May 2020

UR EAR Version 2020b - User Manual

(Edited from UR EAR Version 2.2 and 2020a User Manual by Afagh Farhadi & Laurel Carney)

A guide to our software tool: “University of Rochester: Envisioning Auditory Responses” (UR_EAR Version 2020b). For modeling, creating stimuli, and more. This modeling tool is an upgrade from UR_EAR Versions 2020a and 2.2.

Table of Contents

1. Exploring the UR_EAR Package
    a. Files and contents of package
    b. Stimulus functions
2. Running and Using the GUI
    a. Running and Using the GUI - the basics
    b. The Built-in AN and IC Models:
    c. The Built-in Stimuli
    d. Model Plots
3. Adding Stimuli
4. Adding AN or IC Models
5. “Quick Plot” of responses for one CF.
References
Contact Us

Chapter 1 - Exploring the UR_EAR Package

This MATLAB code package is designed to run auditory-nerve and midbrain models with different stimuli.

The package contains numerous files and folders, including:

UR_EAR_2020b.m   The main m-file for the GUI code.

A folder containing Stimulus functions (see below)

Complied (Matlab “mex” files) for the Zilany et al (2014) model:
    model_IHC (mex file versions for Windows, macOS, and GNU/Linux)
    model_Synapse (Win/Mac/Linux mex files)

Complied (Matlab “mex” files) for the Zilany et al (2014) model:
    Model_IHC_BEZ2018 (Win/Mac/Linux mex files)
    model_Synapse_BEZ2018 (Win/Mac/Linux mex files)

Several files that are used by both AN models:
    ffGn.m

fitaudiogram2.m
generate_neurogram_UREAR2.m
generateANpopulation.m

Files associated with the IC Models:

SFIE_BE_BS_BMF.m
get_alpha_norm.m
unitgain_bpFilter.m

And Documentation:

manual_UR_EAR_2020b.pdf
readme.txt

These files and folders are all necessary to make the modeling application run properly. The files the user is most likely to use overtly are the stimulus functions, and the main UR_EAR_2020b m-file.

Stimulus functions

Each stimulus type is created by an m-file containing a function that is called by UR_EAR. The parameters are described and assigned to specific variable names to be able to interact with the rest of the UR_EAR code. The most general form of a stimulus, with only the overall sound level as a variable, is an audio file. For more information on how to add your own stimulus along with controllable parameters, see the Creating Your Own Stimulus section in Chapter 3.

In order not to interfere with identically-named functions already in use, the stimulus functions are placed in a MATLAB package folder called +stimuli. Do not put this folder on the MATLAB path. If you want to use any of these functions in your own code outside of UR_EAR, you must obey the rules for using a package. First, import the stimulus package, and then use TIN normally,

```
import stimuli.*
waveform = TIN(...);
```

The command to “import” the package is included in the UR_EAR code.

[Another method is to precede the function name with the package name (not including the + symbol) and a period. For example, to run the stimulus function, TIN.m, your code would be

```
waveform = stimuli.TIN(...);
```

Consult the MATLAB documentation for “`import`” for more details. Note that this method is used in one place in UR_EAR, before the package has been imported. See below.]

The stimulus m-files themselves are ordinary MATLAB functions; they are simply accessed a bit differently until the stimulus package has been imported. If you happen to have a package

folder called “+stimuli” already, you should change the name of the package folder in UR_EAR by renaming the folder and then changing the UR_EAR code to match.
The package name (“stimuli”) appears twice in the UR_EAR code:
once as “`stimuli.TIN`” and once as “`import stimuli.*`”.

Chapter 2 - Running and Using the GUI

The Basics:

Once you have downloaded the package, open up MATLAB and either add the UR_EAR_2020b folder to your MATLAB path (preferably at the end of the path), or change your current folder to the UR_EAR_2020b folder. Then run UR_EAR_2020b in the usual way by typing “`UR_EAR_2020b`” at the command line or opening it in the editor and pressing the editor’s “Run” button. This GUI allows you to choose from a variety of models, stimuli, and parameter values, then see the resulting plots.

The first time you run the program, a window containing helpful hints is displayed. You can tell UR_EAR not to open this automatically at startup or open it on demand by pressing the “Tips…” button near the upper right-hand corner of the UR_EAR window.

When it starts up, UR_EAR will check periodically to see if a newer version is available. You can control the interval (30 days, by default) or turn off this feature entirely from the dialog box obtained by pressing the “Updates…” button, located above the Tips button. The version check requires Internet access. If a new version if found, the Carney Lab web site will be opened in your system browser where you will find a link to the latest version of UR_EAR.

Stimulus Selection and Stimulus Parameters:

At the top left is a pop-up menu where you can select which stimulus you’d like to use. Once a stimulus is selected, the “Parameters” section below it will change based on your selection.

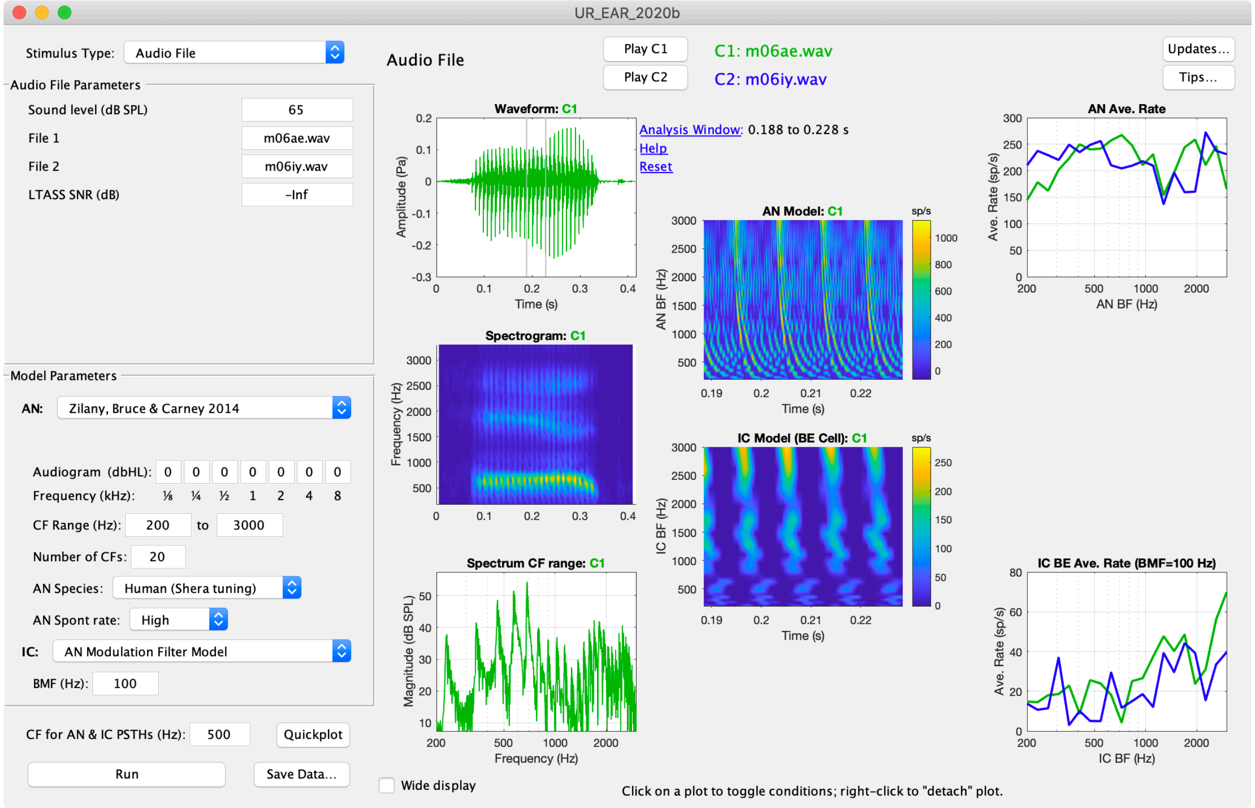
The Parameters section is where you change the values of the parameters specific to the chosen stimulus. You can change the values of duration, frequencies of spectral edges or bandwidths, choose between various comparisons, etc. For some stimuli, there is an automatic comparison between two versions of the same type of stimulus, for example, comparing noise-plus-tone to the same noise without a tone. These plots are made at the same time, and plotted on the same chart. For some comparisons you will have to toggle between the different plots once you press “Run” to run the model. To compare various parameter changes that are not included automatically, simply run the model twice with those different selections. Due to its ease and speed, exploring the parameter space is easy.

Model Parameters:

Below the stimulus parameters is the “Model Parameters” section, where you select which auditory-nerve (AN) model and inferior colliculus (IC) model you’d like to use from the pop-up menus and specify the model parameters. Parameter selection will slightly depend on which model you have selected, but generally there will be options such as auditory-nerve center-frequency range, number of AN fibers (evenly log spaced across the designated frequency range) and hearing status (by audiogram). In UR_EAR_2020b there are two different AN models and two different IC models from which to choose. The plots in the center column show a small portion (40 ms by default, but this can be adjusted) of each response. You also have the option to examine the entire response using the “wide display” option (see check box at bottom of GUI). Another feature is the possibility of choosing the number of fibers at each CF for the Bruce et al. AN model; this feature allows averaging across a set of independent AN simulations at each CF.

Press “Run”:

Once you have specified the desired values, press the “Run” button to run the model. If running correctly, you should see a progress bar below the Run button. Otherwise, you will be able to see in the command window any errors that occurred. (Note: While running, the Run button is disabled. If there is an error preventing UR_EAR from completing, the Run button will remain



Stimulus Type: Audio File

Sound level (dB SPL): 65

File 1: m06ae.wav

File 2: m06iy.wav

LTASS SNR (dB): -Inf

C1: m06ae.wav

C2: m06iy.wav

Analysis Window: 0.188 to 0.228 s

AN: Zilany, Bruce & Carney 2014

Audiogram (dbHL):	0	0	0	0	0	0	0
Frequency (kHz):	⅛	¼	½	1	2	4	8

CF Range (Hz): 200 to 3000

Number of CFs: 20

AN Species: Human (Shera tuning)

AN Spont rate: High

IC: AN Modulation Filter Model

BMF (Hz): 100

CF for AN & IC PSTHs (Hz): 500

☐ Wide display

Click on a plot to toggle conditions; right-click to "detach" plot.

Figure 1 – Screen shot of UR_EAR GUI, with responses to two (default) VCV waveforms.

disabled. It can be re-enabled simply by selecting a stimulus type, usually the same one already selected.) After the model has finished running, the area of the figure to the right will show various plots, toggling capabilities, and ‘play’ buttons to hear the stimuli you selected.

The Built-In AN and IC Models:

This software includes two AN models and two IC models. The purpose of this code is to explore a variety of stimuli with various models, and having these built-in models contributes to that goal. These models can be used with both built-in stimuli and novel stimuli added by users. The built-in AN models are Zilany, Bruce, & Carney 2014 Auditory Nerve (AN) Model and Bruce, Erfani, & Zilany 2018 Auditory Nerve (AN) Model. The two IC models are the Same-Frequency Inhibition and Excitation Model (Nelson & Carney 2004), and the simpler Inferior Colliculus (IC) Modulation Filter Model (Mao et al. 2013). All of these models have various parameters that can be easily adjusted in our software. See Chapter 3 for more information on the sections of UR_EAR which run the models, plot the responses, and interact with parameter selection.

The Built-In Stimuli:

This software includes many built-in stimuli ready to interact with the built-in models. The stimuli are complex sounds based on psychophysical studies and built specifically for use with the built-in models. Our models focus on the inferior colliculus, and so the complex sound stimuli included were also intended for studying the inferior colliculus. Following is a list of default stimuli and their descriptions:

Audio file:

We have included an audio file option that allows any type of audio file to be used as a stimulus. This is one of the simplest ways to interact with the GUI; the program will read in any audio file and give the response. We have included two example files as the default: vowel-consonant-vowels, ‘m06ae.wav’ (“had”) and ‘m06iy.wav’ (“heed”) (from the Hillenbrand et al. 1995 database). You can set the overall sound level in dB SPL. Leaving the sound level field blank will cause the program not to scale the raw data from the file in any way, in case it has already been scaled to pascals. (MATLAB will normally read an audio file into values between −1 and 1. A sinusoid with a peak amplitude of 1 Pa represents a sound level of 97 dB SPL.) If noise is desired, you can set the SNR for LTASS noise (Long Term Average Speech Spectrum). If the sound level is blank, the noise field becomes noise level in db SPL instead, because SNR cannot be calculated.

Noise Band:

Gaussian noise stimulus, with a band specified by selecting the low and high frequency limits. This stimulus is useful for exploring *Edge Pitch,* based on Klein and Hartmann (1981). A pitch is perceived at about 4% inside the noise band from a sharp spectral edge. The IC population response near the ‘edge’ is intriguing… (compare it to the response to a tone plus noise.)

Notched Noise:

Notched noise is a commonly used masking stimulus for psychophysical experiments, such as in Patterson, 1976. The notch is centered at the signal frequency, and is used to derive shapes of critical bands. See Maxwell et al., 2020, JASA (in press) for a study of this stimulus.

Tone in Noise:

Another popular research topic in auditory processing, tone-in-noise paradigms involve discerning the difference between a stimulus of just noise, and of the same noise with an added tone. The noise frequency range can be specified by typing two frequencies, or a fraction of an octave by typing a fraction with a slash (/). For example, “1/3” is interpreted as ⅓-octave noise centered on the specified tone frequency. If you want 1-octave noise, type “1/1”.

Profile Analysis:

This stimulus is a waveform constructed of evenly log-spaced components centered around 1,000 Hz. This stimulus can be presented with and without an increment of the center frequency component. See Green (1983) for details.

Pinna Cues:

Filters band-limited noises using an artificial notch, modeling head-related transfer functions that simulates a simple spectral cue for sound localization.

SAM Tone (or pure tone):

Creates an AM tone with specified carrier and modulation frequency and depth. Set modulation depth to -Inf (−∞) dB for 0% modulation (i.e., a pure tone), 0 dB for 100% modulation, or any other value.

Complex Tone / Harmonic Complex:

Specify the pitch and range of harmonics. Specify whether or not to include the fundamental.

Single Formant:

Simple triangular spectral peak, similar to those seen in vowels, à la Lyzenga & Horst 1995.

Double Formant:

Two formant peaks, generated using Klatt resonators. Similar to stimuli in Lyzenga & Horst 1997, 1998.

Schroeder Phase Harmonic Complex:

Set F0 and number of components. Positive and Negative C values are compared.

Noise in Notched Noise:

A band of noise (target noise) is added to a complementary band-reject noise (masker noise) which is 10 dB higher in spectrum level. In test stimulus the target noise spectrum level is increased and in the standard stimulus the target noise level is fixed. Discrimination between these two stimuli was studied in Viemeister 1983.

FM tone:

Two stimuli both Sinusoidal frequency modulated Tone with two different modulation frequency and modulation excursion that you can adjust in the GUI.

Forward Masking:

First condition is a sinusoid masker with a short delay before the probe tone, and second condition is just the masker. You can change properties of both masker and tone in stimulus parameters in GUI.

CMR-BW:

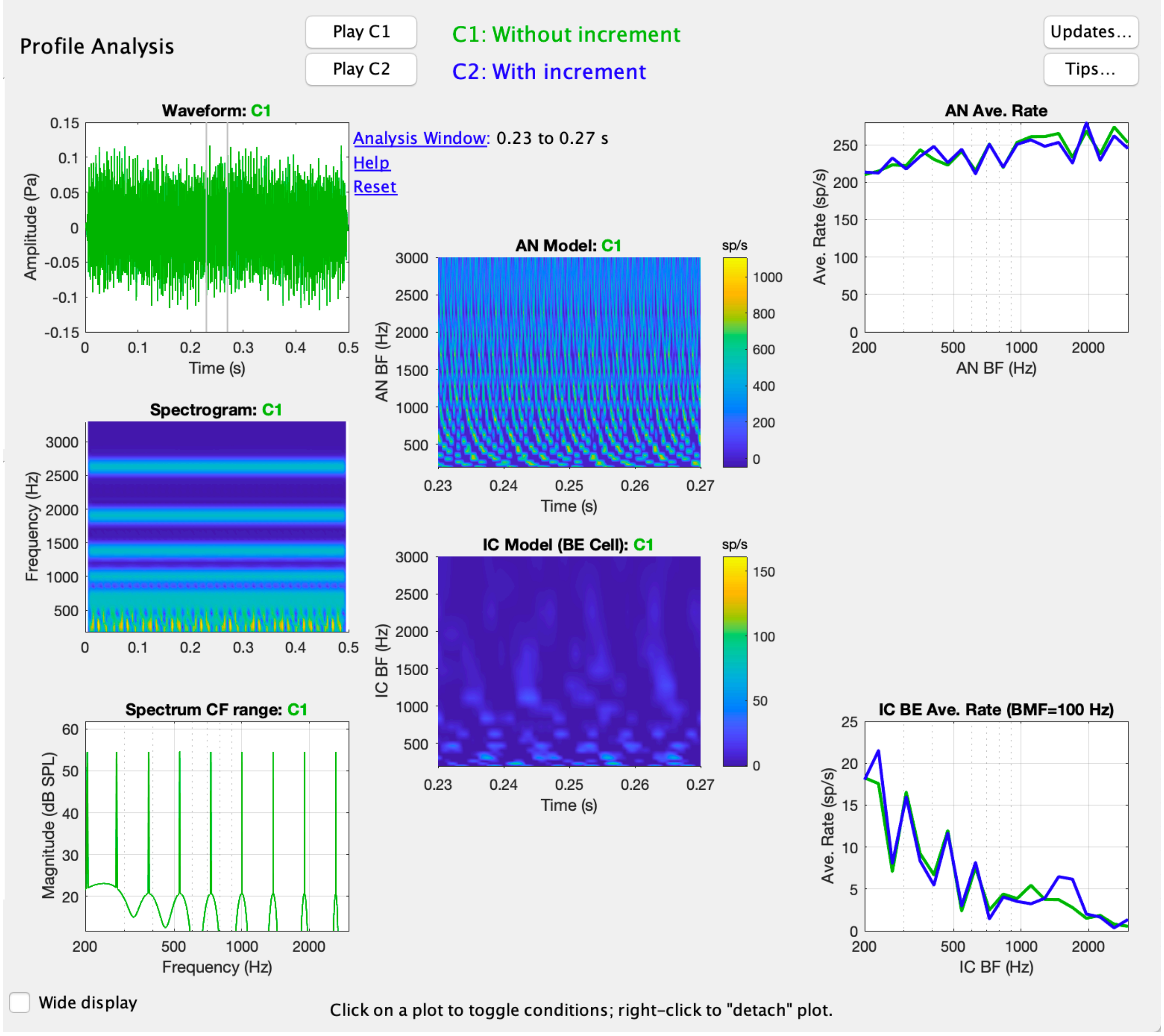
In Comodulation Masking Release (CMR) the Gaussian masker noise is modulated with an envelope created by a low-pass filtered noise (shifted up by a constant). Modulation results in lower detection threshold as bandwidth (BW) of the masker noise increases. Two stimuli are unmodulated and modulated Gaussian masker noise with tone. You can change tone properties and the bandwidth of the masker in stimulus design. Similar to stimuli in Haggard, Hall & Grose 1990.

CMR-FB:

These stimuli consist of a tone with narrow noise band centered on the tone frequency added to a series of other narrow noise bands. In one of the stimuli the flanking noise bands are modulated with the same frequency as the central noise band, which is shown in plots as the comod (comodulated bands) and in the other stimulus they are modulated with independent noise bands, shown by codev (codeviant bands). Similar to stimuli in Hall & Grose 1994.

Model Plots (Right panel of UR_EAR)

Interacting with the plots is fairly simple. This plot generator showcases comparisons between stimuli with slight variations, such as noise-plus-tone compared to noise alone. Here is an example model plot response:


C1: Without increment

C2: With increment

Analysis Window: 0.23 to 0.27 s

☐ Wide display

Click on a plot to toggle conditions; right-click to "detach" plot.

Figure 2 – Model output panel.

The left-most column of plots in the panel is the stimulus details, to allow the user to check if the stimulus is what they expected. The top left plot is the stimulus waveform for the selected stimulus. The second plot in the leftmost column is the spectrogram of the stimulus. The bottom plot in the leftmost column is the spectrum over the selected characteristic frequency (CF) range. The top middle plot is the auditory-nerve population model response vs. time. The bottom middle plot is the inferior colliculus population model response vs. time. The top rightmost plot is the auditory-nerve average spike rate as a function of best frequency. The bottom rightmost plot is the inferior colliculus band-enhanced model average population discharge rate as a function of best frequency. If the SFIE model is run, the middle rightmost plot is the cochlear nucleus model response.

The legend at the top shows the various conditions run (1 or 2), with a “play” button located next to each condition's name. If you left-click on a plot, it will toggle between conditions. If you right-click on a plot, you can detach that one plot from the entire response figure, a useful tool for saving and editing individual plots. You also have the option of writing the data from that plot out to a structure variable in the workspace.

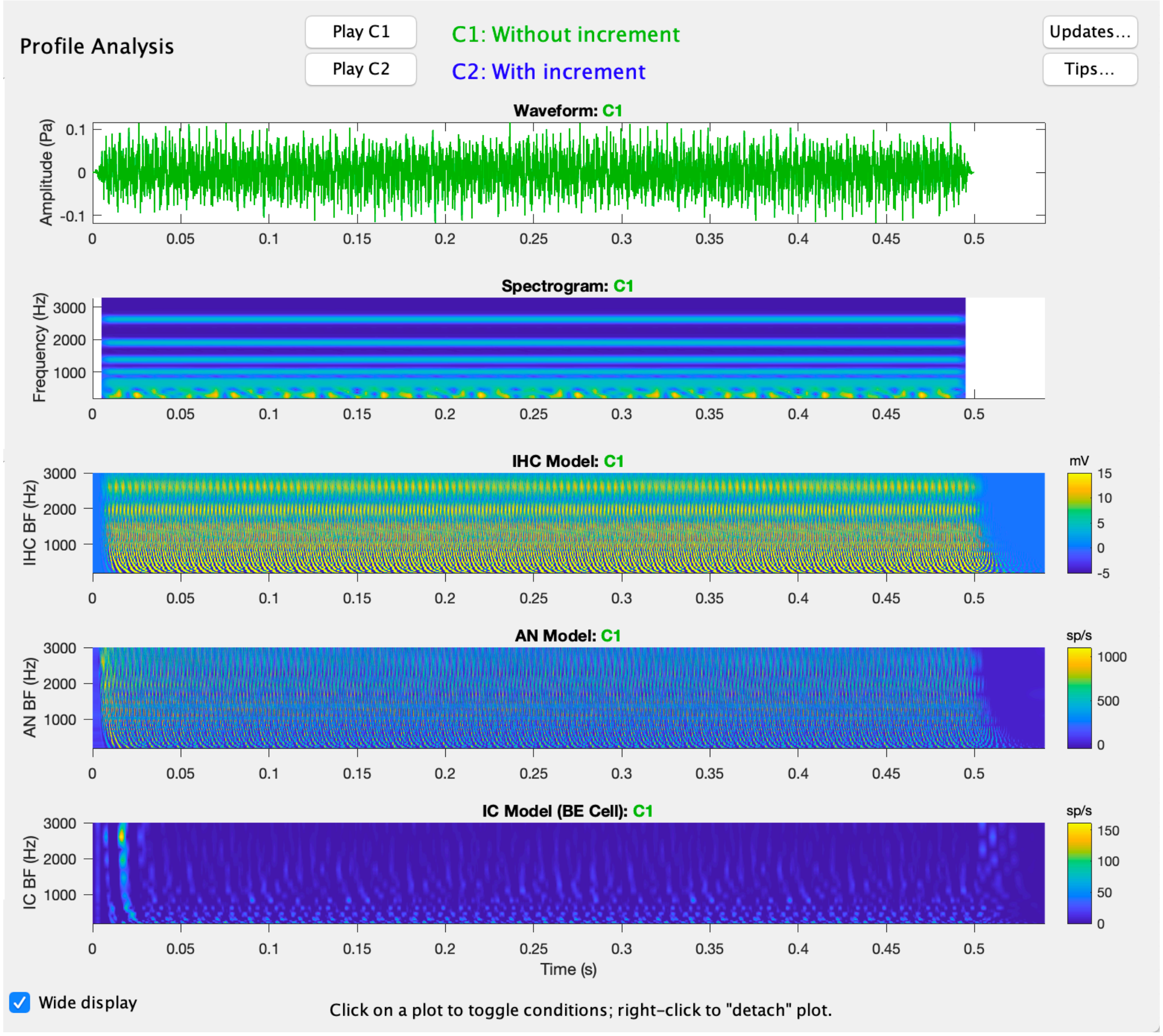
Each model you add will have to create the plots for this panel. If you add a model yourself, you can decide which plots you’d like to utilize, change functionality, and more.

In wide display mode the first two plots are stimuli waveform and spectrogram respectively. The third plot is the Inner hair cell population model response over time. The last two plots are auditory-nerve and inferior colliculus population model responses over time respectively. You can see an example of wide display responses in Figure 3.

Analysis Window

The AN model and IC model plots are shown for only a portion of the full duration. This time span is called the *analysis window* and can be adjusted by left-clicking and dragging one or both of the vertical gray lines appearing in the stimulus waveform plot. Right-click and drag to move both lines together, keeping the same span between them. You can also enter explicit time values into a dialog box by clicking on the analysis window display text at the upper right corner of the waveform plot. By default, the analysis window will be re-set to the default values each time you ‘Run” the model. Right-click on the “Analysis Window” text, and the word “HELD” appears (in Red Font, at the far right of the text box) When you see “HELD”, that indicates that the current analysis window will be held constant across model runs. Right-click again to turn off the ‘HELD” setting. Click on “Help” to see help information, or “Reset” to reset the analysis window back to the default (20 ms on either side of the center of the condition 1 stimulus). Helpful hint: if you accidentally move a gray line to a place where it is no longer visible, you can recover it by clicking on “Reset.”

The averages rates shown in the third column of model output plots are computed only for the time span of the analysis window, and you can see these plots change as you adjust the window.



Click on a plot to toggle conditions; right-click to "detach" plot.

Figure 3 – “Wide Display” mode (see check box on lower left of GUI). This display works well for speech materials, or any non-stationary stimulus that has an interesting time course.

Chapter 3 - Adding a New Stimulus

To add a new stimulus, you need to follow these three steps:

1. **Create your own function** for producing your stimulus.
2. **Edit UR_EAR_2020b.m** to add your stimulus. The code itself is fully commented, so read through the m-file for a better understanding of the flow of the code and how to add to it. The code is a programmatic GUI designed to be edited. There are multiple panels within the GUI, such as a parameter selection panel, and a model plots panel.

1. Creating your own stimulus function

You need to create a function in its own m-file to add your stimulus to UR_EAR. Put your file in the subfolder named “+stimuli”. UR_EAR will be able to find files placed here. Your function will get its inputs from parameters typed into the GUI. You can see the built-in stimulus functions as examples. Start with the most similar existing one! Your function must create the stimulus, convert the output to pascals, and ramp the stimulus on and off (in order to remove any artifacts resulting from abrupt starts or stops). The tukeywin function is a convenient way to generate a raised cosine ramp.

2. Edit UR_EAR Stimulus Section

Going through UR_EAR_2020b.m, search for lines with the term “STIM” in them to spot places that require editing.

Chapter 4 - Adding a New AN or IC Model

Introduction

To add a new model, you need to follow these two steps (similar to adding a new stimulus):
You will be working with UR_EAR_2020b.m. The code itself is fully commented, so read through the m-file for a better understanding of the flow of the code and how to add to it. The code is a programmatic GUI designed to be edited. There are multiple panels within the GUI, such as a parameter selection panel, and a model plots panel.

1. Adding Your Model Code

To add a model, you will need to add the code to the folder containing the UR_EAR folders and files. You will want to keep your model as a separate m-file. See the next two steps and the code itself for more details.

2. Adding to UR_EAR_2020b.m

Going through UR_EAR, you will need to do the following:

1) If your model is an AN model you need to add it to `ANmodelTypeOptions` cell array and for IC models you should add it to `ICmodelTypeOptions` array.
2) For an AN model, add a section below the line containing “`switch Which_AN`”. For an IC model, add your model code below the line containing “`switch Which_IC`”. Use the built-in model code as a guide.

Chapter 5 - Quick plot of Single CF Responses

To look at the time responses (time-varying firing rate) for a single CF, type the desired CF into the quick plot panel at the bottom of GUI and click on the quick plot button. The Quick Plot shows responses of one fiber in the model population that has a CF closest to the requested

CF. Very simple, and sometimes helpful for examining and/or illustrating the detailed responses.

Note that you can save all of the model response values that are displayed in the GUI, after a “run,” into a file by clicking “Save Data…” at the bottom left of the GUI. After you click, you will be prompted for a filename that will be used to save the data structure. Then you can run a stand-alone version of quick_model_plot.m, which reads in the data from any of your saved files and plots the AN and IC responses (this function is nice for making figures the code is simple; you can plot out any of the other model outputs with a little editing, change the format, etc.)

```
quick_model_plot('test_data_save.mat', CF)
```

References

Bruce, I. C., Erfani, Y., & Zilany, M. S. (2018). A phenomenological model of the synapse between the inner hair cell and auditory nerve: Implications of limited neurotransmitter release sites. Hearing research, 360, 40-54.

Krips, R., & Furst, M. (2009). Stochastic properties of coincidence-detector neural cells. Neural computation, 21(9), 2524-2553.

Carney, L. H., Li, T., & McDonough, J. M. (2015). Speech Coding in the Brain: Representation of Vowel Formants by Midbrain Neurons Tuned to Sound Fluctuations. ENeuro, 2(4). doi:10.1523/eneuro.0004-15.2015.

Mao, J., Vosoughi, A., & Carney, L. H. (2013). Predictions of diotic tone-in-noise detection based on a nonlinear optimal combination of energy, envelope, and fine-structure cues. The Journal of the Acoustical Society of America J. Acoust. Soc. Am., 134(1), 396. doi:10.1121/1.4807815.

Nelson, P. C., & Carney, L. H. (2004). A phenomenological model of peripheral and central neural responses to amplitude-modulated tones. The Journal of the Acoustical Society of America J. Acoust. Soc. Am., 116(4), 2173. doi:10.1121/1.1784442.

Zilany, M. S., Bruce, I. C., & Carney, L. H. (2014). Updated parameters and expanded simulation options for a model of the auditory periphery. The Journal of the Acoustical Society of America J. Acoust. Soc. Am., 135(1), 283-286. doi:10.1121/1.4837815.

Green, D. M. (1983). Further studies of auditory profile analysis. The Journal of the Acoustical Society of America J. Acoust. Soc. Am., 73(4), 1260. doi:10.1121/1.389274.

Klein, M. A. (1981). Binaural edge pitch. The Journal of the Acoustical Society of America J. Acoust. Soc. Am., 70(1), 51. doi:10.1121/1.386581.

Lyzenga, J. (1997). Frequency discrimination of stylized synthetic vowels with a single formant. The Journal of the Acoustical Society of America J. Acoust. Soc. Am., 102(3), 1755. doi:10.1121/1.420085.

Lyzenga, J. (1998). Frequency discrimination of stylized synthetic vowels with two formants. The Journal of the Acoustical Society of America J. Acoust. Soc. Am., 104(5), 2956. doi:10.1121/1.423878.

Patterson, R. D. (1976). Auditory filter shapes derived with noise stimuli. The Journal of the Acoustical Society of America J. Acoust. Soc. Am., 59(3), 640. doi:10.1121/1.380914.

Schwarz, Douglas M. make_detachable function.

Contact Us

For more information, please visit our lab website at www.urmc.rochester.edu/labs/Carney-Lab/. You will find contact information, links to other code packages and literature, and more. E-mail inquiries should be directed to Laurel.Carney@Rochester.edu.