Communication acoustics ELEC-E5600: Project work 2017 autumn

Ville Pulkki, Juhani Paasonen

Aalto University

1 Introduction

The course work contains 1) composing listening test sound items, and 2) running multiple-stimulus listening tests to the students in the course. The sound samples are also 3) analyzed with an auditory model, based on the simple filter-bank-based auditory model in the book. 4) The results from the listening test are compared with the results from the listening tests in final report.

2 Deliverables and deadlines

21.9	Exercise topics assigned. You will get one of the topics: pitch strength / sharpness / dissonance / muddyness / tonality / buzzyness
28.9	Listening test items (6) delivered to mycourses
1–3.10	Web-based listening tests start
- 5.10	Listening tests: you have to listen to tests of **all** other groups, and upload results to mycourses. The result page of the software has to be printed to a pdf file, and uploaded.

12.10	Delivery of **Listening test results** as a boxplot + few comments to mycourses as a short pdf document.
10.10-19.10.	Programming auditory model with Matlab and analyzing the test sounds
19.10	Show auditory modeling code and results to Juhani / (pass/fail) grading
24.10	(examination)
26.10	Workshop

3 Listening test items

Definitions for test items:

- Stereo sound having equal signal in both channels, i.e., diotic presentation of sound.
- Format: .wav
- Sampling rate 48kHz
- the items are presented as 6 sound files, naming `01.wav`, `02.wav`, `03.wav`, `04.wav`, `05.wav` and `06.wav`.
- Length of each test item: 2-5 seconds, all test items have to have equal temporal length in audio samples
- Make a 20 ms fade-in and fade-out. I.e., window the start of the signal with a ramp from 0 to 1, and the end of the signal with ramp from 1 to 0.
- Items can be recorded with, e.g., smartphone. In some cases synthesized items are accepted, if mentioned in the description of the work.
- The test signals are presented by $x_n(t)$, where n is the sound file number (1..6), and t is time index. Adjust the loudnesses of the test signals $x_n(t)$ to be equal by your own listening. This can be done by hand-tuning a test-item-specific gain factor g_n for each item to derive signals with equal loudness $y_n(t) = g_n x_n(t)$.
- None of the signals may contain values over 1.0 or below -1.0. Thus, normalize also the level of all signals using the largest maximum absolute value of audio sample values found in the signals. This means that you have to divide the signals by the highest peak absolute value of all of the test signals, this can be done as

$$z_n(t) = 0.999 \frac{y_n(t)}{\max_n \left(\max_t \left(|y_n(t)| \right) \right)},$$

where $z_n(t)$ presents the final signals to be delivered.

- The test items have to include sound from only one prominent source. If you are recording in real environments, avoid background sounds and background noise. The microphone noise may also be harmful to the test, if the real environment sounds are soft. Choose sounds that can be recorded with your device, for example, too high sound pressure levels may distort the sound in undesired manner.
- Provide also the question made to the listener (e.g., How muddy is sound?), and also the low and high words describing the ends of the scale (clean – muddy).

I will listen to the test items, and accept or reject them. In case of rejection I will ask you to correct something and you have very limited time to conduct the changes. Take care that the loudnesses perceived from the items are equal. This means that you really have to tune the individual levels of the items until they are perceived to be equally loud. There is no simple method to do this, though quite often equalizing the p_{rms} values of each item to be the same can be a good starting point.

4 Listening test results deliverable

Make a report with short introduction, description what was done, and plotting of the results. The recording of items should be described, and the reasons why some sounds were selected should be explained. The results can be plotted using Matlab boxplot -function. The template will be provided for the document later.

5 Workshop presentation

10-minute presentation of the work. The presenter from each group is decided by Ville in the workshop. Make slides in pdf format, and send to mycourses before the workshop.

6 Grading the project work

The work will be graded with accept/reject, based on following attributes:

- Listening test item quality, following the guidelines
- Taking part with concentration to other group's listening tests. Lack of concentration is shown in test results.
- Listening test results -report
- Meeting deadlines.

Pitch strength

Listening test. Collect six sounds with different pitch strength, but the same pitch. See Ch 10.1.1 in the book. You may synthesize the sounds using Matlab, or use, e.g., a synthesizer or different music instruments to generate the test items, or you can record some of them from real world. Use also a pure tone to represent higher end of the scale, and a low- or high-passed noise signal as low-end anchor.

Measure the pitch strength of the items using listening test. Plot the result as Matlab boxplot to the report.

Auditory model. Use the auditory model shown in the book chapter 13.6.1. Plot the outputs as in figure 13.12.c). Plot some of them to the report, and comment what you see there. Do you see a method to estimate perceived pitch strength from the critical band outputs?

Sharpness of sounds

Listening test. Collect six sounds with different sharpnesses. Two of the sounds can be created by DSP in Matlab, namely, a sound with very low sharpness (noise low-passed with 200 Hz cutoff), and the other sound should be noise with very high sharpness (such as noise filtered with 10 kHz high-pass). Four of the sounds have to be recorded from environment, and they should have varying sharpness by initial listening. For example, ventilation humming, motor sound, people babbling in reverberant environment. All the sounds have to be continuous without prominent changes in level, so for example speech, high-tempo music or gun fire are not accepted.

Measure the sharpness of the items using listening test. Plot the result as Matlab boxplot to the report.

Auditory model. Use the auditory model shown in the book chapter 13.6.1. Use the program lines before the line with comment "autocorrelation for each band". Plot the outputs as in figure 13.12.c). Plot some of them to the report, and comment what you see there. Implement the computation of sharpness (ch 11.1). Compute a sharpness value for each test item, and plot the result for each item. Compare to the listening test result.

Dissonance of intervals

Listening test. Collect six sounds with different intervals. Synthesize the sounds using additive synthesis in matlab. Use 5–10 harmonics to produce clear perception of dissonance and consonance, and use always the same lower note. Choose such lower note that clear changes in dissonance are heard. Use one interval with very low dissonance, e.g., one octave, and use also another interval with very high dissonance, e.g., semitone. Different versions of different intervals would be interesting, such as tunings of the third found in different scales.

Measure the dissonance of the items using listening test. Plot the result as Matlab boxplot to the report.

Auditory model. Use the auditory model shown in the book chapter 13.6.1, all lines before line with comment "autocorrelation for each band". Plot the outputs as in figure 13.12.c). Plot some of them to the report, and comment what you see there. Implement a model for roughness (ch 11.13). In principle, it should be possible by

- use auditory filter bank to get signals $x_n(t)$
- half-wave rectify each of them
- downsample the signals with ratio of 1:100
- design a band-pass filter with center frequency near 60Hz
- sum the absolute values of outputs

As single operation this would look something like this:

$$r_n = \sum_t |\mathrm{bandpassfilter}(\mathrm{downsample}(\max(0, x_n(t))))|$$

Take absolute value of the new outputs, and sum over frequencies and time.

The result r_n should have higher value for dissonant intervals, and lower value for consonant intervals.

Brightness of guitar power chords with different distortions

Listening test. Collect six sounds with same interval (fifth). Use single guitar, and use different distortion effects. Choose such effects that clear changes in scale *dark – bright* are heard. Record one sound without distortion, and another sound with very high distortion. Different versions of distortion would be interesting.

Measure the brightness of the items using listening test. Plot the result as Matlab boxplot to the report.

Auditory model. Use the auditory model shown in the book chapter 13.6.1. Plot the outputs as in figure 13.12.c) and d). Plot some of them to the report, and comment what you see there. Does autocorrelation value or change in specific loudness with frequency correlate with perceived brightness? Is there something else in filter bank outputs that could explain the perception?

Tonality

Listening test. Collect six sounds with different tonality. Record continuous sounds that have more and less static frequency components. Choose such effects that clear changes in tonality are heard. **This is not about musical tonality, this is about psychoacoustic tonality: how much a sound resembles noise, or resembles a sinusoid.** For reference, one of the items should be pink noise, and another should be a sinusoid. If you use musical instruments, do not make intervals or chords, only single notes.

Measure the tonality of the items using listening test. Plot the result as Matlab boxplot to the report.

Auditory model. Use the auditory model shown in the book chapter 13.6.1. Plot the outputs as in figure 13.12.c) and d). Plot some of them to the report, and comment what you see there. Do autocorrelation functions correlate with perceived tonality? If you plot time-averaged output of the auditory filters depending on frequency for different cases, do you see prominent frequency peaks with most tonal sounds?

Impulsiveness

Listening test. Collect six sounds with different impulsiveness. Choose such sounds that clear changes in impulsiveness are heard. To provide lower anchor, one of the sounds should be continuous sound with no temporal fluctuations. The other end should be maximally impulsive, such as impulse train generated with matlab. The remaining four sounds should be recorded from your surroundings. They should originate from a single source, and to have different level of impulsiveness. Select such sounds that have constant impulsiveness along time, i.e., the impulsiveness of the sample is similar in each temporal part of the test item.

Measure the tonality of the items using listening test. Plot the result as Matlab boxplot to the report.

Auditory model. Use the auditory model shown in the book chapter 13.6.1. Plot the outputs as in figure 13.12.c) and d). Plot some of them to the report, and comment what you see there. Consider some methods to measure impusiveness. For example, does the crest factor computed over single auditory band, or over a sum of bands explain perceived impulsiveness?

Buzzyness

Listening test. Use Matlab to form six test items with different buzzyness. Take a sawtooth signal with fundamental frequency in range of 50-100 Hz, and filter only the partials, e.g., around 2 kHz – 8 kHz. Alter the phase relationships between partials using, e.g., convolution with short noise burst. Choose such sounds that clear changes in buzzyness are heard. To provide lower anchor, one of the sounds should be minimally buzzy, and the other end should be maximally buzzy.

Measure the buzzyness of the items using listening test. Plot the result as Matlab boxplot to the report.

Auditory model. Use the auditory model shown in the book chapter 13.6.1. Plot the outputs as in figure 13.12.c) and d). Plot some of them to the report, and comment what you see there. Consider some methods to measure buzzyness. For example, does the crest factor computed over single auditory band, or over a sum of bands explain perceived buzzyness?