Auditory Model Initiative Framework

This is a description of the model framework. It describes its aim, its components including tables with definitions and existing example modules and it briefly explains how to contribute. It is not an operation manual but contains exemplary operation instructions in the appendix.

1 Aim

This framework is intended to be useful to various groups:

i. For experimentalists who want to test several existing models (theories) on their new data set. Specifically to better contradict certain theories.

ii. For model developers who want to quantitatively test their model on several cardinal data sets and compare the performance to other models

iii. For model developers who want to make their code publicly available without adopting it to the coding conventions of a specific toolbox.

iv. For setting up “model challenges” where competing models can be fitted to training data and then be compared on later released test data.

We are seven auditory model developers, all from different institutions, sharing the motivation to be able to compare various models on fair grounds and on a broad range of data sets without the need to change the model when changing the test stimulus. We came to realize that most of the models that we have developed in the past (e.g., Stern and Shear, 1997; Marquardt and McAlpine, 2007; Dietz et al. 2009; Ewert et al. 2010; Goodman et al. 2013; Hartmann et al. 2016; Majdak? 20XX??) do not meet our own expectations in this regard. We are not able to compare the majority of our models in their published form.

Our own background has a focus on models of binaural interaction and, albeit to a much lesser extent, on human psychoacoustics. The binaural focus is kept for this first version, in order to keep the project manageable. If fact we even narrow it down to models of normal binaural hearing for the time being. This is notwithstanding, that the proposed framework should be able to serve all other auditory modelling disciplines with little or no modifications.

Our loose secondary focus on psychoacoustics is non-exclusive. We want to include pure neural models from the outset, since binaural modelling takes much of its momentum from the apparent contradictions between physiologic and perception oriented model concepts. It may explain why most of the examples are on psychoacoustic experiments and model predictions. However, one aspect that led us to develop this framework is specific to models of auditory perception: Most models to not replicate the actual psychoacoustic experiment, but rather give out a specific parameter that is mapped to the experimental result quantity. For example a cross-correlation coefficient is mapped to a threshold ITD, obtained from an adaptive staircase procedure. We have identified this indirect modelling as one of the key limitations for comparing models. Therefore, we propose as the only principal requirement for participation in this initiative, that the model performs the exact same procedure that was used for the underlying experiment. The model output has to be in the same dimension as the subject response. In the case of psychoacoustics we propose to use a task-specific model back-end called “artificial observer” and a task independent auditory pathway model.

The final aim is to keep the entry bar for contributors as low as possible. We also acknowledge that model developers do not necessarily want to change the programming of their own model to the conventions of existing toolboxes such as the Auditory Model Toolbox. Therefore, we do not ask for specific coding conventions. The only requirement is to run the model in the exact same paradigm as it was used in the experimental study.

2 Framework structure

In order to simulate a listening experiment, we define a structure with four objects: (1) pathway model; (2) artificial observer; (3) experiment; (4) experiment-model interface. Each object is introduced and defined in the following subsections. For each object several examples are provided which can be used or own modules can be written.

Currently, the framework is based on file exchange. This means that the receiving objects are waiting until a file appears in a predefined directory. The file is created by the sending objects. In the following, the interface assumes that all objects have access to the same directory of a file system. Such a file-based interface has the advantage of a simple definition and of a platform independent communication. On the other hand, both experiment and model need access to the same directory of the same file system. However, this is not as much of an inconvenience as it might first appear, as directories on different computers or in cloud storage can be mapped to local directories.

As said above, the primary focus is on modelling the processing of sound within the ears and brain. Given this focus, the default input to a model is the sound as it is received at the ear. This still leaves some room for further definition (such as level of the pinna vs. level of the ear drum). To emphasize the focus on 2-channel sound input, wav-audio files are used as the default.

2.1 Pathway model

This object handles binaural signals and calculates an internal representation (e.g. the output of a binaural neuron, or a cross-correlogram, or just the centroid of a cross-correlogram or ITD(f,t))

Type	**Name**	**Format**	**Explanation**
Input	Stimulus	double N x 2	Digital representation of input sound. One interval only! Default is 2-channel.
Input	fs	double	Sample rate in Hz
Input optional		Arbitrary variables	e.g. feedback, a-priori information, model settings
Output		Arbitrary variable	auditory pathway model output "internal representation" (everything is possible: scalar value, spike train, cross correlogram, structure)

Table 1: Pathway model conventions

Exemplary pathway model functions:

```
cross_correlogram = Bernstein_Trahiotis_crosscor_2012(stimulus,fs,f_mod)
cross_correlation_coeff=klein_hennig_2011(stimulus,fs,adapt_loops,internal_noise_ratio)

cross_correlation_coeff=NCC(stim,noise,time_constant_array)

spike_count_array=spiking_model(stim,number_neurons,noise)
```

2.2 Artificial observer

This object outputs a decision based on one or several internal representations of a pathway model. The output has to be the same as from the real subject (e.g. number of target interval in case of an alternative forced choice experiment or spike train in case this the actual experiment was an extracellular recording)

Type	Name	Format	Explanation
Input	path way_ out	Cell Array	Output from pathway stages. Note this is a cell-array containing the output from all N intervals.
Input optional		Arbitrary variables	e.g. feedback, a-priori information, detector settings, sampling rate
Output	resp onse	Arbitrary variable (ideally csv-file compatible)	Same format as experimental procedure (e.g. interval number for AFC, angle for localization, spike train in physiology where detector output=input)

Table 3: Artificial observer conventions for Matlab / Octave

```
function response = argmin(pathway_out)
function response = centroid_lateralization(pathway_out,response_type_is_IID)

lowest_cross_correlation()
max_index()
```

2.3 Experiment

This object handles the experiment by preparing signals for the pathway model and potentially receives input from the artificial observer. It stores the experimental results and may have routines for plotting and analyzing data. It is thought to be the same software as used for the actual experiment. Requirement: save sound output as file.

2.4 Experiment-model interface:

This object calls the pathway model for each experimental interval, and calls the decision model for each experimental trial. It handles the data I/O between the other objects.

Currently two versions are available:

i. Matlab only. Model and artificial observer must be in Matlab and are called by this function. The function call specifies # of intervals, name and parameters of pathway model, and name and parameters of artificial observer. Experiment is started separately – does not have to be in Matlab. If also in Matlab a second Matlab has to be started.

ii. Python based platform independent interface. Each component can be in a different programming language. Supports Matlab, Python and octave. Model and detectors are called using the python “model_server_python” function.

3 Contributing

If you want to contribute any object (model, observer, experiment) at this stage simply email it to us after you tested it with the framework. With respect to experiments we are primarily interested in theory-challenging experiments, ideally published and together with the original plot and analysis scripts.

We are an open group with limited resources and no specific funding. Therefore, feedback, bug-reports and requests for active participation are appreciated; however, please accept our apologies that we cannot guarantee consideration of contributions or requests. We hope to be able to provide a git-based and/or a webpage based contribution channel at some point.

4 Appendix

4.1 Folder structure

4.2 Exemplary operation instruction for the pure Matlab framework

4.3 Table of currently implemented models

Model name	References	Function name	Input param.	Output	Code
Cross-Correlogram	Bernstein & Trahiotis 2012 using CC-Toolbox (Akeroyd 2001)	Bernstein_Trahiotis_crosscor_2012	Modulation frequency f_mod (a-priori knowledge on stimulus)	Cross-correlogram	Matlab / Octave

Cross correlation coefficient	Klein-Hennig et al. 2011 based on Dau et al. 1996 and Bernstein & Trahiotis 1996	klein_hennig_2011	adapt_loops (number of adaptation loops: 0,1, or 5) internal_noise_ratio (newly introduced to work with art. observer)	Correlation coefficient (scalar value)	Matlab / Octave
Dummy EI	N/A (example code to be substituted by Breebaart EI)	Super_simple_EI	N/A	Time trace with difference between left and right	Matlab / Octave
Cross-Correlation coefficient	Klein-Hennig et al. 2011 based on Dau et al. 1996 and Bernstein & Trahiotis 1996	klein_hennig_2011_python	Stimulus, noise, time constant of the adaptation loops	Correlation coefficient (scalar value)	Python2
Goodman_brette_2010_python	Based on Goodman and Brette, 2010	Goodman_brette_2010_python	Stimulus, number of neurons, noise	Index of the neuron in the network which fired the most	Python2

Table 2: List of available pathway models (as of May 2017)

4.4 Table of currently implemented artificial observers

AO name	References	Function name	Input param.	Output	Code
CC-Centroid	Stern and Shear 1997.Implemen. as in B&T 2012 using CC-Toolbox (Akeroyd 2001)	centroid_lateralization	One Cross-correlog.; response_type_is_IID set as 1 to measure centroid in units of pointer IID or 0 to measure as ITD	Centroid of the c-correlogram in units of pointer IID (scalar value)	Matlab / Octave
argmin	N/A. Inspired by Bernstein and Trahiotis 1996	argmin	N scalar values	Interval number with smallest input	Matlab / Octave

Fluctuation detector	N/A. Inspired by Breebaart et al. 2001 and Goupell 2010	largest_residual	N intervals (time traces) processed by an E-I unit (e.g. super-simple-EI)	Interval with largest E-I output (target in NoSpi task)	Matlab / Octave
argmax		max_index	pathway_out structure	Return argmax value	Python2
argmin	N/A. Inspired by Bernstein and Trahiotis 1996	lowest_cross_correlation	Void. Load numpy array from pathway_out.mat file	Return argmin value	Python2

Table 4: List of available artificial observers

4.5 Table of currently available experiments

Exp name	**References**	**Function name**	**Exp. environment**	**Plotting & Analysis**	**Code**
Raised sine lateralization	Model only reimplementation of Bernstein and Trahiotis 2012 w/o the low-pass noise	RaisedSine_Lateralization_Bernstein_Trahiotis_2012	m-file only	Bernstein_Trahiotis_2012_plot Perfect reproduction of paper model data (different plot format)	Matlab / Octave
Envelope ITD JND	Klein-Hennig el al. 2011 (Mostly original code partly reimpl. No low-pass noise. Listening test possible. Works for 9 out of 10 experiments)	KleinHennig2011	AFC software toolbox by Ewert 2013	afc_plot Simple boxplot script. Qualitatively correct but not identical to paper due to use of artificial obser.	Matlab / Octave

Table 5: List of existing experiments