



# Audio Engineering Society Convention Paper

Presented at the 126th Convention  
2009 May 7–10 Munich, Germany

*The papers at this Convention have been selected on the basis of a submitted abstract and extended precis that have been peer reviewed by at least two qualified anonymous reviewers. This convention paper has been reproduced from the author's advance manuscript, without editing, corrections, or consideration by the Review Board. The AES takes no responsibility for the contents. Additional papers may be obtained by sending request and remittance to Audio Engineering Society, 60 East 42<sup>nd</sup> Street, New York, New York 10165-2520, USA; also see [www.aes.org](http://www.aes.org). All rights reserved. Reproduction of this paper, or any portion thereof, is not permitted without direct permission from the Journal of the Audio Engineering Society.*

## WhisPER – A new tool for performing listening tests

Simon Ciba<sup>1</sup>, André Wlodarski<sup>1</sup>, Hans-Joachim Maempel<sup>1</sup>

<sup>1</sup> Audio Communication Group, TU Berlin  
Sekt. EN-08, Einsteinufer 17c, 10587 Berlin, Germany

[sciba2@hotmail.com](mailto:sciba2@hotmail.com), [andre@wlodarski.net](mailto:andre@wlodarski.net), [hans-joachim.maempel@tu-berlin.de](mailto:hans-joachim.maempel@tu-berlin.de)

### ABSTRACT

A software tool is presented for performing experiments in the field of perceptual audio evaluation and psychoacoustic measurement, controlling the interaction with both the subject and the playback environment. For this purpose a repertoire of test procedures has been implemented, including popular qualitative and quantitative approaches. By using OpenSound Control commands, not only traditional multi-channel reproduction is supported, but also advanced spatial audio reproduction such as dynamic binaural synthesis or wavefield synthesis. WhisPER has been written in MATLAB to facilitate its further development within the scientific community. As opposed to existing libraries it provides a coherent graphical user interface system allowing easier access and configuration also for users without advanced programming experience.

### 1. INTRODUCTION

For today's listening tests computers are indispensable tools. They are used to generate stimuli, to instruct subjects, to control test procedures, to analyze test results, and they are of increasing importance for the control of audio reproduction systems, particularly when the auralisation of virtual acoustic scenes by binaural technology or multichannel loudspeaker reproduction is involved. Although a variety of software tools is available for commercial or non-commercial use (for a non-exhaustive overview see [1], chap. 9.2.3), they usually cover only a limited repertoire of

standardized test methods, and they provide only limited playback options mostly restricted to mono or stereo reproduction. In the case of commercial software, when no source code is available, tests are often difficult to adjust to individual requirements or a particular laboratory environment. For these reasons the software for listening tests is quite often written by the investigators themselves, usually in high-level interpreted and interactive languages to achieve and maintain flexibility. However, programming can still be a tedious task, prone to algorithmic and maybe unnoticed errors, so that approaches towards an integrated test control environment have already been made. One example is the software package PsyLab [2] providing a collection of MATLAB scripts for the

specification and presentation of sound stimuli and the collection of subject responses. It employs threshold seeking algorithms based on simple and transformed staircase methods combined with the forced choice paradigm. Similar developments have taken place in other scientific domains. The Psychophysics Toolbox [3] focuses on visual research and supplies a variety of MATLAB routines which may be embedded into personal script writing. Even though these solutions offer a high degree of flexibility and expandability they still require more or less advanced programming skills to set up an experiment.

Due to these limitations inherent in all programs known to the authors, a new software is presented here. It was developed to provide

- a modular and easy-to-extend software library, running on a widespread algorithmic language,
- a collection of important test procedures,
- a graphical user interface easily accessible to experimenters without advanced programming skills and
- integrated options to control complex sound reproduction environments.

The paper will give an overview on the program's conceptual structure and the implemented features, some insight into the programming style, and an account on its current development status.

## 2. THE SOFTWARE

### 2.1. Running environment

WhisPER consists of scripts and functions designed for MATLAB, which is thus required for running the software. MATLAB was chosen because of several reasons:

- the rich library of high-level functions which makes a fast development possible,
- the ability to design graphical user interfaces and do plotting easily,
- its wide-spread availability in the audio community,

- the (almost) platform independent environment.

WhisPER is developed for the Windows version of MATLAB 7. Porting WhisPER to the Mac or Linux versions of MATLAB will be possible by low effort. The scripts do not require any of the supplementary MATLAB toolboxes. For the generation of plots in PDF format Ghostscript, a freely available postscript interpreter, is used. An installation is therefore recommended but not essential since other formats can be used for saving plots.

### 2.2. The user interface

The user interface of WhisPER (for its structural outline see figure 1) is divided into two parts. One is dedicated to the investigator and one to the subject. Both parts are connected, while the latter is controlled by the former and accessed in the beginning of each run. The graphical user interface (GUI) of WhisPER allows access to almost all features of the program.

#### 2.2.1. Basic characteristics

As one of its basic characteristics the user interface allows to control runs corresponding to one coherent investigation within a virtual environment called “test series”. This provides an organization of both the empirical data and the experimental setup.

Quite often a set of listening tests is conducted by applying the same test method to different subjects, with some of the runs performed under partially different experimental conditions than others. Such a variation for instance may refer to the selection of stimuli or the order of their presentation. In this case runs under different conditions need sets of slightly different specifications while a large amount of parameter values stay constant. To avoid redundancy in the configuration process, WhisPER provides a structural framework which allows organizing subsets of runs assigned to different “sessions”. This feature shall facilitate the application of factorial designs and especially the testing of sequence effects.

Furthermore, the program allows to divide a test run into several sections potentially related to different test methods. One possible application is the configuration of a familiarization stage to precede the actual listening test.

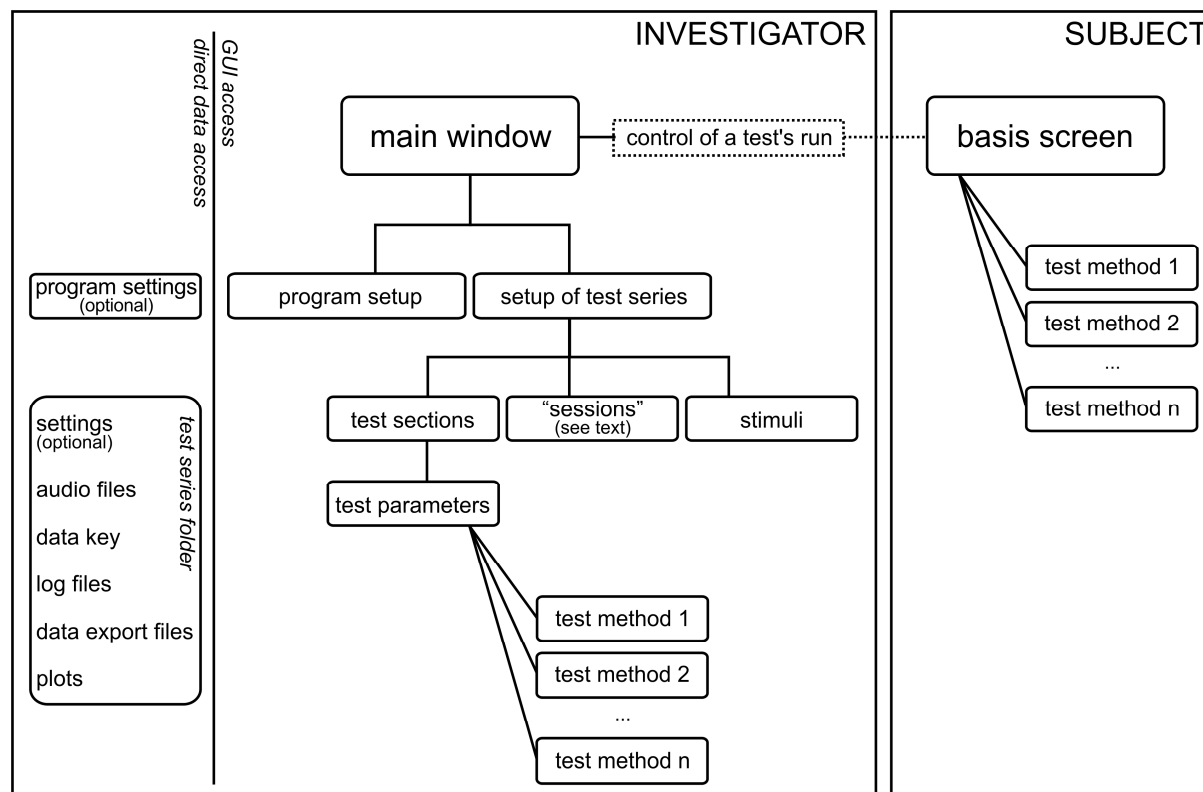


Figure 1: The structural body of the user interface.

The GUI system is structured in a way that windows of higher hierarchical layers are independent of a certain test method. Only windows of lower layers refer to the method. Also windows belonging to different methods are independent from each other. This principle facilitates expansion by new test methods as further windows may easily be integrated.

### 2.2.2. The experimenter's interface

The experimenter's "main window" appears after starting the program (screenshot in figure 2). It provides a general program setup, a test series setup and the control of one single experimental run. The program setup includes the configuration of basic behavior such as communication with external hard- and software components. The test series setup (sec. 2.2.1) includes the configuration of experimental conditions, the structural and methodical organization of the runs, i.e. the arrangement of test sections and the assignment of test methods and their parameter values.

The graphical user interface (GUI) provides access to most of the program's features. In certain cases the direct handling of data by manual operations outside the user interface provides a time saving alternative to the GUI access (labeled as "optional" in fig. 1). However this requires an experienced user and is not essential.

Generally excluded from GUI access are output data which are stored in specifically formatted files including log files, sets of empirical data, and data plots. While log files are saved in plain text, plots can be exported as EPS, PNG, PDF and MATLAB's FIG format to provide wide compatibility. Special attention was directed to the empirical data sets elicited during a run, where the program accomplishes automatic export to suitably formatted CSV files which may be directly opened in spreadsheet or statistical packages.

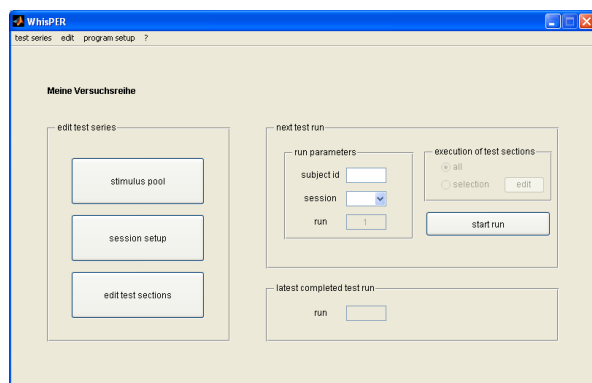


Figure 2: The investigator's main window.

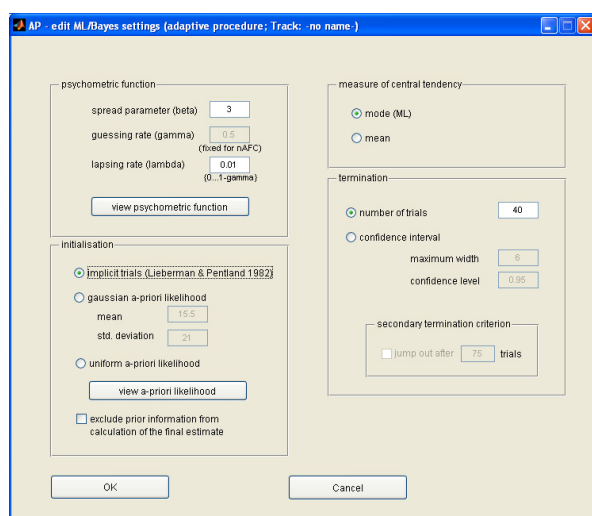


Figure 3: Exemplary window for the parameter setup of a Maximum-Likelihood/Bayesian adaptive procedure (see sec. 2.3.1).

### 2.2.3. The subject's interface

In the beginning of each run the subject's GUI is activated by the experimenter by setting a minimum number of parameters necessary for the identification of subject and experimental setup. After an initial basis screen (fig. 4) with instructions and status information the subject will be confronted with windows depending on the current test method. Exemplary screenshots for the implemented test methods (see sec. 2.3) are shown in figures 5, 6, and 7.

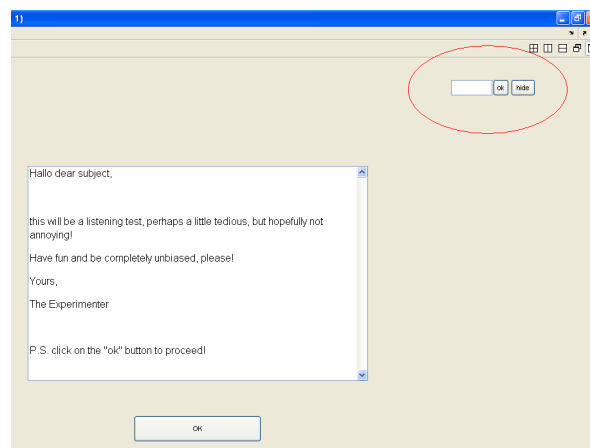


Figure 4: The subject's basis screen currently showing initial instructions. The window permanently stays in the background during a run. Its content alternates between status information and instructions for subject guidance. The window also comprises a mechanism for early cancellation of a run by the experimenter, which is accessible by a code word (see right upper corner)

## 2.3. Implemented tests

WhisPER currently provides three types of tests: A selection of adaptive psychophysical methods, the Repertory Grid Technique (RGT) and the Semantic Differential. A short overview will be given, without a detailed description of the methods and their theoretical background.

### 2.3.1. Adaptive psychophysical methods

Adaptive psychophysical procedures estimate thresholds as single points on a psychometric function. The intensity of a one-dimensional stimulus attribute is varied and the subject's response is collected in a binary format. Adaptive procedures are expected to yield a more efficient threshold measurement than classical methods, e.g. the method of constants, by adapting the stimulus intensity according to the history of preceding responses. They can be distinguished by the adaption rule and the method for estimating the final output. The current implementation includes simple and transformed staircase procedures, parameter estimation by sequential testing (PEST) and a maximum likelihood (ML) approach which may be extended by elements of Bayesian statistics.



Figure 5: The subject's interface for an adaptive procedure combined with a 3-AFC paradigm

The staircase procedures as described by Levitt [4] are implemented with 1-down/1-up (simple staircase method), 2-down/1-up and 3-down/1-up tracking rules (transformed staircase methods), a stopping rule ending the run after a certain number of reversals and a method for threshold estimation by averaging stimulus intensity over a certain number of last consecutive reversals. The PEST procedure is implemented according to the original version by Taylor and Creelman [5] and in a modified form referred to as A More Virulent PEST [6]. The ML procedure is based on the Best PEST algorithm by Lieberman and Pentland [7] but includes some optional extensions which rest upon the ideas of other authors and have been derived from Bayesian theory. These refer to the inclusion of prior information, the estimation method (i.e. the measure of central tendency) and the stopping rule. As far as the first aspect is concerned, the experimenter may either use a non-informative, i.e. a uniform a-priori density, a Gaussian one as suggested by Watson and Pelli [8] or an initialization procedure by defining implicit trials as described by the authors of Best PEST. For the measure of central tendency one may either select the mode of the likelihood function which constitutes a ML estimator, or the mean, which is additionally embedded in the more general Bayesian approach and suggested by Emerson [9] and King-Smith et al. [10]. The termination of a ML/Bayesian procedure's run may either be achieved by employing the traditional stopping rule (i.e. stopping after a fixed number of trials) or by a dynamic criterion which catches as soon as the confidence interval of the current estimate falls below a predetermined limit. The calculation of the current

confidence interval is carried out by assuming a two-sided symmetrical Bayesian probability interval as suggested by Treutwein [11, see eq. 23].

These adaptive methods can be used in combination with a n-fold alternative forced choice (nAFC) paradigm. Random interleaving of several tracks is supported. For a comprehensive overview of the group of adaptive procedures including an illustration of underlying theoretical concepts the reader is referred to [11].

### 2.3.2. Repertory Grid Technique

The Repertory Grid Technique (RGT) is a hybrid test combining a qualitative and a quantitative approach. It allows the elicitation and the arrangement of constructs on which judgments of subjects are based. It was devised for psychological application on the basis of Kelly's theory of personal constructs [12] and has later been introduced into audio research by Berg and Rumsey [13]. The method has already been implemented in a computer program by Berg [14]. The performance of the procedure is split in two parts. During the first part stimuli are combined to groups of three (called triads) and presented to the subject that is supposed to report on attributes that are crucial for similarities and contrasts among the stimuli. The result of each triad comparison is a pair of opposite terms or phrases, referred to as a bipolar construct. In the second part the stimuli are graded on scales formed by the elicited constructs. For the latter different possibilities (dichotomization, ranking, category rating) exist, out of which the rating method (cp. semantic differential below) was chosen.

In the implementation provided by the authors the generation of triads may be carried out either by an automatic variation of the stimuli over the triads or a manual distribution. Similar applies to the order of trials which may be determined manually or by automatically generated play lists.

As sometimes investigators allow subjects to report on more than one construct per triad [13], the program offers to repeat the elicitation step if desired by the subject. In between the procedure's two parts there is a chance for editing the elicited constructs by the investigator which in some cases might be desired, especially when employing the program for an automatized data collection.

Figure 6: The subject's interface for the elicitation of triads, i.e. the first part of the RGT.

Figure 7: The subject's interface for the Semantic Differential.

### 2.3.3. Semantic Differential

The semantic differential according to Osgood et al. [15] uses a set of category rating scales for the

multidimensional measurement of gradual aesthetic judgments, often applied to audio stimuli.

The current implementation supports up to 30 definable items in gradations selectable from 2 to 11 categories which may be labeled by arbitrary marks. It enables further settings on scale format as well as supplying the possibility of random play lists.

### 2.4. Playback environment

WhisPER offers two options for the stimuli's playback. They can be played directly as WAV files using the audio interface of the local machine. This is done via the DirectSound driver (under Microsoft Windows), supported by virtually every audio interface on the market. DirectSound currently offers playback of up to two audio channels. The second option is to control external playback via OpenSound Control (OSC) [16] commands. An external program may be installed on the same or on another machine, as OSC works via standard TCP/IP network protocol. Although playback of the stimuli is done externally, WhisPER keeps full control. It allows sending up to 6 OSC commands per stimulus, because, depending on the playback environment, multiple commands can be necessary for playing one stimulus. E.g. the first command could control the playback channel while the second one would start the playback. The commands can be sent to one or more OSC servers. Thus a variety of different kinds of playback environments may be controlled, including multi-channel reproduction, dynamic binaural synthesis and wavefield synthesis (WFS). Even the presentation of visual material is possible.

## 3. PROGRAMMING ISSUES AND EXPANDABILITY

One of the aims was to create a program that easily enables further development by the scientific community. Due to this a functional and modular programming style was realized. Furthermore variables are organized in hierarchical and universal structures. The different types of data belonging to one test series are stored in a conjoint folder but in separate files and formats (see "test series folder" in fig. 1). This way those may be accessed independently from each other.

All in all WhisPER offers some structural dispositions on which future programmers may build when expanding the program by implementations of new test methods:

- Its GUI system is structurally open for expansion (see sec. 2.2.1).
- It provides the framework for the organization and the playback of the stimuli's contents (see also sec. 2.4). If using the local audio interface for two-channel sound reproduction, by now the MATLAB function "audioplayer" is involved.
- An environment for organizing and saving empirical data is available.
- There is a structured storage for saving a test's parameter settings.
- It provides a general frame to organize variations of experimental conditions by grouping them into "sessions" (see also sec. 2.2.1).
- WhisPER offers sub-routines that make generating log files easy.

#### 4. DEVELOPMENT STATUS AND AVAILABILITY

The present first version of WhisPER necessitates still work on evaluation and bug fixing. There also might be some adjustments of the user interface system necessary in order to improve usability. As described above, its structural constitution implies future work on upgrades in terms of adding new testing paradigms. To facilitate its further development within the scientific community, WhisPER will be freely available. Information about the project and the latest software version can be obtained on the audio communication group website (<http://www.ak.tu-berlin.de/>) or on the project website (<http://www.whisper-evaluation.de/>).

#### 5. ACKNOWLEDGMENT

The authors wish to thank the following people: Alexander Lindau for helpful advice and support during all steps of development; Bernhard Treutwein for delivering insight into the Modula-2 code of his YAAP procedure and providing his knowledge in critical moments during the implementation of the Bayesian method; Stefanie Otto for an inspiring discussion on the topic of adaptive procedures; Stefan Weinzierl for valuable contributions to shape and content of this paper.

#### 6. REFERENCES

- [1] Bech, S. and Zacharov, N. (2006): *Perceptual Audio Evaluation*. John Wiley & Sons, Ltd., Chichester.
- [2] Hansen, M. (2006): Lehre und Ausbildung in Psychoakustik mit psylab: freie Software für psychoakustische Experimente. In *Fortschritte der Akustik*, DAGA, Braunschweig.
- [3] Brainard, D.H. (1997): The Psychophysics Toolbox, *Spatial Vision*, 10:443-446.
- [4] Levitt, H. (1971): Transformed up-down methods in psychoacoustics. *Journal of the Acoustical Society of America*, 49(2):467-477.
- [5] Taylor, M.M. and Creelman, C.D. (1967): PEST: Efficient estimates on probability functions. *Journal of the Acoustical Society of America*, 41(4):782-787.
- [6] Findlay, J.M. (1978): Estimates on probability functions: A more virulent PEST. *Perception & Psychophysics*, 23(2):181-185.
- [7] Lieberman, H. and Pentland, A. (1982): Microcomputer-based estimation of psychophysical thresholds: The Best PEST. *Behavioral Research Methods & Instrumentation*, 14(1):21-25.
- [8] Watson, A.B. and Pelli, D. (1983): QUEST: A Bayesian adaptive psychometric method. *Perception & Psychophysics*, 33(2):113-120.
- [9] Emerson, P. L. (1986): A quadrature method for bayesian sequential threshold estimation. *Perception & Psychophysics*, 39:381-383.
- [10] King-Smith, P.E., Grigsby, S.S., Vingrys, A.J., Benes, S.C. and Supowit, A. (1994): Efficient and unbiased modifications of the QUEST threshold method: Theory, simulations, experimental evaluation and practical implementation. *Vision Research*, 34(7):885-912.
- [11] Treutwein, B. (1995): Adaptive psychophysical procedures. *Vision Research*, 35(17):2503-2522.
- [12] Kelly, G. (1955): *The psychology of personal constructs*. Norton, New York.

- [13] Berg, J. and Rumsey, F. (1999): Spatial attribute identification and scaling by repertory grid technique and other methods. In *Proceedings of the AES 16th International Conference*, Munich, pp. 51–56.
- [14] Berg, J. (2005): OPAQUE - a tool for the elicitation and grading of audioquality attributes. In *Proceedings of The 118th AES Convention*. Barcelona.
- [15] Osgood, C.E., Suci, G.J. and Tannenbaum, P.H. (1957): *The measurement of meaning*. University of Illinois Press, Urbana. Reprint of the original edition from 1957.
- [16] opensoundcontrol.org [Online] March 2009, <http://opensoundcontrol.org/>