### ***Music 257***

### **LAB 5: DESIGNING AN EXPERIMENT**

Keep thinking about your final project! What do you want to do? Maybe talk to some other students about collaborating!

I. DESIGNING AN EXPERIMENT: Due 5/10/15

In this lab, you and your group will be designing and capturing data to measure one of

four perceptual metrics identified below.

*OPTIONS:*

- 2-3 equal loudness contours derived from data captured at:

[100 500 1000 2500 5000 10000] Hz

[Intensities of your choice – one softer, one

loud but comfortable]

**[Helpful tips – You will want to use a form of Method of Adjustment for this test.**

**Here the reference is always 1kHz. The listener is tasked with manipulating the**

**level of each comparison frequency to the level of the 1kHz reference. Record the**

**value at the point that they identify a given frequency as equal in loudness to the**

**1 kHz reference. The curve is the line that connects the values of your averaged**

**subjects across frequency. You will need to collect data for each listener at ALL**

**frequencies.]**

- A listener’s audiogram captured at the following frequencies:

[50 100 250 500 1000 2000 4000 8000 16000] Hz

**[Helpful tips – You will want to use a 2-Interval- Forced Choice (2IFC) for**

**this test or a Method of Adjustment. If you use Method of Adjustment you must**

**capture both directions of change on separate trials to mitigate effects of**

**hysteresis influencing your threshold values. If you use 2IFC you can use either a**

**Method of Constant Stimuli or an Up-Down Tracking Method. Each trial of a 2IFC**

**will always be in the form Reference-A- B or Reference-B- A and include [R-An**

**audible reference tone of the frequency that is always played first, A-a target**

**interval that contains the level adjusted target tone, B-a non-target interval that**

**does not contain the level adjusted tone. The listener would need to press a**

**button to make each play, and the ordering of A and B should be randomized**

**across trials. For a 2IFC Up-Down tracking Method or any 2IFC test, target 70-75%**

**correct as your Threshold. The audiogram curve is the line that connects the**

**values of your averaged subjects across frequency. You will need to collect data**

**for each listener at ALL frequencies.]**

- A critical band curve (frequency as a function of bandwidth) captured for the

following two frequencies:

[800 8000] Hz

**[Helpful tips – You will want to use a 2-Interval- Forced Choice (2IFC) for this test.**

**You are tracking the threshold level of a tone at each of the above frequencies in**

**the presence of a band-passed noise. The noise will need to be tested at**

**incrementally increasing steps as the width of the noise band increases. The**

**noise should be centered on the frequency you are testing. The energy in each**

**frequency should be identical. If you use 2IFC you can use either a Method of**

**Constant Stimuli or an Up-Down Tracking Method. For a 2IFC Up-Down tracking**

**Method or any 2IFC test, target 70-75% correct as your Threshold. Plot the**

**threshold value of the tone as a function of the masker bandwidth. The key**

**bandwidth you are looking for is the point of inflection.]**

- A curve depicting the fission boundary of proximity streaming as a function of

speed across the following frequency pairs:

[400 500] Hz

[400 650] Hz

[400 800] Hz

[400 2000] Hz

**[Helpful tips – You will want to build a galloping sequence that alternates**

**between the two paired frequencies in sets of three. For each pair you will build a**

**galloping sequence that increase in speed across time. It is critical you instruct**

**your listeners to work to hold the galloping sequence together as a single stream**

**(eg they hear the triple pattern) as long as possible. This must be their task in the**

**trial. You will want to use a controlled increase staircase method. Here you are**

**increasing the speed of the sequence and measuring the point at which the**

**listener can no longer hold the sequence together. YOU MUST identify the speed**

**(time between each of the elements in the triple) at the point where the listener**

**hear the sequence separate into two streams. That is the number you would**

**record for each of the pairs. The fission boundary curve is the line that connects**

**the values of your averaged subjects across these frequencies and speeds. You**

**will need to collect data for each listener at ALL frequencies. You will want to**

**plot frequency separation between the elements in the galloping triple as a**

**function of speed.]**

Once you have decided on a perceptual metric you will capture, decide how you will

build your test. Unity, GameSalad, or any other platform can be used. Matlab or any

other software may be used for stimulus generation or test interface.

Consider what the curves you identify from your measurements might be expected to look like. What method will you use for your measurements? How many trials will you capture for each comparison point? You should capture your data on at least 2-3 subjects. You may use yourselves as subjects. After you have collected all your data, you will need to write a brief report [1-2 pages]

summarizing your findings.

This should include:

*Intro*

* A brief description of the question you were trying to answer – what and why you

collected your measurements. [1 paragraph]

*Methods*

● The way the experiment was designed. What type of methodology did you use

and why?

● Potential problems with the experiment (Confounds? Error?)

● Information about the sample size and demographics of your experiment [1-2 paragraphs]

*Results*

● A plot or plots summarizing the data you captured

● A few sentences summarizing what you found [1-2 plots, 1-2 paragraphs]

E-mail us the report, and please be sure to send a short video of your experiment running!

II. MATLAB: Due 5/12/16 at Midnight

**BUILDING A BASIC PITCH TRACKER:**

For this (final!) Matlab assignment, we’re going to be building a function called trackPitch:

[output] = trackPitch(input,fs, windowSize)

This function is going to go through the input signal, break it up into chunks that are windowSize samples long, and write to output a sawtoothTone at the loudest frequency in that window.

To do this, we will need to do the following:

1. Create a for loop that repeats length(input)/windowSize times.
2. For each repetition of the for loop we get a chunk of the input signal. For example, for the first repetition we get chunk = input(1:windowSize), for the second iteration we get chunk = input(windowSize+1:2\*windowSize), so on and so forth. How can we use our for loop to get the right chunk based on what number iteration we are on?
3. Next, we use our getSpectrum function from lab 3 on that chunk. We want to get the loudest frequency in that chunk, so use [M,I] = max(Y), where Y is the Y returned from getSpectrum. The I value will be the index of the biggest value in Y, so we are looking for the corresponding value in the F from getSpectrum, which will be F(I).
4. Append a sawtooth at the frequency we found to output. We can do this using the [A,B] notation, which will concatenate vectors A and B. Thus, we can recursively append a sawtooth to our output on each iteration of our for loop by typing output = [output, sawtoothTone(F(I),0, windowSize/fs, 8, fs)];

Here I’ve used 8 harmonics in my sawtooth tone, but feel free to use however many you want.

After you’re done with the function, send me a recording of yourself singing a melody for 15 seconds or so (or playing one on a monophonic instrument), and a wav file of the output of trackPitch where the input and fs is from your recording. You can choose the window size, but let me know what it is!