VirtualCall: User Guide (May 15, 2014)

VirtualCall V2.0

Written by Chia-Jung Chang (2014)
Modified from VirtualCall V1.0 by Yi Zhou (2006)
Based upon template structures created by Christopher Dimattina (2005)

Introduction

VirtualCall is a Matlab GUI for synthesizing marmoset vocalizations and complex tone stimuli. It can be used to facilitate neural coding and behavioral studies, in which we can systematically investigate how auditory cortical neurons encode vocalizations and explore the categorical boundary for vocalization representations. It can visualize the spectrogram and envelope of the synthesized vocalizations on your PC, and generate parameterized stimuli for TDT recording system. VirtualCall is implemented based on the methodology described in [Reference Paper]. With this tool, we will be able to understand the neural basis of vocal perception in marmosets quantitatively.

[Reference Paper]
DiMattina C and Wang X (2006) “Virtual vocalization stimuli for investigating neural representations of species-specific vocalizations.” *J Neurophysiol* **95**:1244–1262

Getting Started

Before getting started, you may want to check whether you have all the files with latest version in your VirtualCall V2.0 package, which is put under VirtualCall2.0 folder.

- Package Content ----------

The **VirtualCall2.0** folder consists of the following files.

1. Main GUI files: VirtualCall.m, VirtualCall.fig
2. Call synthesis functions (.m): narrow_synth, twitter_synth
3. Data structure (.mat): TONE, PHEE, TRILL, TRILL_PHEE, TWIT
4. Sub-functions (.m): binone, makeam, makefm
5. Figures (.png): speaker, virtualcall
6. Documents (.pdf): VirtualCall_UserGuide

All the files in this package should be the ones last modified in 2014, and are expected to see % Notes: - VirtualCall V2.0 notified in the codes. If there is any file missing or not being the latest version, feel free to contact the person who maintains these codes.

- Execution Environment ----------

All the programs in this package were written under Matlab R2012a, which should be compatible to any other versions. If there is any compatibility problem, you may want to update your execution environment or extend the supporting functions.

- Outline ----------

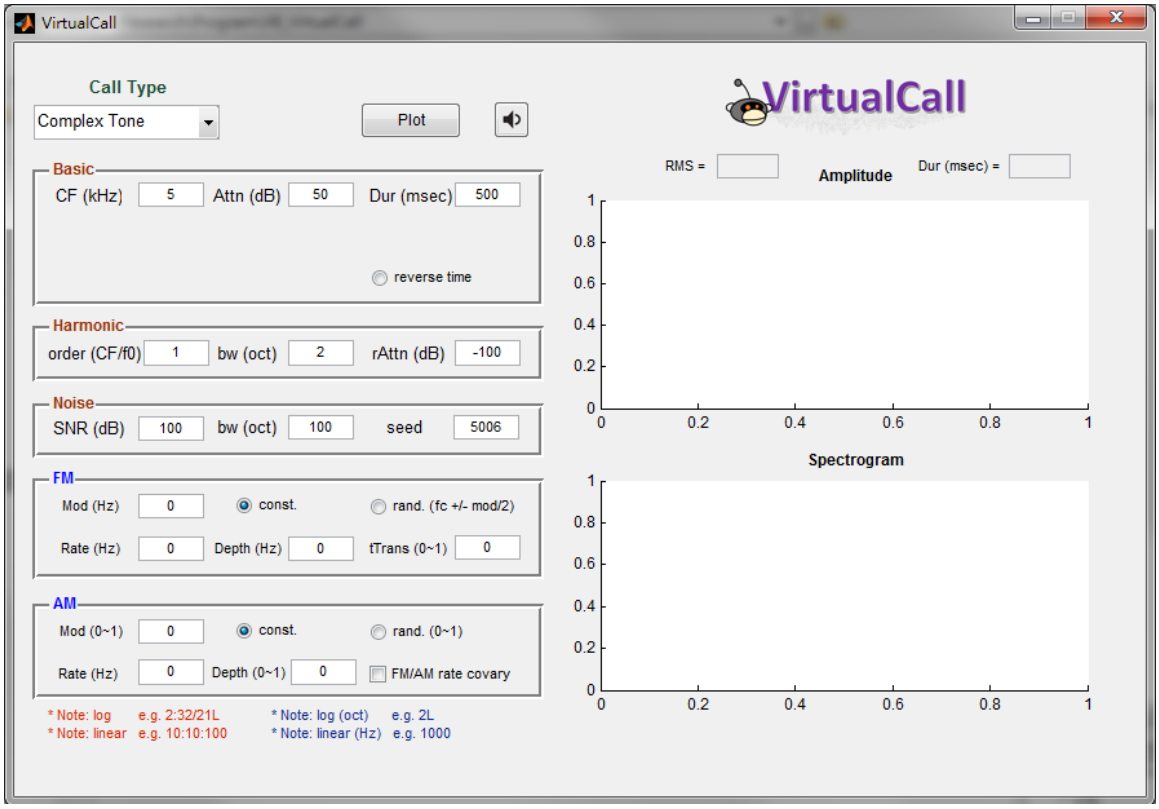
This user guide introduces the major features of VirtualCall using some of the synthesized stimuli screenshot as examples.

In the following you'll learn how to:

- Understand the basic layout of the VirtualCall panel in [**GUI Layout]**
- Visualize the spectrogram and play the sound of the stimuli in [**Plot and Play**]
- Manipulate the parameters for synthesized stimuli in [**Parameter Specification**]
- Deliver the synthesized stimuli for RX6 TDT System [**Interface with TDT System**]

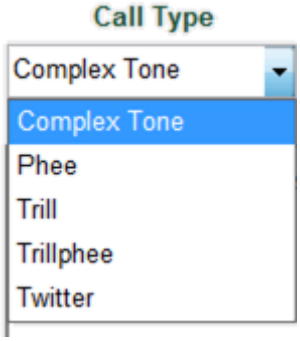
GUI Layout

When you open **VirtualCall** panel, the following window will pop out in your first sight. The default parameters are set to a 5 kHz pure tone at 50 dB attenuated level for 500 msec.

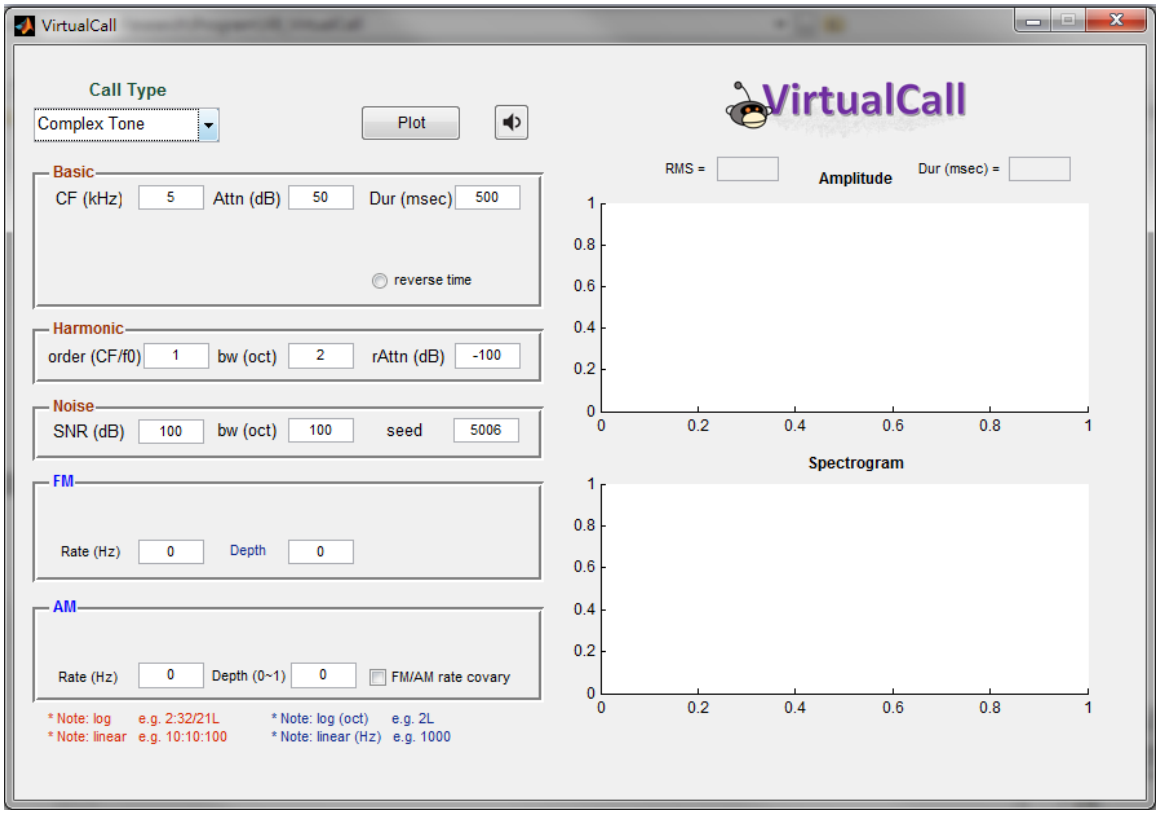


- Call Type Menu ----------

However, before you change any other parameters and synthesize the stimuli, please select a call type from the **Call Type** menu. Even if a 5 kHz pure tone at 50 dB attenuated sound level for 500 msec is exactly what you want, click **Complex Tone** before you do it.



Then, the normal window for **Complex Tone** will be set ready for further use. Note the layout difference between the following figure and the figure on the previous page.



There are five types in total: Complex Tone, Phee, Trill, Trillphee, and Twitter.

1. **Complex Tone**: It not only supports the basic stimulus type in mixed panel such as pure tone, band-pass noise, sinusoidal FM, and sinusoidal AM, but also provides more options like harmonic sounds, co-varied sinusoidal FM/AM, flexibility in modulation depth for FM and AM.
2. **Phee**: It can not only be used to synthesize phee calls with different parameters to investigate how the statistical structures are encoded, but also used with background noise masking and harmonic structure modulation.
3. **Trill**: It is used to synthesize trill calls in a similar way like Phee.
4. **Trillphee**: It is used to synthesize trillphee calls in a similar way like Phee. It provides options to morph trill calls and phee calls.
5. **Twitter**: It is used to synthesize twitter calls in a similar way like Phee, but it does not support changing harmonic number.

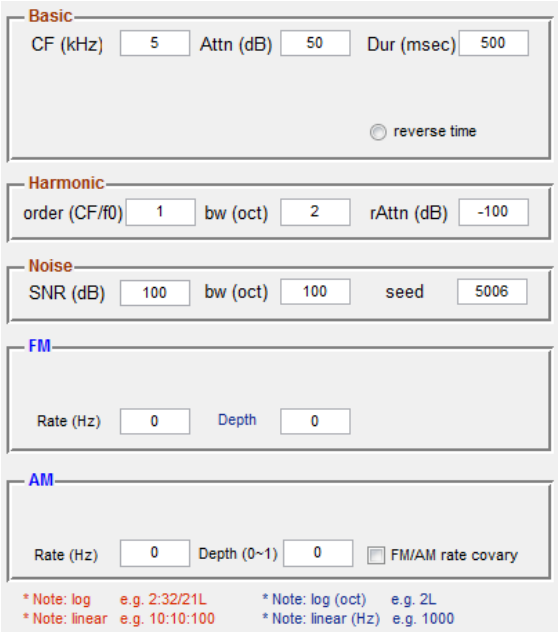
- Parameter Panel ----------

There are five sections on the left-hand side of GUI: Basic, Harmonic, Noise, FM, and AM.

1. **Basic**: It includes basic information like center frequency, sound level attenuation, and duration for complex tones and narrowband calls, or global parameters for twitter calls.
2. **Harmonic**: It allows changing the harmonic number, density, and attenuation.
3. **Noise**: It allows adding background noise to you original signal.
4. **FM**: It allows changing the frequency contour of the calls.
5. **AM**: It allows changing the amplitude contour (envelope) of the calls.

In these sections, you can see several white squares for users to input parameter values. Left to each square is the parameter name. Detailed explanation for what each parameter does and the appropriate input format is described in [**Parameter Specification**].

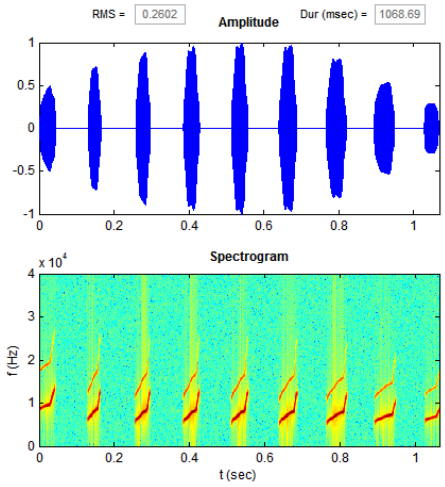
There are default values set for each call type, and you can **input** a set of parameter values to modify stimuli in two different forms as described in the *Note region at the bottom of the panel. One is log scale, and the other is linear scale.



All the parameters will be **reset** to the default values (template) once you click to select a call type from the menu, or re-open the GUI panel.

- Spectrograms Axes ----------

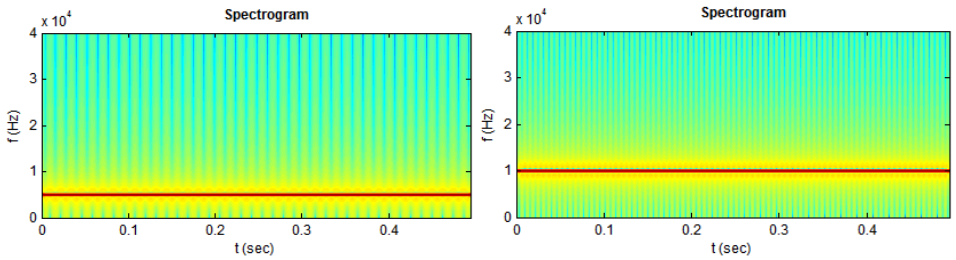
On the right-hand side of this GUI panel, you can see two axes showing the envelope and the spectrograms of your synthesized stimulus. There are two inactive squares showing calculated root-mean-square and duration as well. The following example is a twitter call with background noise synthesized after pushing **Plot** button.



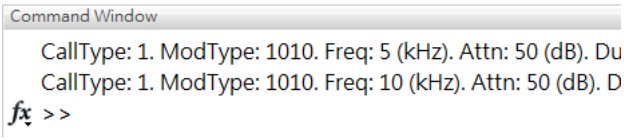
Plot and Play

If you want to visualize the envelope and the spectrogram of stimuli before you actually synthesize them and deliver through TDT System, you can push **Plot** button. If you want to hear what it sounds like, you can push **Speaker** button that is right next to **Plot**.

If there is a set of stimuli, you will see the envelope and the spectrogram of each shown sequentially if you push **Plot**. However, if you push **Speaker**, you will only be able to hear one sound, which belongs to the last stimulus. For example, if you want to visualize 2 pure tones with 5 kHz and 10 kHz, you will see the following, yet you can only hear 10 kHz tone.



At the same time, you will see the header information for each stimulus generated shown sequentially in the command window after pushing **Plot** button.

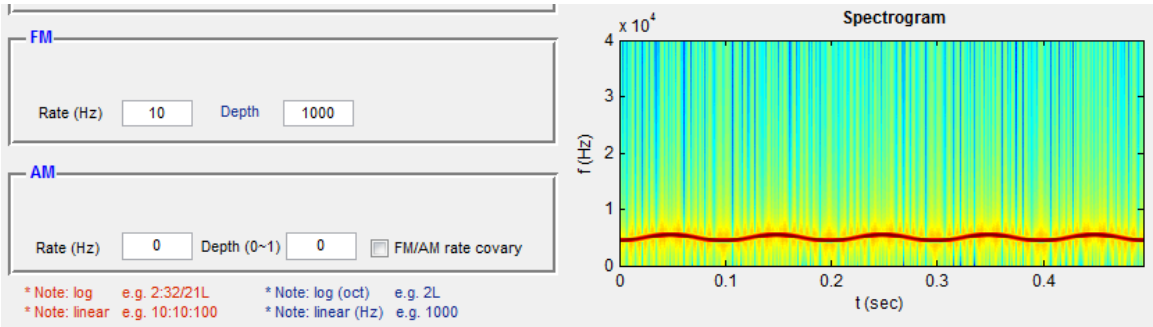


Parameter Specification

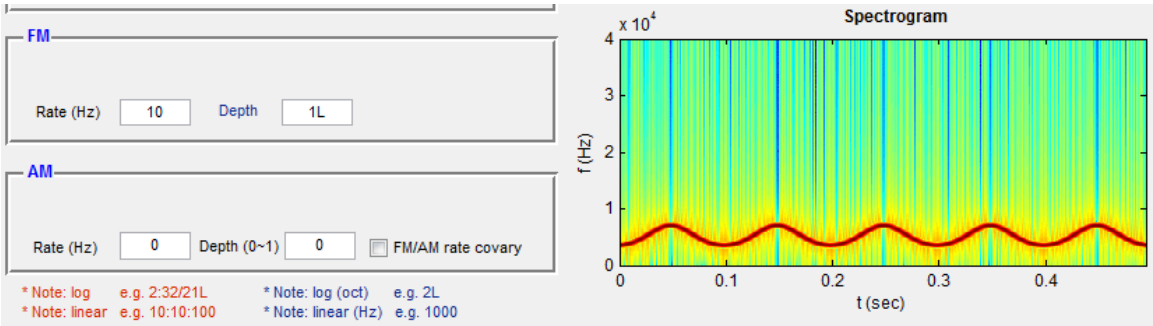
There will be default values set for each call type; however, you can change those directly.

- Input Format ----------

Your input can be a **scalar** value. For example, if you want to synthesize a 10 Hz sinusoidal FM sound with 1000 Hz modulation depth, you can do the following.

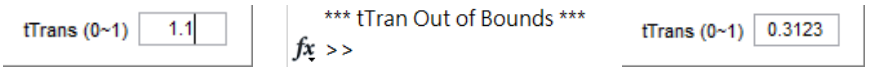


You can also synthesize a 10 Hz sinusoidal FM sound with 1 octave bandwidth by adding '**L**' behind your value. Note that it only applies to FM Depth, not the other parameters.



Your input can also be a **vector.** For example, if you want to synthesize trill calls with fast modulation rate from 10 Hz to 100 Hz with **linear** 10 Hz step, put 10:10:100 in FM Rate. If you want to set the fast modulation rate from 2 Hz to 512 Hz with 1 step per octave, which gives 9 stimuli in **logarithmic** scale, put 2:512/9L in FM Rate.

If you give input with inappropriate value, it will automatically **correct** to the default value and show you the reason on the command window. For example, you input 1.1 in tTrans and push either Plot in VirtualCall, or Make in xblaster3, but the appropriate range should be between 0 and 1, then the value will immediately be set back to 0.3123 (default). At the same time, you can see "*** tTrans Out of Bounds ***" shown on command window.



- Parameter Meaning ----------

The following list explains what each parameter represents in the synthesized stimulus, and the appropriate range for each parameter.

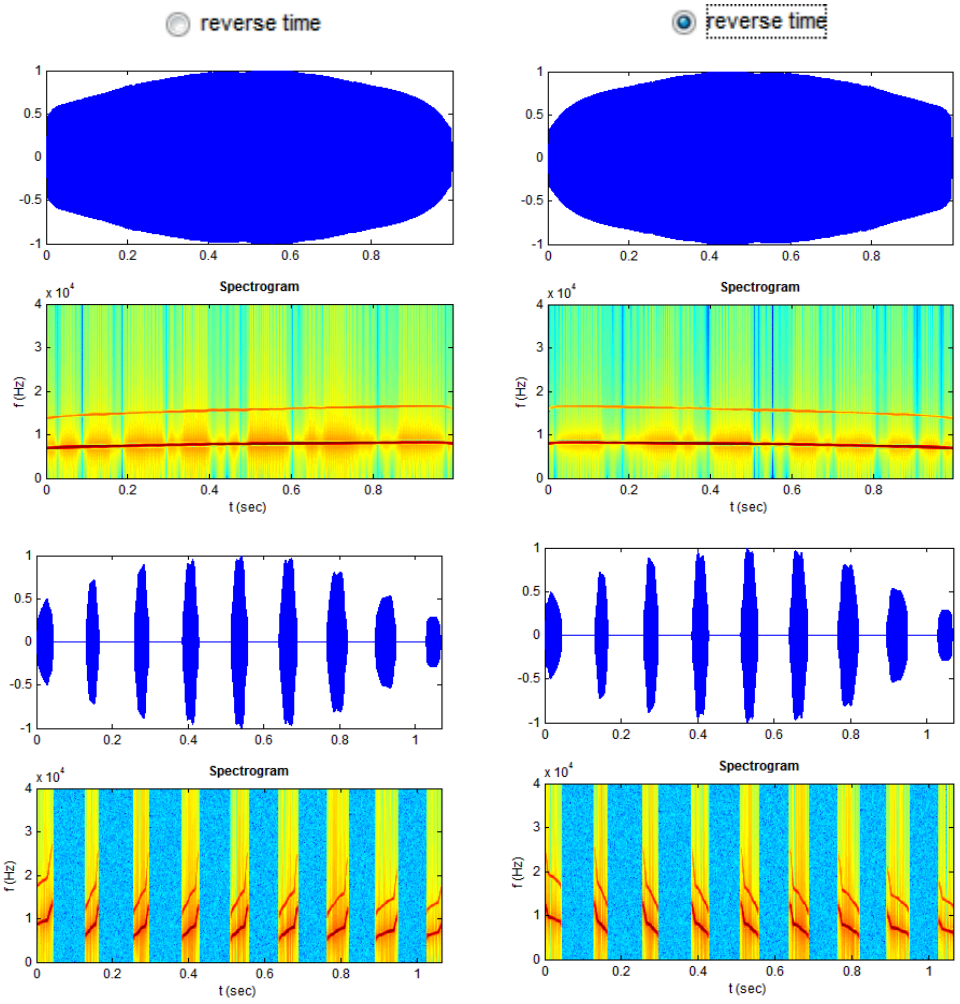
Name	Unit	Definition	Range
CF	(kHz)	Center frequency.	[0, 40]
Attn	(dB)	Attenuation in sound level.	N^0
Dur	(msec)	Duration. Not applicable for user to change in Twitter call, but you can see the duration of the synthesized twitter on the right-hand side of the panel.	[10, ~)
Nphr	---	Number of phrase for Twitter call.	N^+
IPI	(msec)	Mean inter-phrase interval for Twitter call, which is measured between the peaks of each phrase.	(tPhr, ~)
tPhr	(msec)	Mean phrase sweep time for Twitter call.	(0, IPI)
order	---	Ratio of center frequency to fundamental frequency for harmonics. Not applicable for Twitter call.	N^+
rAttn	(dB)	Harmonic attenuation.	Z^- U {0}
bw	(oct)	in Harmonic: Bandwidth for harmonic, centered at CF.	R^+
bw	(oct)	in Noise: Bandwidth for noise, centered at CF.	R^+ U {0}
SNR	(dB)	Signal-to-noise ratio of the signal to the background noise for masking	R^+ U {0}
seed	---	Seed for random generation in Noise, or rand.	N^+
Mod	(Hz)	in FM: Slow frequency modulation depth (M_{FM1})	R^+ U {0}
Mod	---	in AM: Slow amplitude modulation depth (M_{AM1})	[0, 1]
Rate	(Hz)	in FM: Mean fast frequency modulation rate (f_{FM1})	R^+ U {0}
Rate	(Hz)	in AM: Mean fast amplitude modulation rate (f_{AM1})	R^+ U {0}
Depth	(Hz)	in FM: Max fast frequency modulation depth (d_{FM1}^{Max})	R^+ U {0}
Depth	---	in AM: Max fast amplitude modulation depth (d_{AM1}^{Max})	[0, 1]

tTrans	---	Fraction of transition time for Trillphee call.	[0, 1]
bwPhr	(Hz)	Mean phrase bandwidth for Twitter call.	$R^+ \cup \{0\}$
fKnee	---	Fraction of 'Knee' frequency for Twitter call.	[0, 1]
tKnee	---	Fraction of 'Knee' time for Twitter call.	[0, 1]

The following explains what each radio button or checkbox does, by showing the examples.

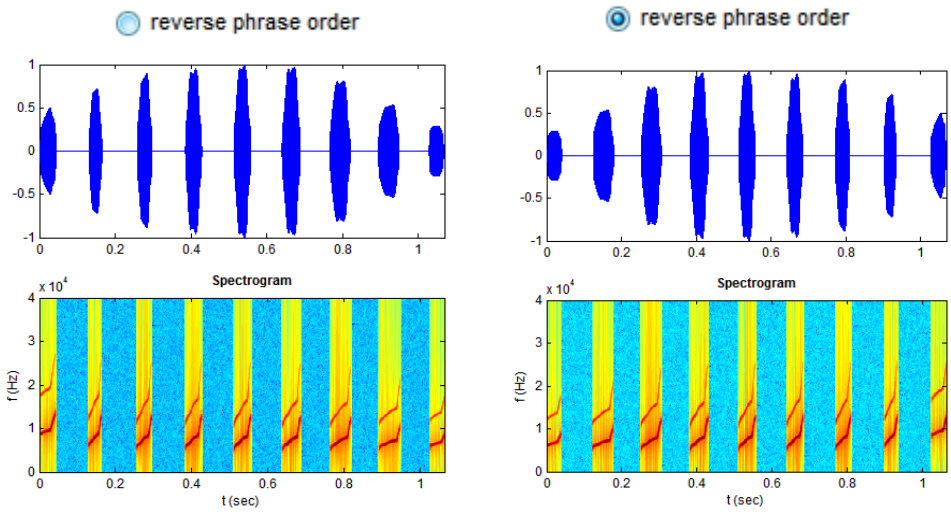
- reverse time

  For complex tone and narrowband call types, selecting reverse time will reverse the temporal structure of both envelope and spectrogram of original stimulus. For twitter call types, reverse time will reverse temporal structure of each phrase individually, but not the whole call.



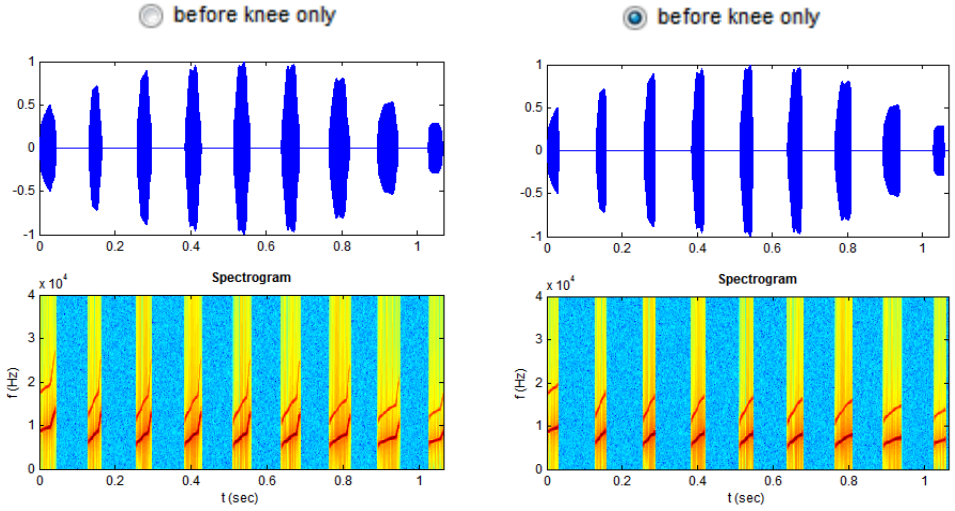
- reverse phrase order

  For twitter call types, reverse phrase order will reverse the order of the phrases, but not the temporal structure within each phrase. If you want to reverse overall twitter call, select both reverse phrase order and reverse time at the same time.



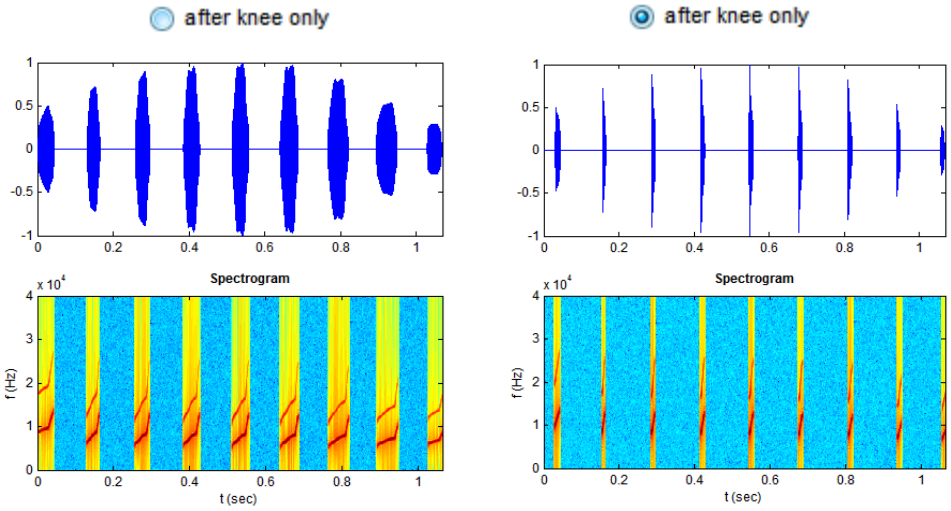
- before knee only

  For twitter call types, selecting before knee only will chop each phrase to the part that is before time of Knee.



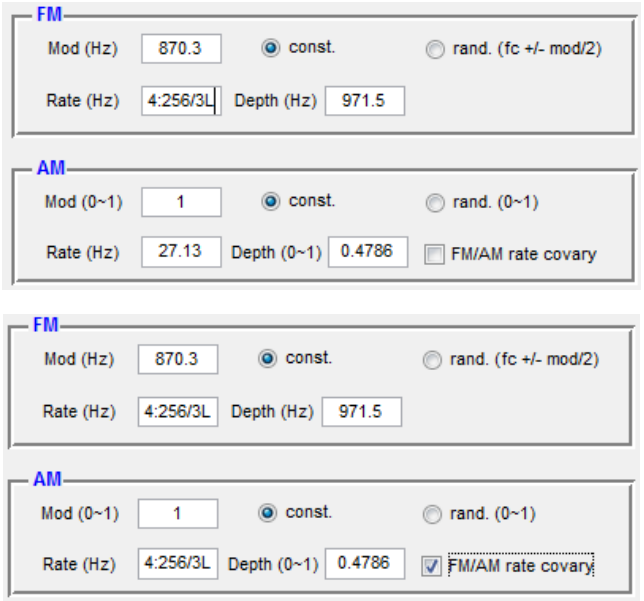
- after knee only

  For twitter call types, selecting before knee only will chop each phrase to the part that is after time of Knee.



- FM/AM covary

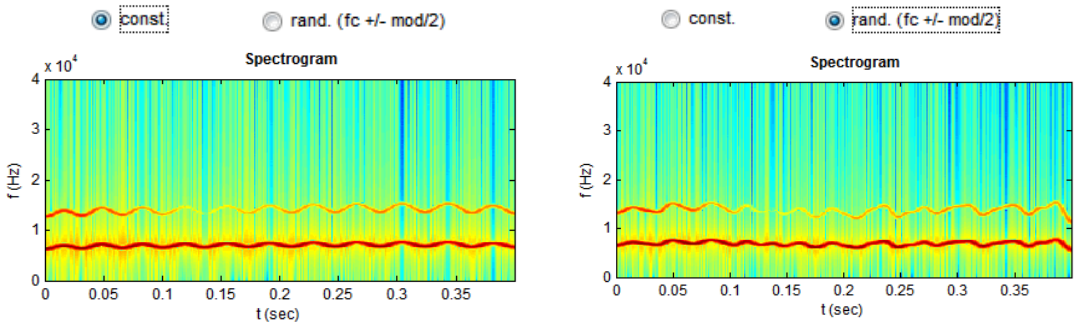
  For complex tone and narrowband call types, if you want to synthesize stimuli which FM rate and AM rate co-vary instead of generating permutations of changing FM rate and changing AM rate, you can set the parameters in either FM Rate or AM Rate. Then click FM/AM covary checkbox, it will automatically change the other values. Note that this checkbox only applies to the situation when one of the input parameters is vector.



- rand. (fc +/- mod/2)

  For narrowband call types, selecting rand. (fc +/- mod/2) will randomize the frequency contour by adding an oscillation (default: mean 4 Hz and std 1 Hz). You can change this oscillation frequency range for randomness in code (contourF and contourFstd).

  Note that the default is set to select const. for normal situations. The program will force you to select to one and only one from const. and rand. (fc +/- mod/2) buttons.

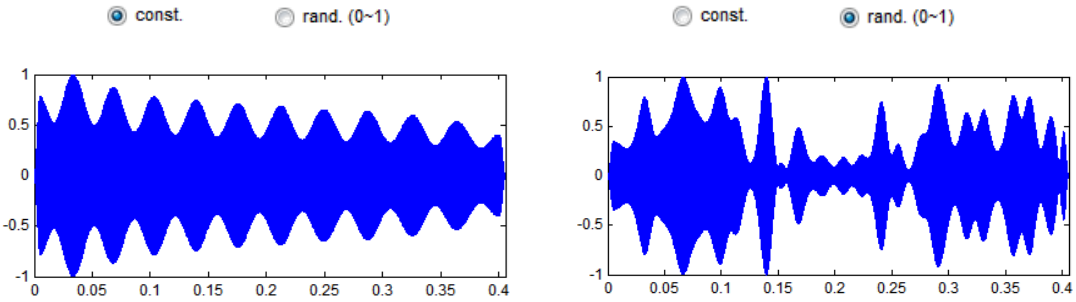


  Note rand. (fc +/- mod/2) cannot be selected with rand. (0~1) at the same time.

- rand. (0~1)

  For narrowband call types, selecting rand. (0~1) will randomize the amplitude contour by adding an oscillation (default: mean 4 Hz and std 1 Hz). You can change this oscillation frequency range for randomness in code (contourF and contourFstd).

  Note that the default is set to select const. for normal situations. The program will force you to select to one and only one from const. and rand. (0~1) buttons.



  Note rand. (0~1) cannot be selected with rand. (fc +/- mod/2) at the same time.

Interface with TDT System

To add VirtualCall panel in xblaster3 for single-unit recording, do the following steps.

1. Rename the folder as **XB_VirtualCall** and put it under \XB_Stimulus_Panels
2. Open xb_xblaster_stimulus_types.dat under \XB_DAT&MAT
   Append VirtualCall (the name of the program) to the above list
3. Open xb_xblaster_cb_get_code.m
   Add **analysis_code** for each analysis_type specified in VirtualCall.m.
   Note that analysis codes assigned for VirtualCall are between 2301 and 2400.

```
case 'phee_cf'
    analysis_code = 2320;
case 'phee_attn'
    analysis_code = 2321;
```

4. Add the command to open the figure file of VirtualCall in xb_xblaster_cb.m

```
case 'virtualcall'
    handle = findobj('Tag','VirtualCall');
    if isempty(handle)
        VirtualCall;
    else
        figure(handle)
    end
```

5. Add the command to evoke **gen_callstim** (sub-function for making stimulus matrix) in xb_xblaster_cb_doit.m

```
case 'virtualcall'
    s_log{end+1,1}='VirtualCall';
    VirtualCall('gen_callstim');
```

6. Open xblaster3 and select VirtualCall in the stimulus list.
7. Set up all the parameters in VirtualCall panel to your desired value, and click **Make** or **DO IT** button to deliver stimuli in your chamber through RX6 TDT system. Note that if you want to deliver the stimuli randomly, just remember to select **Randomly Present** in xblaster3 panel.