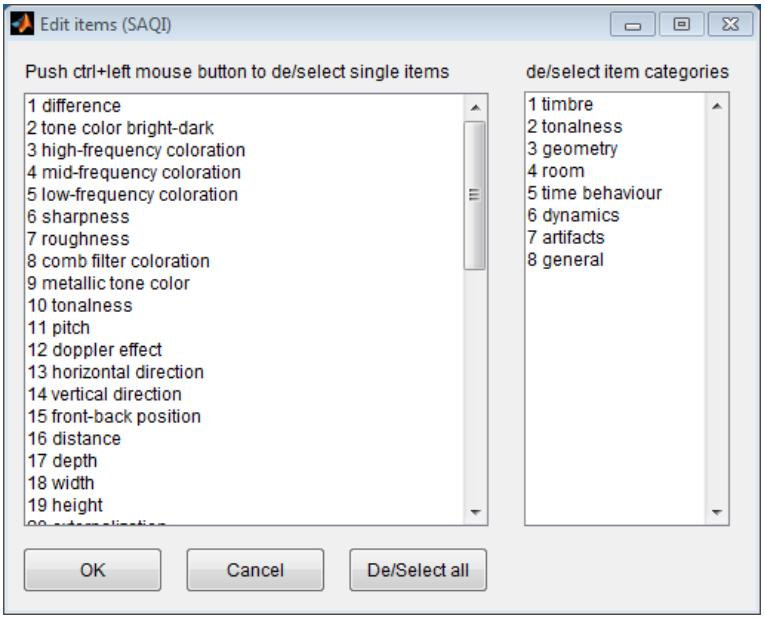

Figure 7.30: GUI for the selection of perceptual qualities. Left: singular selection, right: category-based selection.

Randomize qualities (*check box*)
Randomizes the selected perceptual qualities during test presentation according to the following scheme: Firstly, the order of categories is randomized, and secondly, the order of qualities within the categories. Exceptions are again the first and the last quality, which will be invariably presented at first and last position, respectively.

Ask for modification (*check boxes*)
By default, all three types of modifications a pre-selected (checked) for assessment. Please note, that modification 2 depends on modification 1. Hence, if modification 1 is disabled for assessment, modification 2 will automatically be disabled, too. Modification 2 cannot be

chosen for exclusive assessment.

1. modification 1 (temporal variation): constant/varying periodically or otherwise rule-based/varying non-regularly
2. modification 2 (temporal variation): continuous/discontinuous
3. modification 3 (causality): depending on scene events/depending on user interaction/independent

Ask for assessment entities (*check box / push button*; Fig. 7.31)
By pushing the 'Edit' push button it is possible to define five entities a perceived difference may be further ascribed to. If needed, the default ones can be overwritten. Furthermore, the entities can be selected exclusively by the check boxes to the left. The last two entities 'I don't know' and 'other' cannot be renamed. By default, all entities are enabled.

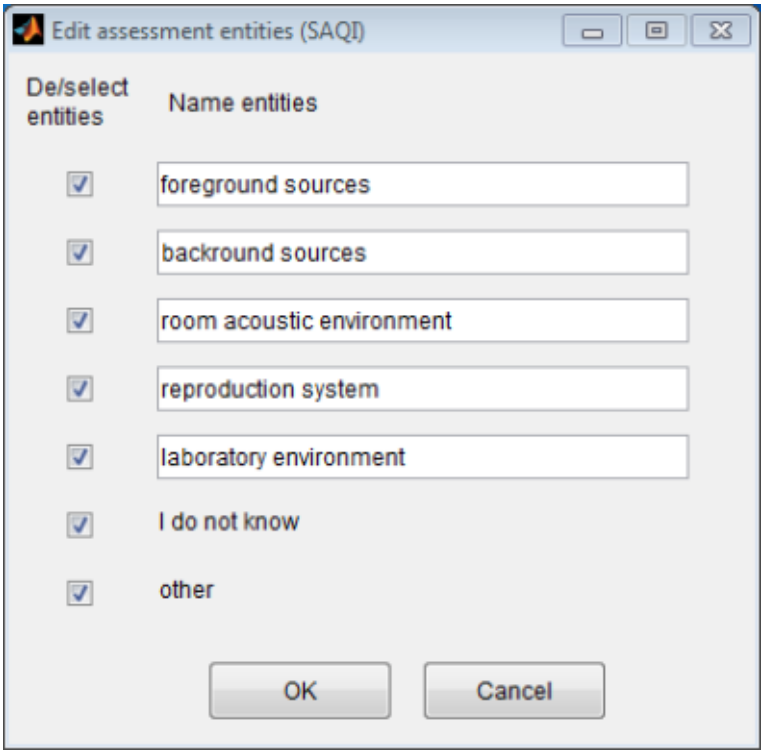

Figure 7.31: Default assessment entities

DW *"SAQI – Scale settings (panel)"*

Set scale steps (*radio button*)
The user can choose between the scale steps '7', '5' and '3'. The default setting is '5'. Scale steps only refer to the bipolar and unipolar qualities.
For example: A bipolar scale with '7' steps will range from +/-3, where 0 marks the middle ('no difference'). At the same time, an unipolar scale will range from 0 to 3, enabling to encode the **amount** of a perceived difference, only, and not its **direction**.

Important note: Please note that scale steps are intended only as visual guidance for test subjects. Internally, all ratings will always be coded in a range of -1 to +1 in floating point resolution independent of the chosen number of scale steps to be displayed.

Randomize scale poles (*check box*)
Check to randomize scale orientation and corresponding label while test presentation. The default setting is 'off'. During default setting scales are arranged in a way that perceived increases are 'logically' encoded by moving a slider upwards. The label, that is listed first in the definition file *saqi_scale_label_language.m* is coded with +1, regardless of randomization.

DW *"SAQI – Subject instruction (edit text field)"*

Space for entering terms of instructions that will be displayed on the subject's *basic screen* (see sec. 6) before the test section is started.

DW *"SAQI – Set stimuli (panel)"*

Compare to inner reference (*check box*)
Check this option to allow the user to rate the test stimulus against his/hers inner reference. In this case, the second play button and stimulus will be omitted in the respective GUIs and GUI instructions will be reformulated accordingly. Refer to the SAQI Test Manual Lindau [2014] for hints towards a proper test subject instruction.

Randomize Assignment of stimuli to 'A' and 'B' button (*check box*)
Check to randomize assignment of the the stimuli 'A' and 'B' to play buttons when assessing the next perceptual quality. In this case, the subject must listen to the stimuli before being able to rate the current quality. This will noticeably increase the test duration. If stimuli are randomized, the rating is decoded by switching the sign, and thus, the effect of randomization is compensated in the rating.

Edit Assignment (*push button*)
By pushing "Edit Assignment" you enter the menu shown below, where you can define which stimuli you want to compare in this SAQI test (you define them first in the main menu). By pressing the "Assign"-Button it will be decided, which of the stimuli will be stimulus A (test stimulus), and which one will be B (reference stimulus). By hitting the "refresh"-button in the main settings menu, the assigned stimuli will also show up in this menu.

7.5.3 How to: Implementation of a new language

Adding a new language is possible, and the required steps are described in the following. However, this reconfiguration will need some understanding of Matlab and is not thought to be a normal usage procedure of WhisPER. Hence, if you feel not save with these requirements, please contact us for help. We can also - as a service - add national versions of SAQI to WhisPER. In any case, please send us the new language files, so we can update the WhisPERrelease on the homepage.

Step 1: Create new language files

As a first step, it will be necessary to re-create some of WhisPER's Matlab scripts in your desired language, containing the names for:

- qualities (original example file: saqi_items_german.m),

- categories (original example file: saqi_categories_german.m),
- circumscriptions (original example file: saqi_definitions_german.m),
- GUI phrases (original example file: saqi_phrases_german.m),
- scale label ends (original example file: saqi_scale_label_german.m),
- assessment entities (original example file: saqi_entities_german.m)

The best way will be to copy the original (German or English) versions of these files at a save place and change them accordingly, one by one. The structure of these files is pretty straight-forward and may be checked out by retrieving them from the WhisPER source code folder. Please note, that your new files need to be known in the Matlab's path environment (at best they are saved in the WhisPER source code folder).

In case letters with accents occur in the implemented language please decode them as done in *saqi_scale_label_german.m* to ensure compatibility across different system and system locales.

Step 2: Adding a new language case to the setup GUI

Now, you should update the language selection switch of the setup GUI by adding another case for your language. You can find the program code in the *saqi_edit_main file*. The switch is arranged in a function called 'pop_language_Callback' (line 365 et seq.).

The easiest way of adding, is to copy-and-paste one of the existing cases and to rename the applied m-files (e.g. saqi_items_french) in the new one (e.g. case 3).

Step 3: Additions on the GUI

For the last step you will need to edit the figure 'saqi_edit_main.fig', for example using the GUIDE-Tool of Matlab (right-click on the fig-file and choose 'open with GUIDE').
Next, double click on the language drop-down list in the panel 'General Settings' and a Matlab 'Inspector' will open. In this 'Inspector' it is possible to change settings. Open the edit-box by double clicking on 'String' an add your language under the last enlisted.

8 Test procedures - default values' presets

This section is about a feature of WhisPER that allows the user to manipulate all the procedures' default values (as they are mentioned in chap. 7), which the procedures are initialized with upon adding them to a test series via the test section setup (cf. sec. 5.4).

The DW that allows you to do this is accessible via the main window's menu *edit\procedure's defaults*.

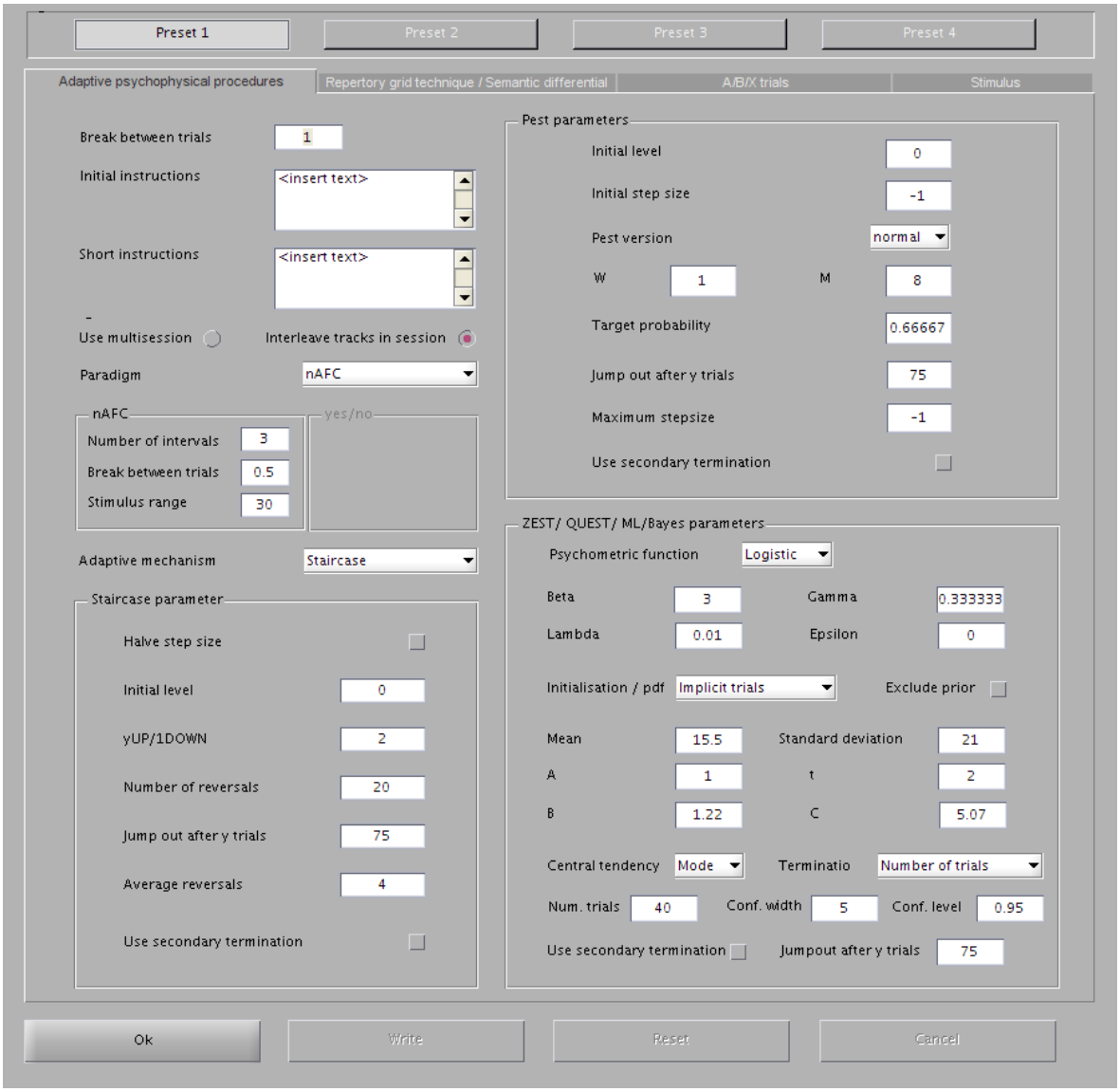

Figure 8.1: DW *Configure defaults' presets*

This DW consists of a middle part comprising four *tabs* containing all the configurable default values for all the procedures (plus for the stimuli), a lower part consisting of four *buttons* for

closing, writing, resetting and canceling your configurations and an upper part consisting of four *preset buttons*, each selecting one of four presets. Following that order the description in detail:

Adaptive psychophysical procedures (*tab*) This *tab* contains spaces for all the configurable defaults of the parameters of the AP procedures as explained in sec. 7.1.2.
Please note: As stimulus definitions take place separately from procedure setup (cf. sec. 5.2 and 5.4 of chap. 5) all AP configuration options involving (already defined) stimuli in any way (e. g. stimulus assignments to intensities or reference level) sensibly simply cannot be available here. They have to be set later.
Please also note that as this DW is intended for the power user all the input fields are presented in a condensed and often abbreviated way, thus lacking the guidance of the DWs of the *in procedure setup* as in sec. 7.1.2, but instead allowing all the options to be set in one window only in the flattest hierarchy and the fastest way possible. And please also note that changing values here will of course not keep you from making changes later, after adding a procedure to your test series – changing values here will only affect the values your procedure *already has* just after you have added it. Thus it is perfectly possible to just once set those values here that are most likely to *not* change from listening experiment to listening experiment (like for example the psychometric function or the a priori p. d. f. or the number of trials etc.) and thus are always initialized to *your* defaults, and later fill in those values that are specific to a single listening experiments (like e. g. the participants' instructions).
Of course you can also use this DW to just make all your wanted changes for one specific listening experiment in case you simply prefer the availability of all the configuration options in one place without having to click through multiple configuration DWs.

Repertory grid technique / Semantic differential (*tab*) This *tab* contains spaces for all the configurable defaults of the parameters of the RGT procedure as well as the SD procedure as explained in sec. 7.2.2 and in sec. 7.3.2 respectively. For notes and hints please see above.

A / B / X trials (*tab*) This *tab* contains spaces for all the configurable defaults of the parameters of the ABX procedure as explained in sec. 7.4.2. For notes and hints please see above.

Stimulus (*tab*) This *tab* contains spaces for all the configurable defaults of stimuli (cf. sec. 5.2), see fig. 8.2. Notes and hints above also apply here, further explanation see below.

- **New Stimulus default name** (*input field*) Space for entering what is to be the default name of a stimulus each time one is created either by pushing the *new button* or the *multi new button* in the stimulus pool DW (as explained in sec. 5.2). As changing this from " - no name - " to whatever you desire will effectively *inhibit* automatic name generation for stimuli based on a .wav file name, this will only be useful under two conditions: First, when you just do not want your stimuli to be named corresponding to the .wav files loaded, and second, when there are no .wav files to be loaded to begin with, i. e. your stimuli are OSC only. Please be advised, that in both cases the name entered here will only be a "base name", as it will result in the *same* default name for every stimulus generated. With that "base name" already filled in the stimulus' name field, however, individualizing each name by simply manually appending a number, for example, is facilitated a lot. Cf. below for OSC usage scenarios.
- **New Stimulus default OSC data** (*panel*) Contains spaces for entering what is to be the default OSC parameters of newly generated stimuli. As explained above for the *New Stimulus default name* the values entered here will result in exactly the *same*

default OSC data values for every stimulus newly generated. Thus their use is to serve as "base parameters", with later only individual addition of e. g. numbers in the *OSC data field* for each stimulus facilitating the creation and configuration of a bigger set of OSC only stimuli.

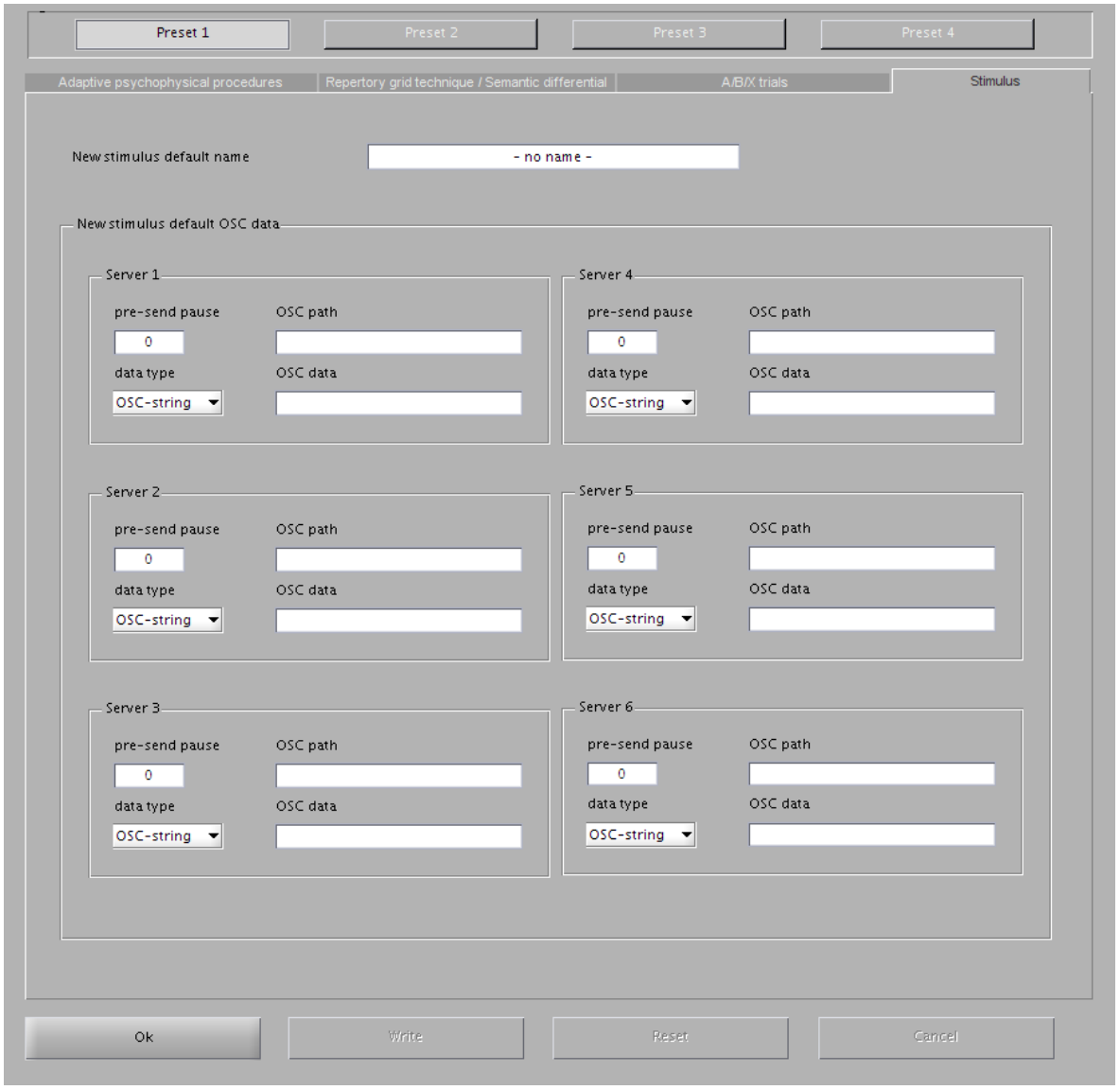

Figure 8.2: DW *Configure defaults' presets* – tab "Stimulus"

Ok (*button*) This *button* will save your changes and close the DW. Please note that "saving your changes" means making them immediately effective (for the addition of test procedures for example) as well as storing them so they stay the way you configured them even over the course of exiting the program (WhisPER as well as MATLAB) and starting it again.

Write (*button*) This *button* will save your changes and leave the DW open. Again "saving your changes" means making them immediately effective (for the addition of test procedures for example) as well as storing them so they stay the way you configured them even over the course of exiting the program (WhisPER as well as MATLAB) and starting it again.

The purpose of *not* closing the DW but making your changes effective is the following: You might have noticed that when you open the *Configure defaults' presets* DW it does not force the main program window to disappear or close. As the addition of a test procedure to a test series (or a test section respectively) or stimuli to the pool are not done in the main window but "some DWs away", having to go back and forth after changing the defaults and wanting to add a test procedure would result in a lot of unnecessary clicking. So instead the *Configure defaults' presets* DW stays open after pushing *Write*, all changes are effective, you navigate to e. g. the test section setup and add a procedure and you can still change the defaults in this still open DW, push *write* again and your new defaults are immediately in effect again.

Cancel (*button*) This *button* will close the DW discarding your changes – neither will they come to effect, nor will they be stored for later. Please note that the state of this *button* will also signal to you if changes have been made at all – it will only be enabled if they have.

Reset (*button*) This *button* will reset the values and selections in all the above explained four *tabs* – which constitute the current preset (see below) – to the values originally intended by the programmers of WHISPER and made available to you when you first got the software (as introduced above in sec. 7.1.2, sec. 7.2.2, sec. 7.3.2 and sec. 7.4.2). The reset values will also be immediately effective and will be stored for later, too – this means a "Reset" action cannot be canceled via the *Cancel button*! Please note that the state of this *button* will also signal to you if the values in the current four *tabs* differ at least in one respect from the original provided ones – it will only be enabled if they do.

Preset 1-4 (*buttons*) These *buttons* select one of the corresponding four presets. A preset comprises *all* values and selections of *all the four tabs*, and not just the momentarily visible one. So preset 1 has a tab for AP, a tab for RGT/SD, a tab for ABX and a tab for stimulus, and these four *make* preset 1, preset 2 has a tab for AP, a tab for [...] and so on. Changing tabs and entering values will only affect the *current* preset, symbolized by the preset button at the top that is pushed. Pushing *Ok*, *Write* and *Reset* will also only act on the current selected preset [19]. Changing presets by pushing a different preset button at the top will do two things: a) It will always store the preset you are leaving for later (this means you cannot have unsaved changes over the course of a preset change), and b) it will make the newly selected preset immediately effective. This preset system can be used in an even more refined way than what was described above along with the *Write button*: Leaving the *Configure defaults' presets* DW open in the background and having configured two presets differing e. g. only in the types of the adaptive mechanism and their corresponding settings, it is easily possible to first select one of those presets, add a new track to your AP test section, then select the other preset and just again add a new track – thus with just four clicks you really quickly generated two tracks containing the same procedures, albeit with totally different settings – enabling e. g. the comparison of different adaptive mechanisms.

Previously unaltered presets default to the initial defaults (cf. above, (*Reset button*), presets stay persistent over the course of exiting the program and opening it again, and the choice of which preset is currently active is persistent to ending the program and starting it up again, too.

Presets are stored each in a file in the program folder, named PC.1.mat to PC.4.mat, these files are only generated when that preset has been selected at least once.[20] WHIS-

[19] And *Cancel* will actually "act" on nothing

[20] The original defaults are present in the file PC.mat, this should better not be touched.

PER will not crash if one or even all of the PC.*n*.mat is missing – even if that preset had been the selected one – but will simply generate a new one as a clone of the original defaults values. [21]

[21] Should you wish to use more than the provided four presets, you could thus simply move (or rename) your configured PC.*n*.mat files and later exchange them or put them back – WhisPER will read them as if they had never been elsewhere.

9 Program set up

In this chapter the approach of adjusting the *program settings* will be described. These refer to technical properties which are mostly independent of methodical issues and stored in the file *PS.mat* which will be described in section 10.2.

9.1 Basic audio setup

As WhisPER currently uses the internal MATLAB function audioplayer, there are no general audio properties to be set.

9.2 Basic network setup

Basic network settings can be specified by selecting the item *program setup\network* from the menu bar. Then a DW will appear for entering host names or IP addresses respectively and port numbers of up to 6 OSC-servers. Each server is related to a single parameter which has to be determined when defining the stimuli's contents (see sec. 5.2). As in some cases one might want to address several parameters on one server, the servers need not necessarily be different. Note that the order in which the servers are defined has to be the same as the order in which the respective OSC-commands should be executed.

10 Data handling

In this chapter basic information about the program's data structure will be provided as far as it is required to make use of the functionalities implemented. The aim is not to give a complete description which one needs to make modifications or to work on upgrades or kind of these things. For the latter purposes one is directed to the *technical documentation* (see sec. 10.3). There are basically two kinds of data sets which may be accessed by the user. One contains all data belonging to a certain test series and is represented by the test series folder which may be placed at an arbitrary location (cp. sec. 5.1). The second data set is contained in the file *PS.mat* which is located in the program folder 'whisper1.0' and consists of global program settings (cp. chap. 9). Moreover there is a sub folder containing this user documentation.

10.1 The test series folder

This folder is quite essential to know for the user as it contains both the test series' empirical data set and its setup including audio data. In the following subsections the several files and subfolders, which will be automatically generated if a new test series is created, are described.

10.1.1 The file 'testseries.info'

This file contains information about the configuration of the test series and may be regarded as a documentation of the test series setup that should enable the user to understand the information provided by the log files or the export sheets. This information can be summed up by the following aspects:

- Global settings: What stimuli have been defined? How many sessions have been set?
- Decoding of test section key numbers: What are the label names? What type of test procedure is used in each test section?
- Properties of the test procedures: How are stimuli grouped and assigned to single test units (elements, objects, triads, tracks)? What scales (format/labels) have been defined? (RGT/SD) Which response paradigm and which adaption mechanism have been selected? (AP)
- Decoding of session key numbers (experimental conditions): What subsets of test units have been selected and what sequences have been defined (manually defined vs. random sequences)?

Note: The file also serves as an identifier whose existence is a premise for the test series folder to be accepted as such by the program.

10.1.2 The files 'TSD.mat' and 'TSP.mat'

These two files constitute some of the program's most important internal data storages which in general should be handled with great care and therefore not be manipulated by the common user. Though there are a few exceptions which can be applied to simplify the experimenter's life and which, besides a short general description of the files' contents, will be noted in the following two paragraphs.

TSD.mat ("test series data") In this file all so far collected empirical data (i.e. more precisely the contents of the dependent variables which are to be exported) are stored. The only occasion for the common user to manually access this file is when one wants to clear all these data or remove them from the test series. This might be necessary either if one wants a kind of "virginal" test series folder (e.g. for replication of the test series) or if one wants to reset the counter of the current test run (i.e. more precisely of both the counters of the latest completed and the next test run) in order to be able to make some changes on the test series' setup after the latter has already been completed (cp. chap. 5). The task at hand will be performed by deleting the file TSD.mat or at least moving it to a location outside the test series folder if one wants to preserve the data. Additionally it is recommended to clear the contents of the sub folders *plot*, *export* and *logfiles*. Otherwise this could lead to confusion as old and new files will be mixed and old files might be gradually overwritten (not in general but in most cases if old and new files carrying the same labels). Do not forget also to make a copy of the folders' contents if you want to preserve the data.

Important Note on SAQI! Please note, that as referring to the previous paragraph SAQI-tests resemble an exception. In case of SAQI tests the TSD.mat should NOT be deleted, because it contains all empirical data collected from all subjects so far! Instead, it should be saved after every test run in another folder to avoid loosing collected data. As already stated in section 5 you can find the TSD.mat in the current test series folder.

TSP.mat ("test series properties") Here all data concerning the test series' setup are stored. In the standard case it is not necessary for the user to manually access this file, as the setup normally is performed by using the GUI system (see chap. 5). Anyhow in some cases it might be necessary to export these configuration data to another test series folder if it is desired to use a quite similar set up there. This can be done by copying the *TSP.mat* to the destination test series folder and replacing the local file with it. Note that in such a case one also has to copy the folder *audio* if the files in it should be played.

10.1.3 The subfolder 'audio'

This subfolder contains the audio files which are copied to there by the program in the context of the stimuli's definition (cp. sec. 5.2). Note that once an audio file has been uploaded, it will not be deleted at a later time even if it is no longer used at a certain point. Of course, in the latter case this may be manually done in order to save storage space. Note furthermore that files all must have different names as otherwise the program will not be able to distinguish between them and errors will occur. This can be ensured by using the GUI system to define the stimuli. If then a .wav file of the same name already exists it will be overwritten.

10.1.4 The subfolder 'export'

This subfolder contains the *export sheets*, i.e. text files that include all so far collected empirical data (i.e. more precisely the contents of the dependent variables which are to be exported). The export sheets are saved as .csv files and therefore may directly be opened by the spreadsheet software Excel. They can also be imported to the statistical evaluation software SPSS[22]. Each file includes the data of a certain test section across all runs. Therefore there are as much files as there have been test sections defined in a test series and the information provided by a file's name only consists of the section number. After each run the export sheets are updated by being created anew. This can also be done manually at any time by using the menu item *test series**export empirical data* (see sec. 4.1).

The internal structure of an export sheet is constituted in the following way. Each line represents the data belonging to a different run. Lines are sorted in ascending run numbers. Each column comprises a different variable's values. The first line includes the column labels, i.e. the variables' names in an abbreviated format. Columns are separated by semicolons. The first variable always is the run number, the second is the subject id and the third the session key number. Further variables and columns respectively depend on the specific type of test procedure applied. They will be stated in the following three paragraphs (see also fig. 10.1).

Important Note on SAQI! Please note, that in case of an SAQI test run the previous explanation is not valid. For extracting the empirical data please look up the paragraph about SAQI in this same section.

Adaptive psychophysical procedures For an adaptive psychophysical procedure the dependent variable is the threshold value estimated in the end of each track. As there may be several tracks defined, i.e. if tracks are interleaved and/or performed under different experimental conditions (i.e. assigned to different sessions), each column comprises the threshold estimates of all the test runs but for a different track. So if the number of tracks is given by T, there will be T additional columns reserved. Thereby columns are arranged by ascending track numbers. Note that there may be empty entries because not every track necessarily has to be performed in every run. In many applications there is only one track defined and performed by each subject, so that there is only one dependent variable and one additional column respectively in the export sheet.

Repertory grid technique For the repertory grid technique there are two groups of elicited data: the first is of a qualitative format and constituted by the elicited constructs, the second is numeric data and consists of the ratings. Considering the first, the number of elicited constructs depends on the test run. For each construct there are two verbal descriptors – one for the high pole and one for the low pole[23]. So if C is the maximum number of constructs which has been reached in a run at all, there will be $2 \cdot C$ columns reserved for the storage

[22] Tested on version 15.0. Note: Use the menu item "Read text data ..." and select the option "All Files (*.*)" from the drop-down list in the file dialogue to make .csv files visible.

[23] Only those constructs which have been used for rating, i.e. those that passed the editing in between the two parts of the procedure, will be exported. Because the information about whether a descriptor evolved from either a similarity or a contrast between elements is lost, we do not speak of *initial* and *contrast pole* anymore.

of the verbal descriptors. Moreover if E is the total number of different elements involved in the rating process at all, there will be $E \cdot C$ further columns reserved for storing the rating values.

Semantic differential For the semantic differential the elicited data consists of numerical values evolved from the ratings. If O is the total number of objects involved and S is the number of items (i.e. scales), then $O \cdot S$ columns will be reserved for the storage of the dependent variables' values.

ABX Test As results from the ABX test only the number of correct answers will be exported. **Please note**: The absolute number of trials per condition is not exported! This means, that, for later statistical analysis you will have to note this number by yourself! The number of correct answers will be saved in 4th column in the export sheet, i.e. in the row of the corresponding subject.

SAQI The internal structure of an SAQI-TSD-file is constituted in two levels. On the first level one can find information about the run number, the subject ID, the session number, the subject data, the current SAQI data version and the selected scale size. The second level is located in the fourth column named 'Subject Data'. There, you will find the elicited ratings in a 49x30 cell. Each line in this cell represents the rating of one perceptual quality. Even if a quality/modification/entity was not included or rated one will find an entry about it (see table 10.1.4).

Important note!
When saving results, these are always encoded as if being assigned to the stimulus under test when compared to the reference stimulus. This direction of encoding is also retained in saved results, if test and references stimuli are chosen to vary randomly for each assessed perceptual quality. Further, ratings are encoded to reflect perceived differences according to a logical increase of the quality under test. Thus, raw SAQI ratings may be interpreted intuitively: Positive difference ratings in terms of perceived high frequency coloration, sharpness, distance or clarity etc. refer to a perception of increase distance, emphasized high frequencies and a sharper, more distant and more clear sound of the test stimulus as compared to the reference.

In case of an open-ended bipolar or a unipolar quality the value of the slider will be suitably transferred. In order to maintain an identical representation of values in the TSD.mat, in the case of the two bipolar qualities with closed ends, the subject's input will be divided by 180 before being saved. Furthermore, the entry will be set negative, if 'shifted clockwise' ('horizontal direction') or 'shifted down' ('vertical direction') was chosen. For further information about the TSD-entries please look up the schematic representation (see table 10.1.4).

For exporting and plotting the collected empirical data there are two Matlab functions available from here: //(http://dx.doi.org/10.14279/depositonce-1.
For further information on these files, please check section 9 in the SAQI Test Manual (Lindau [2014]).

Other File Formats: GridXML and GRD Since there exist some software packages on the market which are specialized in evaluating repertory grid data, export of their respective (proprietary) and most widely adopted formats was implemented into Whisper for users convenience. The Files will be placed into the export subfolder anytime you hit the menu item *test series\export empirical data* manually. (see sec. 4.1) Note: Only rating data of RGT and SD sections will be written into these formats (for obvious reasons).
Idiogrid (http://www.idiogrid.com/) is a *free* package by James W. Grice with a lot of powerful features like Multi Grid Analysis (MGA) and Adobe Illustrator (AI) export for figures (file extension: GRD).
GridXML is an open grid format suggested by Martin Fromm, author of the commercial package Gridsuite (http://www.gridsuite.de).[24] GridXML Files can be imported into Gridsuite and converted to several other formats of commercial packages like Flexigrid, RepGrid/RepIV, EnquireWithin, Gridstat and Gridcor in a second step.

[24] Walter et al. [2004]

run	subject	session	track 1	…	track T
1	sub_id(1)	ses_key(1)	θ(1, 1)	…	θ(T, 1)
⋮	⋮	⋮	⋮	⋮	⋮
R	sub_id(R)	ses_key(R)	θ(1, R)	…	θ(T, R)

T x R threshold estimates

(a) Adaptive psychophysical procedures

run	subject	session	construct 1		…	construct C		element 1			…	element E		
			low pole	high pole		low pole	high pole	construct1	…	constructC	…	construct1	…	constructC
1	sub_id(1)	ses_key(1)	lo(1, 1)	hi(1, 1)	…	lo(C, 1)	hi(C, 1)	r(1, 1, 1)	…	r(1, C, 1)	…	r(E, 1, 1)	…	r(E, C, 1)
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
R	sub_id(R)	ses_key(R)	lo(1, R)	hi(1, R)	…	lo(C, R)	hi(C, R)	r(1, 1, R)	…	r(1, C, R)	…	r(E, 1, R)	…	r(E, C, R))

C x 2 x R verbal descriptors

E x C x R ratings

(b) Repertory grid technique

run	subject	session	object 1			…	object O		
			item 1		item S		item 1	…	item S
1	sub_id(1)	ses_key(1)	r(1, 1, 1)	…	r(1, S, 1)	…	r(O, 1, 1)	…	r(O, S, 1))
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
R	sub_id(R)	ses_key(R)	r(1, 1, R)	…	r(1, S, R)	…	r(O, 1, R)	…	r(O, S, R)

O x S x R ratings

(c) Semantic differential

Figure 10.1: Schematic representations of the data export sheets (cp. text on p. 75). Note that conceptually the schemes may include missing values, as not every test unit (object, element, triad, track) necessarily has to be assigned to each test run.

run	subject	session	section 1	…	section T
1	sub_id(1)	ses_key(1)	correct answers(1, 1)	...	correct answers(T, 1)
⋮	⋮	⋮	⋮	⋮	⋮
R	sub_id(R)	ses_key(R)	correct answers(1, R)	...	correct answers(T, R)

T x R correct answers

(a) ABX

run	subject	session	session 1	session 1	session 1	…	track T
1	sub_id(1)	ses_key(1)	subject data(1, 1)	version(1, 1)	scale size(1, 1)	⋯	(T, 1)
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
R	sub_id(R)	ses_key(R)	(1, R)	(1, R)	(1, R)	…	(T, R)

T x R subject data sets

(b) Spatial Audio Quality Inventory - Level 1

Qualitiy ID	Qualitiy		Entity 7 Name	entity 7
1	'difference'		'other'	selection
⋮	⋮	... see table for details ...	⋮	⋮
48	'other'		'other'	selection

48 qualities

30 properties per quality

(c) Spatial Audio Quality Inventory - Level 2 'Subject Data'

Figure 10.2:

SAQI Empirical Data

Column number	Column name	Encoding
1	Quality_ID	1: difference, 2, 3, 4 etc.... 48: other
2	Quality	Quality Name
3	Category_ID	0: 'difference' and 'other' other items have category 1-8
4	Category	Category Name
5	Included	1: quality was included in test run 0: quality was not included in test run
6	Rated	1: quality was rated (subject heard a difference) 0: quality was not rated (subject did not hear a difference) NaN: was not presented in the test
7	Modification 1	1: modification 1 was used 0: modification 1 was not used
8	Answer 1	1: constant 2: varying periodically or otherwise rule-based 3: varying non regularly NaN: was not presented in the test
9	Modification 2	1: modification 2 was selected by the subject 0: modification 2 was not used
10	Answer 2	1: steady 0: not steady NaN: was not presented in the test
11	Modification 3	1: modification 3 was used 0: modification 3 was not used
12	Answer 3.1	1: 'depending on scene events' was used 0: 'depending on scene events' was not used NaN: was not presented in the test
13	Answer 3.2	1: 'depending on user interaction' was used 0: 'depending on user interaction' was not used NaN: was not presented in the test
14	Answer 3.3	1: 'independent' was used 0: 'independent was not used NaN: was not presented in the test
15	Rating	Value of the slider or radio button NaN: was not presented in the test
16	Entity1str	1: assessment entities were included 0: assessment entities were not included NaN: was not presented in the test
17	Entity 1	User specific name of entity 1 NaN: was not presented in the test
18	Entity1str	1: Entity 1 was selected 0: Entity 1 was not selected NaN: was not presented in the test
19	Entity2str	User specific name of entity 2 NaN: was not presented in the test

Column number	Column name	Encoding
20	Entity2str	1: Entity 2 was selected 0: Entity 2 was not selected NaN: was not presented in the test
21	Entity3str	User specific name of entity 3 NaN: was not presented in the test
22	Entity3str	1: Entity 3 was selected 0: Entity 3 was not selected NaN: was not presented in the test
23	Entity4str	User specific name of entity 4 NaN: was not presented in the test
24	Entity4str	1: Entity 4 was selected 0: Entity 4 was not selected NaN: was not presented in the test
25	Entity5str	User specific name of entity 5 NaN: was not presented in the test
26	Entity5str	1: Entity 5 was selected 0: Entity 5 was not selected NaN: was not presented in the test
27	Entity6str	User specific name of entity 6 NaN: was not presented in the test
28	Entity6str	1: Entity 6 was selected 0: Entity 6 was not selected NaN: was not presented in the test
29	Entity7str	User specific name of entity 7 NaN: was not presented in the test
30	Entity7str	1: Entity 7 was selected 0: Entity 7 was not selected NaN: was not presented in the test

Table 10.1: Content of the TSD cell 'Subject Data'

10.1.5 The subfolder 'log files'

This sub folder contains the log files of the runs that have been performed up to the current point. A new log file will always be written right after a test run has been completed. A log file's name consists of the following information in given order: the run number, the subject ID and the session key number. Note that once a log file has been created, it will not be deleted automatically even if the run counter has been reset (cp. sec. 5). So if one wants the log files to be deleted, e.g. to achieve a "virginal" test series folder, one has to empty this sub folder by hand. Note that if a log file is to be created and one of the same name already exists, the latter will be overwritten.

10.1.6 The subfolder 'plots'

This sub folder contains all data plots that have been created during the runs so far. At the program's current state plots are only produced for psychophysical adaptive procedures. Each time after a procedure of this type has been performed, a MATLAB figure depicting the respective track will be automatically created. The respective files' names consists of the following information in given order: the run number, the subject ID, the session key number, the section number and the track's number. Additionally or alternatively to the MATLAB figure format one may choose a different file format for the plot being saved in. This can be done by using the menu item *program setup\plottings* from the main window's menu bar (see. sec. 4.1). Note that once a plot has been created, it will not be deleted automatically even if the run counter has been reset (cp. sec. 5). So if one wants the plots to be deleted, e.g. to achieve a "virginal" test series folder, one has to empty this sub folder by hand. Note that if a plot is to be created and one of the same name already exists, the latter will be overwritten.

10.2 The file 'PS.mat' ("program settings")

This file is placed in the program folder and contains all settings concerning the program's basic setup. These relate to more technical properties which are (mostly) independent of methodical issues belonging to a certain test series. The standard procedure for adjusting these settings is to use the GUI system (see. chap. 9). Though there may be some cases when a different approach yields quicker results. This is for example when one wants to take along these *program settings* to another computer or to switch between several set ups very quickly and there is no time for making all the necessary inputs by hand. In such cases one just might exchange the versions of the file *PS.mat* which is located in the program folder ('whisper1.0'). Further adjustments then again can be made by using the GUI system.

10.3 The subfolder 'doc'

This sub folder is placed in the program folder ('whisper1.0'). It includes the user documentation, i.e. the document you are currently reading, the file *release_notes.txt* containing information on the history of updates and the current software version, and the file *license_notes.txt* viewing license information. For a description of both – basic ideas and

details of programming – a technical documentation is available in consultation with the FG Audiokommunikation.

Bibliography

S. Bech and N. Zacharov. *Perceptual Audio Evaluation.* John Wiley & Sons, Ltd., Chichester, 2006.

J. Berg and F. Rumsey. Spatial attribute identification and scaling by repertory grid technique and other methods. In *AES 16th International Conference Munich*, pages 51–56. Audio Engineering Society, 1999.

J. Bortz and N. Döring. *Forschungsmethoden und Evaluation.* Springer, Berlin, 4 edition, 2006.

H. Burstein. Approximation formulas for error risk and sample size in abx testing. *Journal of the Audio Engineering Society*, 36(11):879–883, 1988.

S. Ciba. Erstellung einer Softwarebibliothek für Hörversuche - Programmkonzept und zu implementierende Testverfahren. Magisterarbeit, FG Audiokommunikation, Institut für Sprache und Kommunikation, Fakultät I, TU Berlin, 2008.

D. L. Clark. Ten years of a/b/x testing. In *AES 91st Convention New York.* Audio Engineering Society, 1991.

T. N. Cornsweet. The staircase-method in psychophysics. *American Journal of Psychology*, 75:485–491, 1962.

J.M. Findlay. Estimates on probability functions: A more virulent pest. *Perception & Psychophysics*, 23(2):181–185, 1978.

F. Fransella and D. Bannister. *A manual for repertory grid technique.* academic press, London, 1977.

G.A. Gescheider. *Psychophysics: The Fundamentals.* Lawrence Erlbaum Associates, Mahwah, NJ., 1997.

J. L. Hall. Hybrid adaptive procedure for estimation of psychometric functions. *Journal of the Acoustical Society of America*, 69(6):1763–1769, 1981.

L. O. Harvey, jr. Efficient estimation of sensory thresholds. *Behavior Research Methods, Instruments & Computers*, 18(6):623–632, 1986.

N. A. Hastings and J. B. Peacock. *Statistical Distributions.* Butterworth & Co (Publishers) Ltd, London, 1975.

http://opensoundcontrol.org/spec-1_0. http://opensoundcontrol.org/spec-1_0 (18/10/2008).

G.A. Kelly. *The Psychology of Personal Constructs.* Norton, New York, 1955.

P. E. King-Smith, S. S. Grigsby, A. J. Vingrys, S. C. Benes, and A. Supowit. Efficient and unbiased modifications of the quest threshold method: Theory, simulations, experimental evaluation and practical implementation. *Vision Research*, 34(7):885–912, 1994.

S. A. Klein. Measuring, estimating and understanding the psychometric function: A commentary. *Perception & Psychophysics*, 63(8):1421–1455, 2001.

M.R. Leek. Adaptive procedures in psychophysical research. *Perception & Psychophysics*, 63 (8):1279–1292, 2001.

L. Leventhal. Type i and type ii errors in the statistical analysis of listening tests. *Journal of the Audio Engineering Society*, 34(6):437–453, 1986.

H. Levitt. Transformed up-down methods in psychoacoustics. *Journal of the Acoustical Society of America*, 49(2):467–477, 1971.

H. Lieberman and A. Pentland. Microcomputer-based estimation of psychophysical thresholds: The best pest. *Behavioral Research Methods & Instrumentation*, 14(1):21–25, 1982.

A. Lindau. *Spatial Audio Quality Inventory. Test Manual. v1.0. http:// dx.doi.org/ 10.14279/ depositonce-1*. TU Berlin, Berlin, 2014. http://dx.doi.org/10.14279/depositonce-1.

A. Lindau, V. Erbes, H.-J. Maempel, S. Lepa, F. Brinkmann, and S. Weinzierl. A spatial audio quality inventory for virtual acoustic environments (saqi). In *EAA Joint Symposium on Auralization and Ambisonics*. TU Berlin, 2014a.

A. Lindau, V. Erbes, H.-J. Maempel, S. Lepa, F. Brinkmann, and S. Weinzierl. Eine fokusgruppe für die entwicklung eines vokabulars zur sensorischen beurteilung virtueller akustischer umgebungen. In *Fortschritte der Akustik: Proc. of the 40th German Annual Conference on Acoustics, DAGA*. DEGA, 2014b.

H. F. Martz and R. A. Waller. *Bayesian reliability analysis*. John Wiley & Sons, Ltd., New York, 1982.

C.E. Osgood, G.J. Suci, and P.H. Tannenbaum. *The measurement of meaning*. University of Illinois Press, Urbana, 1967. Neuauflage der Originalausgabe von 1957.

S. Otto. Vergleichende Simulation adaptiver, psychometrischer Verfahren zur Schätzung von Wahrnehmungsschwellen. Magisterarbeit, FG Audiokommunikation, Institut für Sprache und Kommunikation, Fakultät I, TU Berlin, 2008.

A. Pentland. Maximum likelihood estimation: The best pest. *Perception & Psychophysics*, 28(4):377–379, 1980.

M. A. Pokorny. Using maximum-likelihood adaptive methods to estimate difference thresholds and points of subjective equality. *Behavior Research Methods, Instruments, & Computers*, 30(4):624–636, 1998.

M. M. Taylor. On the efficiency of psychophysical measurement. *Journal of the Acoustical Society of America*, 49(2):505–508, 1971.

M.M. Taylor and C.D. Creelman. Pest: Efficient estimates on probability functions. *Journal of the Acoustical Society of America*, 41(4):782–787, 1967.

B. Treutwein. Adaptive psychophysical procedures. *Vision Research*, 35(17):2503–2522, 1995.

B. Treutwein. Yaap: yet another adaptive procedure. *Spatial Vision*, 11(1):129–134, 1997.

O. B. Walter, A. Bacher, and Martin Fromm. A proposal for a common data exchange format for repertory grid data. *Journal of Constructivist Psychology*, 17(3):247–251, 2004.

A. B. Watson and D.G. Pelli. Quest: A bayesian adaptive psychometric method. *Perception & Psychophysics*, 33(2):113–120, 1983.

G. Wetherill. Sequential estimation of quantal response curves. *Journal of the Royal Statistical Society*, B25:1–48, 1963.

A. Wlodarski. Erstellung einer Softwarebibliothek für Hörversuche - Implementierung und Strukturierung des Quellcodes. Magisterarbeit, FG Audiokommunikation, Institut für Sprache und Kommunikation, Fakultät I, TU Berlin, 2008.

List of Figures

4.1 The *main window*. 13

5.1 DW *stimulus pool* 17
5.2 DW *edit stimulus* 18
5.3 DW *edit test sections*. 21

6.1 GUI system for subject guidance – basic screen showing information on the test run's status of progression. 22
6.2 GUI system for subject guidance – *basic screen* showing the initial instructions. 23

7.1 The subject's GUI for the performance of an adaptive psychophysical procedure. 25
7.2 DW *AP – main settings (...)* 26
7.3 DW *AP – track settings (...)* 28
7.4 DW *AP – edit staircase settings (...)* 30
7.5 DW *AP – edit PEST settings (...)* 31
7.6 DW *AP – edit ZEST settings (...)* 33
7.7 The subject's GUI for the performance of the RGT's first part. 39
7.8 DW *RGT – edit constructs (...)* 41
7.9 The subject's GUI for the performance of the RGT's second part. 42
7.10 DW *RGT – main settings (...)* 44
7.11 DW *RGT – triad/element sequence setup (...)* 46
7.12 DW *RGT – edit scale format (...)* 47
7.13 The subject's GUI for the performance of the SD. 48
7.14 DW *SD – main settings (...)* 49
7.15 DW *SD – object pool (...)* (see text on p. 48) 50
7.16 DW *SD – edit scales (...)* 51
7.17 DW *SD – object sequence setup (...)* 52
7.18 The subject's GUI through one A/B/X Trial. 54
7.19 ABX Trial Main Settings 55
7.20 Assigning the stimuli to be judged in the current trial. 56
7.21 Example presenting a name and circumscription of a auditory quality 58
7.22 Example of a quality to be rated using a bipolar, open-ended rating scale . . 59
7.23 Example of a quality to be rated using a unipolar, closed/open-ended rating scale 59
7.24 Example of a quality to be rated using direct bipolar, closed-ended rating . . 60
7.25 Example of a quality to be rated using a dichotomous scale 60
7.26 GUI for the rating the temporal variability of perceived differences 61
7.27 GUI for the rating the cause of perceived differences 61
7.28 GUI for assigning assessment entities to perceived differences 62
7.29 SAQI Main Settings 62
7.30 GUI for the selection of perceptual qualities. Left: singular selection, right: category-based selection. 63

7.31 Default assessment entities 64

8.1 DW *Configure defaults' presets* 67
8.2 DW *Configure defaults' presets* – tab "Stimulus" 69

10.1 Schematic representations of the TSD content. 78
10.2 Schematic representations of the TSD content. 79

Note: All depicted screen shots of the GUI system have been captured under Windows XP.