earPlugin:Tune user's manual

About earPlugins

An earPlugin is an audio plugin made for ear training (not music making). It integrates sound processing, test flow and score keeping for a training session. The overall design goal is to maximize learning efficiency through immediate and detailed feedback, both aural and visual.

The audio plugin implementation allows you to use earPlugins with the same work environment and material that you normally use.

earPlugins:Tune

The first earPlugin is Tune, for training identification of frequency bands.

Tune is compatible with VST2.4, on 32-bit Windows platforms only (tested on XP and 7).

Installation

Tune.dll is the plugin itself, and can be inserted in any VST compatible host application. Several popular DAW programs are VST compatible (including Pro Tools with a €75 wrapper from FXpansion.com), but in case you don't have a DAW, freeware alternatives for Windows are described in the appendix.

Using it

Insert Tune in an audio track (noting it is stereo), open its editor, start playback.

When Tune opens up, you can just click the different frequency keys, and hear what they sound like on your material.

Test play/pause/stop buttons

These buttons do not start or stop the music, the start and stop a test. Here's how it works:

1. Hit **play**
2. After about two seconds, Tune activates a random frequency band. Your job is to guess which and click the key. If you like, you bypass/enable the EQ filter any time using the **EQ on** button.
3. **If you answer correctly**, Tune paints the key you clicked green, and shows the frequency in the display, while the filter is still active – this is called *illustration*. If this was your first answer, the illustration runs just two seconds before continuing to the next test step, repeating from 2. If it was not your first guess, the duration of the illustration is increased depending on how much your first guess was off. In this way, Tune spends most of your testing time illustrating the frequency bands you have most trouble identifying.
4. **If you answer incorrectly**, Tune paints the key you clicked red, and changes the EQ filter to the frequency you actually clicked. This provides you with an immediate demonstration of the difference between the sound of the unknown frequency band and the one you clicked. After two seconds the EQ reverts to the unknown frequency and you can guess again.

5. You can pause the test by clicking **pause**. This does not affect your score; it only stops the test clock and allows you to freely click and listen to the frequency bands. Resume the test by clicking **play**
   Note: After a pause, you will return to the same test step, but a new unknown filter frequency.
6. Hit stop when you want to end the test

Setup

Hit **Setup** to set the range of filter frequencies, 1/1 or 1/3rd octave resolution, and the amount of boost/cut. The -10dB output attenuator is there because some host applications do not handle it well if Tune returns an output that exceeds ±1.0 [1].

Limit the frequency range enough for all the filters to be easily audible on your test material and monitor hardware. In order to effectively train your ability to identify frequency bands, it should not be hard to hear the effect of the filters.

Performance statistics display

The top display shows info and performance statistics for the ongoing test.

Here is Tune just after a correct answer (2kHz) to test step 34:

The “Test” frame on the left shows that

- The test runs with 12 dB boost filters (as set by the Gain knob under Setup)
- This is step 34
- The test has been running for 6:51

The “Test” column shows the average performance during this test

- The first guesses are off by 0.24 octaves
- The number of guesses before getting it right is 1.24
- The correct answer is given after 6.9 seconds

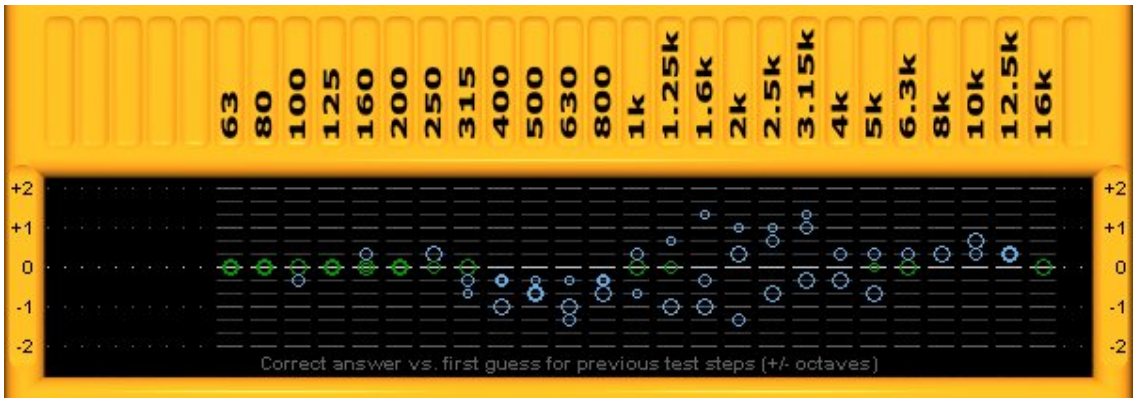
The “Step” column shows the similar data, but for the current test step[2]

[1] Tune itself (and VST I/O in general) uses a 32-bit float sample format, and cannot be overdriven.

[2] The “Step error” is shown as a signed value, but the “Test” average error is unsigned, so it shows variability, not bias.

The history graph

After each test step, a circle is added to the history graph, below the frequency key you used for your first guess. The circle indicates the frequency of the correct answer relative to your first guess. This graph will help you identify trends (if any) in your answers, so you can correct your future answers accordingly. Here is an example of what it could look like after about 60 test steps:



History graph: correct answer vs. first guess, one circle for each test step.
For each key, the 4 latest first-guess clicks are represented, larger circles are most recent.

This tells you that:

- Your answers in the low frequency range are very accurate
- When you have placed answers in the 400-800Hz octave, the actual filter frequencies have consistently been lower. Next time you want to place an answer in this frequency range, try a little lower
- 1k – 4k is your most problematic range
- The highs from 10k and up sound a little higher than you think

Training hints

- Keep an eye on the timer. To keep focus, don't practice more than 20 min. at the time.
- The easiest test signal is pink noise. Preferably left and right channels should be uncorrelated, to avoid comb filtering effects when moving your head.
- Some test material is easier than other. When you find something easy, consider using it for tuning PA's in real life.
- In 1/3rd octave mode, after making wrong answers, you may learn to identify the unknown (correct answer) filter by the sound of the *interval* between it and the wrong answer. 1/3rd octave is a major 3rd interval, though it sometimes sounds more like a 4th, because the ISO filter frequencies are not spaced exactly at 1/3rd octaves. Using the intervals is not cheating, it is useful in real life.
- Suggested progression:
    a. Start with 1/1 oct. +12 dB filters
    b. When your "Test" error drops below 0.15 octaves, move on to 1/3rd octave resolution
    c. When you have a good grip on the +12 dB 1/3rd octave filters, try 1/1 octaves -12dB. It may feel a bit like starting over. At first, use the **EQ on** button to bypass/enable the EQ a few times before answering. It will be easiest to identify the filter right when you bypass it, because the removal of the cut filter will sound a bit like the boost filters you have practiced.

d. As you get a grip on this, challenge yourself not to use the **EQ on** button, since in a real-life scenario, you won't have this ability to bypass the problem you are working to fix.

Motivation

With today's widespread use of computer based audio production, more processing capability is available, at lower cost than ever. This is great, but also means that more than ever, human skill is the limiting factor in audio production. What separates the best productions from the rest is not the exotic gear that went into their making, but the people.

Even this thing can't make a great vocal EQ – on its own

Skill is stronger than gear™

Great training sessions,

Flemming Nyboe
soundguy@nyboe.net

Appendix – free VST hosts for Windows

foobar2000

foobar2000 is a free music player for Windows. It supports ASIO output. It does *not* support VST out of the box, so you need to use George Yohng's VST Wrapper as an adapter between Tune and foobar2000.

1. Download and install foobar2000 from http://www.foobar2000.org/download
2. Download George Yohng's VST Wrapper from http://pelit.koillismaa.fi/plugins/show.php?id=213 (foobarvst.zip)
3. Open the .zip file, locate foo_dsp_vstwrap.dll, and put it in your foobar2000 components folder, typically C:\Program Files\foobar2000\components
4. Start foobar2000, and select File > Preferences (CTRL+P).
5. Under Playback > Output, reduce the buffer to around 200ms (this improves the responsiveness of Tune). If you encounter interrupted playback, try increasing it a little.
6. Under Playback > DSP Manager, in Available DSPs, click George Yohng's VST Wrapper, then click the left arrow to activate it.
7. Click OK at the bottom of the Preferences window
8. A small VST icon appears in the notification area of the Windows task bar. Click it, select VST setup, click + Add folder, and locate the folder containing tune.dll (this can be any folder you choose), and click Close
9. Click the VST icon in the notification area again, this time select Use VST Effect, and select Tune to activate it.

Winamp

Download from http://www.winamp.com/player
The Free version is fine, and you don't need to enter an email address if you deselect announcements. It does *not* support VST out of the box, so you need to use Christian Buddes VST Host Winamp Bridge as an adapter between Tune and Winamp.

Chistian Buddes VST bridge http://www.savioursofsoul.de/Christian/programs/winamp/

1. Install Winamp
2. Run WinAmp_VST_Bridge.exe from any location
3. Install and finally click Yes to launch winamp
4. Left-click winamp (so it gets focus) and press CTRL+P for Preferences
5. Under Plug-ins > Output, select "Nullsoft DirectSound Output" and click "Configure". On the "Buffering" tab, decrease "Buffer length" to 200ms and click "OK" (this improves the responsiveness of Tune). If you encounter interrupted playback, try increasing it a little.
6. Under Plug-ins > DSP/Effect, select "VST Host DSP...", then click "Configure Active plug-in" to open the VST bridge. A small window appears; left click its "VST plugin" text field, and select "Load DLL", then locate the Tune DLL on your hard drive and click "Open"
7. Play your favorite ear training music in winamp while using Tune as described above