DOI: 10.61782/fa.2023.0247

USING AUDITORY MODELS TO MIMIC HUMAN LISTENERS IN REVERSE CORRELATION EXPERIMENTS FROM THE FASTACI TOOLBOX

Alejandro Osses **Léo Varnet**
École normale supérieure, PSL University, CNRS, 75005 Paris, France
alejandro.osses@ens.psl.eu, leo.varnet@ens.psl.eu

ABSTRACT

The fastACI toolbox is a set of MATLAB routines that allows to prepare and perform behavioural listening experiments. The main feature of this toolbox is the inclusion of a reverse correlation (revcorr) post-processing module. The revcorr method allows the assessment of acoustic cues used by a listener with only minimal a priori knowledge of the actual listener's strategy during the experiments. Another special fastACI feature is the possibility to replicate experiments using an artificial listener, consisting of a hearing-inspired model and a decision module which convert trial-by-trial waveforms into a decision variable (e.g., discriminated or not discriminated). In this contribution we explain how to set an artificial listener using an auditory model from the AMT toolbox, combined with one of the decision back-ends that are available in the fastACI toolbox. The selected decision back-ends are all based on a template-matching approach. We provide a list of assumptions and the steps required to derive templates to mimic the performance of human listeners. The emphasis of this contribution is on showing the flexibility of the fastACI framework where any new experiment can be tested using either human or artificial listeners and how to strategically compare collected or simulated results.

Keywords: Auditory modelling, reverse correlation, tone-in-noise

1. INTRODUCTION

Auditory reverse correlation (revcorr) is an experimental paradigm that allows to derive which acoustic features are effectively used by participants engaged in an auditory task. This paradigm was first proposed by Ahumada and colleagues in a set of tone-in-noise experiments [1,2]. In our lab, we have applied reverse correlation to obtain participants' behavioural templates—that we have named auditory classification images (ACIs)—in tasks such as amplitude modulation (AM) perception [3] and speech (phoneme-in-noise) perception [4, 5]. A revcorr experiment basically means that the trial-by-trial behavioural responses for a specific task are related to the properties of the specific sound interval, as a post-processing scheme after data collection. However, for this to be possible, the trials need to be subjected to some form of random fluctuation that interacts with the target sounds, as elicited by a source of external variability, most typically, using background noises. Other practical considerations are that: (1) the target sounds need to be contrasted to each other (e.g., having the same length or sharing some acoustic similarity), and that (2) a considerable number of trials leading to correct and incorrect responses need to be collected. In this sense many listening experiments could be processed using revcorr. Examples of sounds that can (directly) be contrasted are those used in classic psychoacoustic experiments where, e.g., tone-in-noise sounds are compared with noise-alone sounds [1,6] or AM sounds are compared with non-AM sounds [6]. With the goal of enabling these experimental paradigms for revcorr, we built the fastACI toolbox.

A versatile and popular experimental package to conduct psychoacoustic experiments is the so-called AFC toolbox [7] for MATLAB. AFC allows to modularise listening experiments in a set of compulsory and optional scripts and offers the possibility to run the experiments using either human (normal experiments) or artificial listeners. For this reason, fastACI experiments follow a similar rationale, requiring minimal modifications to experiments coded in AFC to be compatible with our toolbox. In this

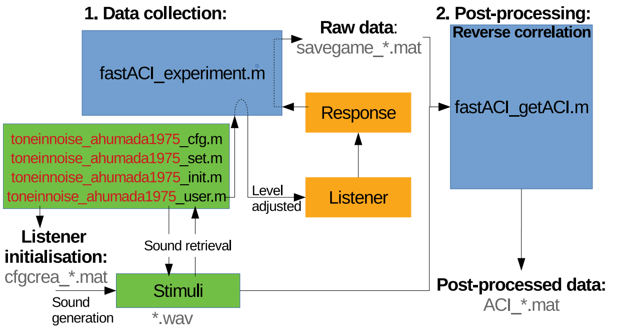

Figure 1: Schematic flowchart of the fastACI toolbox when running the experiment 'toneinnoise_ahumada1975."

sense, fastACI experiments can also be run using either human or artificial listeners. The current study focuses on detailing how an auditory model can be added as an artificial listener in fastACI. Although this feature was already available in the toolbox [3, 8], this is the first time we provide a step-by-step guide about how to include an auditory model as a ready-to-use artificial listener in our toolbox.

2. METHODS

2.1 The fastACI toolbox

The fastACI toolbox developed in our group is freely available online [9]. A description of the revcorr approach can be found in [4, 10], but a schematic flow chart for the execution of a listening experiment is shown in Fig. 1. The block "listener" can either be a human listener, with trials being actually presented (e.g., via headphones with responses recorded using the computer keyboard), or an artificial listener. The focus of this contribution is on this latter mode. An artificial listener consists of an auditory model used as a preprocessing module, attached to one of the available decision back ends. This process is schematised using a tone-in-noise experiment that is already available within the toolbox, with an artificial listener based on the model king2019 [11] from the AMT toolbox [12].

2.2 The test listening experiment

2.2.1 Suitable for the revcorr approach?

The experiments we have evaluated so far in fastACI are based on single-interval trials and the use of random background noises to detriment the performance of the participants. It is the randomness in the stimuli, i.e., the external variability, what it is actually used by revcorr to derive the effective acoustic cues of the (human or artificial) participant or, in other words, to assess a behavioural template that better explains the participant's "listening strategy" adopted during the experiment.

In other words, the revcorr approach assumes a dominant role of the trial external variability on behavioural performance. Considering background noises as the most common source of trial variability, the target itself is often omitted from the revcorr analysis [13]. Therefore, listening experiments with deterministic trials are not suitable for revcorr. Examples of non-suitable experiments are AM detection tasks using pure tones and speech intelligibility tests in quiet.

2.2.2 The selected experiment: fastACI implementation

We chose to replicate the tone-in-noise experiment by Ahumada *et al.* [2], which is coded in fastACI as 'toneinnoise_ahumada1975.' The experiment considers the detection of a 100-ms long pure tone with a carrier frequency of 500 Hz, temporally centred in 500-ms long broadband Gaussian noises, that have an effective bandwidth up to 5000 Hz, half the sampling frequency $f_s = 10$ kHz. Our implementation differed in the presentation level of the background noises, which were set to a lower (more comfortable) level of 70 dB SPL (instead of 90 dB SPL). Additionally, we implemented an adaptive track procedure (tar-

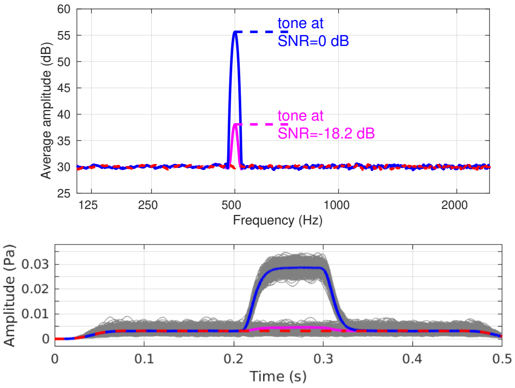

Figure 2: Characterisation of tone-in-noise (in blue) and noise-alone trials (in red) showing their average power spectrum (top) and the low-pass filtered Hilbert envelope (fcut=30 Hz) (bottom). In this figure, the signal-to-noise ratio SNR was fixed at SNR=0 dB (in blue) and -18.2 dB (in pink). The grey traces represent the individual envelopes of 1000 waveforms (only for SNR=0 dB). The target tone was centred at 500 Hz, which was temporally centred in the noises (between $t = 0.2$ and 0.3 s).

get score of 70.7%) with a variable tone level, expressed as a signal-to-noise ratio (SNR), whereas Ahumada and colleagues implemented a constant stimulus paradigm, with the tone presented at a fixed level. Figure 2 depicts the fast Fourier transform and the Hilbert envelope of several tone-in-noise and noise alone trials.

2.2.3 The selected experiment: Reference data

The interest of Ahumada *et al.* (1975) in investigating tone-in-noise paradigms relied, on the one hand, on the popularity of these paradigms at that time, and on the other, on the fact that the most popular trial configuration used tone and noise stimuli that were simultaneously gated (e.g., [1] and, more recently, [14]) in contrast to later preferred configurations in psychoacoustic research where the tone is shorter than the noise [2, 15]. For simultaneously gated noise data, Ahumada and colleagues had observed that the “tone present” and “tone absent” judgements of their participants did not only depend on the current trials but also on the noise information from the immediately previous trials. Thus, they wanted to investigate whether this noise effect could occur at a trial level if the tone, when presented, was made shorter than the noise.

The reference data for the current study can be found in [2] (their Fig. 1) and are replotted in our Fig. 3. We arbitrarily took the data from their first two participants. More particularly, we show the regression coefficients c_{ij} that related the trial information with the participant’s responses, comparable to the ACIs that can be obtained using revcorr (see Fig. 1). Ahumada *et al.*’s results can be summarised as follows: for five frequency bands at and around 500 Hz, there was a dominant positive weight of the information in the segment starting at 200 ms (containing the tone) that led participants to answer “tone present.” There were, however, some weaker (positive and negative) cues in the neighbouring bands. Some of these behavioural cues started before the tone onset, as visible in the on-frequency channel for participant RM (negative cue in the 0-ms segment). For participant KL, there was a consistent negative cue in the 200-ms segment for off-frequency bands (at 450, 550, and 600 Hz). We will come back to these data in the discussion section.

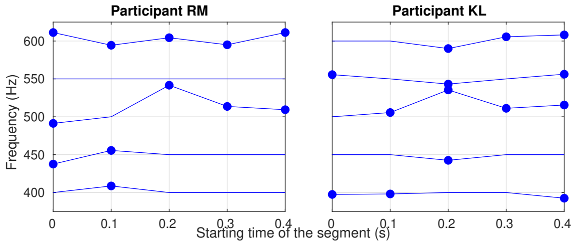

Figure 3: Reference data for two participants, taken from the “all-trial” analysis from [2], their Fig. 1. Non-null weights are indicated by filled markers. See the text for more details.

2.3 Artificial listeners

To evaluate the selected tone-in-noise experiment, the artificial listeners need to be composed of an auditory front-end module, sometimes referred to as a preprocessing model, and a back-end module that provides a simple binary decision, where the incoming sound is labelled as the most likely target interval from a limited set of options, based on signal detection theory. This study shows how to integrate a “preprocessing model” that transforms the input waveforms into an internal representation in fastACI, where two decision back ends are included. The preprocessing models, referred from now on to as “the models,” follow the structure of monaural auditory models in the AMT toolbox [12] and the decision back ends are, at this point, available within the script aci_detect.m from fastACI. We focus on the considerations to enable “any model” to be used with one of the decision back ends.

2.3.1 The models

The auditory front-ends that can be evaluated within the toolbox are required to have input and output parameters in the format as used in the monaural models from the AMT toolbox (as of version 1.0) [16]. In this format, the first two inputs are fixed and correspond to the sound waveform insig and the sampling frequency fs in Hz, respectively. The first output, outsig, contains the internal representation of the sound, which can either be a numerical matrix with dimensions time, frequency, and (optionally) modulation frequency or a cell array with as many elements as frequencies, containing numerical arrays with dimensions time and modulation frequency. The specific format of outsig is only relevant for interpretation, as the internal representation is always converted to a large one-column numerical array before is further processed (and compared between sounds) in the toolbox.

The toolbox has already been used using the AMT models king2019 [3], osses2021 [8], and a model which is not in AMT, but that follows the AMT conventions, osses2022a [5]. In this contribution we will use the model king2019 as a front-end module and the deci-

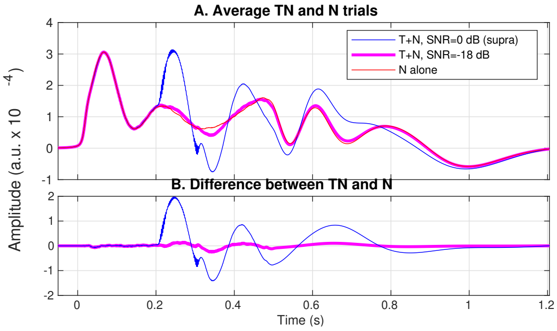

Figure 4: Model internal representations for tone-in-noise (TN) trials at SNRs of 0 dB (in blue) and -18.2 dB (in pink) and noise-alone trials (N, in red) for the band centred at 500 Hz. For this figure, the representations of all modulation filters were added together (see the text for details). Panel A shows average representations from 1000 TN or N trials. Panel B shows the difference between TN and N trials at the corresponding SNR. The blue curve, at SNR=0 dB, represents the difference signal used in the template, whereas the pink curve represents the difference signal at a much lower presentation level of the target tone, in the range of the experimental SNRs, near the detection threshold.

sion option optimal_detector as a back-end module, because we have never used this combination of front-end and back-end modules before. In this sense, king2019 is a new model. The model king2019 is a simplified modulation-filter-bank implementation, described in detail in [11, 16]. For the simulations here, the model "internal" representations had three dimensions, time, frequency, and modulation frequency. Average internal representations obtained from the model for tone-in-noise trials at an SNR of 0 and -18.2 dB, as well as for noise-alone trials are shown in Fig. 4A.

2.3.2 *The decision back ends*

The back-end module used in this study (optimal_detector) corresponds to the template-matching approach defined in [6]. However, because the listening experiments that we are simulating in the toolbox are single-interval trials, we need to compare the current trial with two templates and, hence, the extended decision framework based on two templates from [17] is used instead. For this detector, there are a number of compulsory parameters that need to be set. In particular, the templates need to be derived at a supra-threshold level, that is, at a signal level that is expected to be clearly detectable (variable sim_def.det_lev). At that supra-threshold level, a configurable number of tone-in-noise and noise-alone trials are averaged (variable sim_def.templ_num).

2.4 "Behavioural data:" Running the models

In the toolbox, the simulations are run as any other regular participant but indicating the name of the model (here: 'king2019') as the subject's ID. If the model is correctly configured in the toolbox (in king2019_cfg.m), the subject's ID will be recognised as being a model and then the simulations will automatically start being collected.

Listing 1: MATLAB commands required to run experiment toneinnoise_ahumada1975.

```
1 exp_name = 'toneinnoise_ahumada1975';
2 subj_ID = 'king2019'; % model front-end
3 cond = 'white'; % default noise condition
4 fastACI_experiment(exp_name,subj_ID,cond);
```

The trial progression will be visible in the MATLAB command window. Line 4 in Listing 1 will run for one experimental block of about 400 trials. If the total number of experimental trials is 3200, then that line needs to be run 8 times. The number of trials is a configuration parameter in the *_set.m file of the experiment (here, cfg.N_presentation per target to 1600 in toneinnoise_ahumada1975_set.m). The result file is a MAT file with the prefix savegame. The savegame file contains all the information collected for the corresponding participant (here 'king2019') during the experimental sessions.

2.5 Data post-processing: Reverse correlation

The post-processing is done following the default configurations of the toolbox, where the incoming sounds are transformed into time-frequency representations using a gammatone filterbank with 32 bands between 80 and 1500 Hz, and where the time samples are grouped in 10-ms long bins. Although there are many possible configuration options for the revcorr module, in this paper we adopted the revcorr settings detailed in [10].

This post-processing requires the result "savegame" file as input, from where the exact noises used in each trial are read together with the corresponding answer from the artificial listener. More specifically, the noises are first read from disk and then converted into the described gammatone T-F representations. The T-F noise representations and the artificial listener's answers are used as input to a generalised linear model (GLM) fitting, resulting in an ACI (more details can be found in [4, 5]).

3. ADDING A NEW ARTIFICIAL LISTENER

If the selected auditory preprocessing module complies with the input-output structure described in Sec. 2.3.1. The only remaining aspect is to manually generate a configuration file for the model. For king2019.m the configuration file needs to be named king2019_cfg.m (Listing 2) and is required to be located in the toolbox folder Local. The configuration file contains specific parameters for the decision back end and, if needed (but not required), contains extra parameters for running the front-end model. Some of the compulsory fields in the configuration file are:

Listing 2: File king2019_cfg.m.

```
% Parameters for the model (1 of 2)
def_sim.modelname='king2019';
% Parameters for the detector:
def_sim.decision_script='aci_detect';
def_sim.template_script='model_template';
def_sim.template_every_trial=0;
def_sim.templ_num=100;
def_sim.det_lev=0; % supra lvl, 0 dB SNR
```

The exact configuration file used in this contribution was added to the stored configurations in fastACI (file name: osses2023b_FA_king2019.m in ../Simulations/Stored_cfg/). This configuration file is automatically copied to the local folder if the script publ_osses2023b_FA_1_sim.m is run. After successfully setting the configuration file, the simulations can be run as indicated in Listing 1.

4. RESULTS AND DISCUSSION

4.1 Internal representations from the model

The shape of the internal representations of the test stimuli, here the noise-alone and tone-in-noise intervals, depend on the auditory model and are assumed (by the modeller) to be relevant enough to investigate experimental outcomes. In other words, different models can result in very different internal representations (see, e.g., [16]). Average internal representations for noise-alone and tone-in-noise trials are shown in Fig. 4A, at SNRs of 0 dB and -18.2 dB. Each curve was obtained by averaging 1000 trials. For this illustrative representations, only the outputs of the gammatone filter bank centred at 500 Hz are shown. For this "audio band," all modulation bands were grouped together (sum across modulation bands). In particular, the noise-alone average is shown as red traces, and the tone-in-noise averages at SNRs of 0 and -18.2 dB are shown as blue and pink traces, respectively. Figure 4B, shows the difference between tone-in-noise and noise-alone trials as a schematic representation of the trial-by-trial difference that, after normalisation, is effectively used by the model for the template-matching decision. The SNRs used in the figure were strategically chosen to represent (1) the arbitrary supra-threshold level, i.e., the condition at which the tone is presented at an "easy-to-detect" level (SNR=0 dB, blue traces in Fig. 4), and the (2) tone level at around the model detection threshold (SNR=-18.2 dB, pink traces), as presented in the subsequent section. In other words, because the model decision used the template-matching approach, the correlation between the blue and pink traces in Fig. 4B lead to a correct tone detection in 70.7% (the target score) of the times.

4.2 Behavioural results

The raw data were obtained by running the simulations with the king2019 model (Sec. 2.4, Listing 1), whose main file (savegame_2023_04_19_01_08_king2019_toneinnoise_ahumada1975_white.mat) can be retrieved from [18]. From this savegame file, the experimental variable (expvar), here the SNR expressing the tone level with respect to the background noises, can be retrieved and plotted as an average per session or as an average as a function of SNR, with scores below or above 70.7% for SNRs to the left (lower) or to the right (higher) than the model detection threshold as shown in Fig. 5. The model, king2019, produced a score of 70.7% at an SNR=-18.2 dB, which is indicated as a vertical dashed line in the figure. Because of the adaptive procedure, most of the experimental trials lied in the SNR bin around this threshold (1166 trials of the 3200).

4.3 Auditory classification images

As indicated in Sec. 2.5, the ACIs were obtained by applying a GLM fitting algorithm set up as detailed in [5]. This process required as input the savegame file with model responses and the automatically-retrieved sound waveforms used during the simulations.

The derived ACI fitted to the specific noise waveforms and the corresponding artificial listeners' responses are shown in Fig. 6. The obtained ACI shows a dominant positive weight (maroon region) at around 500 Hz, between 0.2 and 0.3 s, which is the time at which the tone is expected to be presented. In other words, the artificial listener was using the on-frequency information during the tone presentation to respond "tone present." This intuitive

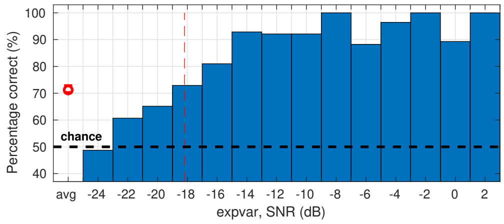

Figure 5: Behavioural results obtained from the simulations using the king2019 model. The results were obtained from the experimental variable, the tone level expressed as a SNR. The overall score, averaged across 400-trial blocks is indicated by the red circle marker, whereas scores as a function of 2-dB wide SNR bins are depicted by the continuous blue line. The vertical dashed line indicates the location of the average SNR threshold of -18.2 dB. Refer to the text for more details.

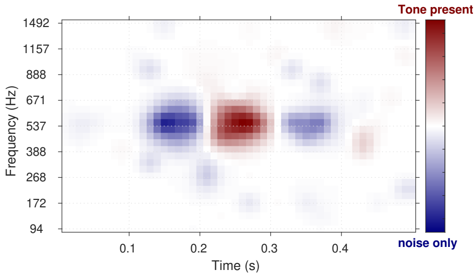

Figure 6: ACI obtained for the responses from the artificial listener ('king2019'). Dominant positive (maroon) cues were found between 0.2 and 0.3 s, simultaneous to the tone presentation, whereas negative cues (blue) precede and follow the tone onset and offset, respectively.

result is in line with the main positive cues in the reference data (Sec. 2.2.3 and Fig. 3). We also found negative cues (blue regions) before and after the tone in the same on-frequency region. Taking the difference internal representations from Fig. 4B that, after normalisation, are used for the template in the template-matching process, the negative cue after the tone offset can be directly attributed to the negative amplitudes between 0.3 and 0.4 s. In that time segment, the tone-in-noise representations show amplitudes that are below those of the noise-alone representations (Fig. 4A). This is a consequence of the undershoot effect related to the auditory adaptation processing in king2019 which, in a simplified form with respect to other auditory models [16], is implemented as a 3-Hz high-pass filter [11, 16]. On the contrary, the cue before the tone onset in the ACI, between 0.1 and 0.2 s, highlights the fact that noise energy preceding the tone can interfere with the detection process, a finding that is not immediately apparent from Fig. 4B, as the difference between tone-in-noise and noise-alone trials seems to be negligible between 0 and 0.2 s. This observation is consistent with the view that "behavioural templates"—as expressed by ACIs—should be seen as descriptors of data structure rather than estimates of processing components [19].

In general, for the evaluated tone-in-noise experiment, we can state that the ACI results obtained for the artificial listener share an overall resemblance with the experimental results from the reference tone-in-noise data [2], having the same main positive cue during the time segment of the expected tone. Further, our results support an influence of the noise during time segments directly before the tone onset and directly after the tone offset. Ahumada and colleagues hypothesised that this influence was possible for tone-in-noise tasks with tones embedded and temporally centred in noise, in contrast to the tasks using the simultaneous gating noise paradigm. Importantly for the current contribution, these observations were derived from an artificial listener mimicking a trial-by-trial detection task, where the artificial listener was based on a simplified modulation-filter-bank model (king2019) adopting a template matching, a combination of front-end and decision modules that we tested within the fastACI toolbox for the first time.

5. CONCLUSION

In this contribution we provided a step-by-step guide to integrate an auditory model, king2019, as an artificial listener within the fastACI toolbox to which a decision device, a template-matching approach, was attached. We used the newly configured artificial listener to mimic human performance for a tone-in-noise paradigm, where the tone was a 100-ms long sinusoid centred at 500 Hz, which was added to 500-ms long broadband Gaussian noises in half of the trials [2]. After applying reverse correlation to the artificial listener' responses we obtained an auditory classification image (ACI) with weights that were broadly similar to the weights obtained by Ahumada and colleagues. At this point, it is important to emphasise that our effort in this study was not on a critical comparison of simulation results with the original behavioural data but rather on how to build an artificial listener from an auditory model attached to a decision device, all in the framework of the fastACI toolbox. We are confident of the flexibility of the toolbox to be used with any other type of model, as soon as the model is adjusted to have a specific format of inputs and outputs (see Sec. 2.4). New psychoacoustic experiments can also be added in a straightforward way, however, a current limitation of the toolbox is that we have evaluated it only using single-interval tasks, e.g., speech-in-noise [4] and amplitude modulation detection [3]. The toolbox is in constant development and, as part of our future work, we plan to extend the toolbox to support tasks using trials with two or more intervals. The challenge of such an extension is not at the level of the experiment implementation and subsequent data collection, but on the extension of the post-processing module to apply the reverse correlation paradigm.

6. DATA AVAILABILITY

The fastACI toolbox for MATLAB, containing all the required scripts to replicate the results from this study can be downloaded from `https://github.com/aosses-tue/fastACI`. The only exception is the auditory model king2019, which is available in the AMT toolbox [12] (routine king2019.m). All figures can be replicated using the fastACI script publ_osses2023b_FA_figs.m. For an exact reproduction of the study figures, the stimuli and raw data need to be retrieved from Zenodo [18], before running the indicated fastACI script.

7. ACKNOWLEDGEMENTS

This study was funded by the ANR grants “fastACI” (Grant No. ANR-20-CE28-0004) and “FrontCog” (Grant No. ANR-17-EURE-0017).

8. REFERENCES

[1] A. Ahumada and J. Lovell, “Stimulus features in signal detection,” *J. Acoust. Soc. Am.*, vol. 49, pp. 1751–1756, 1971.

[2] A. Ahumada, R. Marken, and A. Sandusky, “Time and frequency analyses of auditory signal detection,” *J. Acoust. Soc. Am.*, vol. 57, pp. 385–390, 1975.

[3] L. Varnet and C. Lorenzi, “Probing temporal modulation detection in white noise using intrinsic envelope fluctuations: A reverse correlation study,” *J. Acoust. Soc. Am.*, vol. 151, pp. 1356–1366, 2022.

[4] L. Varnet, K. Knoblauch, F. Meunier, and M. Hoen, “Using auditory classification images for the identification of fine acoustic cues used in speech perception,” *Front. Hum. Neurosci.*, vol. 7, pp. 1–12, 2013.

[5] A. Osses and L. Varnet, “A microscopic investigation of the effect of random envelope fluctuations on phoneme-in-noise perception,” *BioRxiv*, pp. 1–24, 2022.

[6] T. Dau, B. Kollmeier, and A. Kohlrausch, “Modeling auditory processing of amplitude modulation. I. Detection and masking with narrow-band carriers.,” *J. Acoust. Soc. Am.*, vol. 102, pp. 2892–2905, 1997.

[7] S. Ewert, “AFC - A modular framework for running psychoacoustic experiments and computational perception models,” in *Proceedings of the International Conference on Acoustics AIA-DAGA*, pp. 1326–1329, 2013.

[8] A. Osses and L. Varnet, “Consonant-in-noise discrimination using an auditory model with different speech-based decision devices,” in *DAGA*, pp. 298–301, 2021.

[9] A. Osses and L. Varnet, “fastACI toolbox: the MATLAB toolbox for investigating auditory perception using reverse correlation (v1.3),” 2023. Zenodo. doi:`10.5281/zenodo.7888588`.

[10] A. Osses, C. Lorenzi, and L. Varnet, “Assessment of individual listening strategies in amplitude-modulation detection and phoneme categorisation tasks,” in *International Congress on Acoustics*, (Gyeongju, South Korea), pp. 1–12, 2022.

[11] A. King, L. Varnet, and C. Lorenzi, “Accounting for masking of frequency modulation by amplitude modulation with the modulation filter-bank concept,” *J. Acoust. Soc. Am.*, vol. 145, pp. 2277–2293, 2019.

[12] P. Majdak, C. Hollomey, and R. Baumgartner, “AMT 1.x: A toolbox for reproducible research in auditory modeling,” *Acta Acustica*, vol. 6, p. 19, 2022.

[13] R. Murray, “Classification images: A review,” *J. Vis.*, vol. 11, pp. 1–25, 2011.

[14] J. Mao, A. Vosoughi, and L. Carney, “Predictions of diotic tone-in-noise detection based on a nonlinear optimal combination of energy, envelope, and fine-structure cues,” *J. Acoust. Soc. Am.*, vol. 134, pp. 396–406, 2013.

[15] T. Dau, D. Püschel, and A. Kohlrausch, “A quantitative model of the effective signal processing in the auditory system. II. Simulations and measurements,” *J. Acoust. Soc. Am.*, vol. 99, no. 6, pp. 3623–3631, 1996.

[16] A. Osses, L. Varnet, L. Carney, T. Dau, I. Bruce, S. Verhulst, and P. Majdak, “A comparative study of eight human auditory models of monaural processing,” vol. 6, p. 17, 2022.

[17] A. Osses and A. Kohlrausch, “Perceptual similarity between piano notes: Simulations with a template-based perception model,” *J. Acoust. Soc. Am.*, vol. 149, pp. 3534–3552, 2021.

[18] A. Osses and L. Varnet, “Raw and post-processing data for using auditory models to mimic human listeners in reverse correlation experiments from the fastACI toolbox,” 2023. Zenodo. doi:`10.5281/zenodo.7886232`.

[19] P. Neri, “Classification images as descriptive statistics,” *J. Math. Psychol.*, vol. 82, pp. 26–37, 2018.