CMPU4018: Lab 5

Seán O'Leary

Instructions

The aim of this lab is explore human auditory perception using python:

- Explore auditory perception using real examples
- Use python to manipulate tones and visualise masking
- Demonstrate the interaction between speech production and perception

Save your exercises in a single lab5.py script and submit the results via webcourses. Answer the questions as comments in your python script.

Exercises

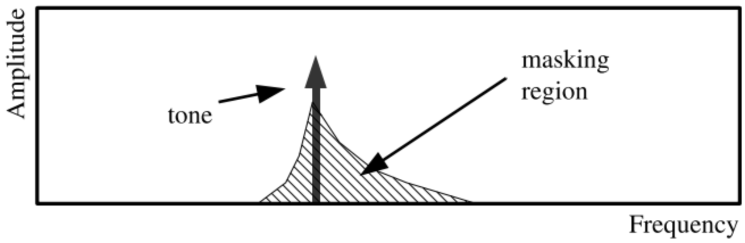

Figure 1: Illustration of frequency masking

1. (3 points) **Combination Tones**
Here we will look at a combination illusion where combining individual tones creates an additional tone fundamental frequency that we can perceive.

The Missing Fundamental
In this exercise you will create four sinusoidal tones. By summing them together you will create a harmonic tone and by leaving out f1, you will see and hear what happens when tones are combined.

(a) Create a tone f1 = 196 Hz and f2 → f4 as multiples of f1 i.e. 392, 588, 784.

(b) Using the four signals, sum them to create two new signals: f_{14} = f1 + f2 + f3 + f4 and f_{24} = f2 + f3 + f4 i.e. the second signal doesn't contain the 196 Hz tone.

(c) Make a wave file output for the signals and play them. what do you notice about the two files (differences/similarities)?

2. (3 points) **Masking Effects**
In the lecture we introduced frequency and temporal masking phenomena, where perceptually only one signal is heard by a listener as one signal masks the presence of another. In this exercise you will create a demonstration of frequency masking, as illustrated in Figure 1

(a) Create a tone with frequency, f1 = 800 Hz, amplitude, A1 = 0.7.

(b) Create a second tone (f2 = 840 Hz, A2 = 0.01).

(c) Make a wave file output for both signals with durations, d = 2 seconds

(d) Combine the signals (sum them) and create a combined wav. Plot the spectrum.

(e) Concatenate the wav files into a single file and listen to the output (each tone playing separately followed by the combination signal)

3. (4 points) **Speech Perception**
In lecture 4 we introduced speech characteristics and the phoneme. In this exercise we will examine how the brain interprets speech by creating piecewise backwards speech, i.e. breaking the speech signal into short segments, reversing them and putting it back together.

(a) Load a speech file

(b) Break the signal into chunks of 30 ms.

(c) Reverse the samples in each chunk but keep the chunks in the right order.

(d) Play the resulting wav. Can you understand it?

(e) Repeat the experiment with chunk durations of 60,120 and 240 ms and listen to the results. Comment on the intelligibility trend and what could be responsible for it.

(f) Instead of reversing the sample chunks in time, flip them in amplitude (i.e. multiply the chunk by -1). Does this impact the intelligibility? Comment on the effect.

Sample code:

```
import soundfile as sf
import numpy as np
y, sr = sf.read('196959_margo-heston_i-see-nine-apples-m.wav')

chunklen=.03 #30ms chunks to flip
chunksamples= int(sr*chunklen)

totalchunks=np.floor(len(y)/chunksamples).astype(int)
yRevChunk = np.zeros(y.shape)
for i in range(0,totalchunks):
```

```
        startidx=i*chunksamples
        stopidx=(i+1)*chunksamples
        # chunk into chunksamples sized samples and flip
        yRevChunk[startidx:stopidx:] = y[startidx:stopidx:][::-1]

sf.write('chunkReverse.wav', yRevChunk, sr)
```