Digital Signal Processing Lab
Experiment 6(b)
Chimaeric sounds reveal dichotomies in auditory perception

Name: Anand Jhunjhunwala
Roll No: 17EC30041
Group No: 63

Objective:

The goal of the assignment is to construct a pseudocode of the stimulus mentioned in the paper titled "Chimaeric sounds reveal dichotomies in auditory perception" .

Code and explanation:

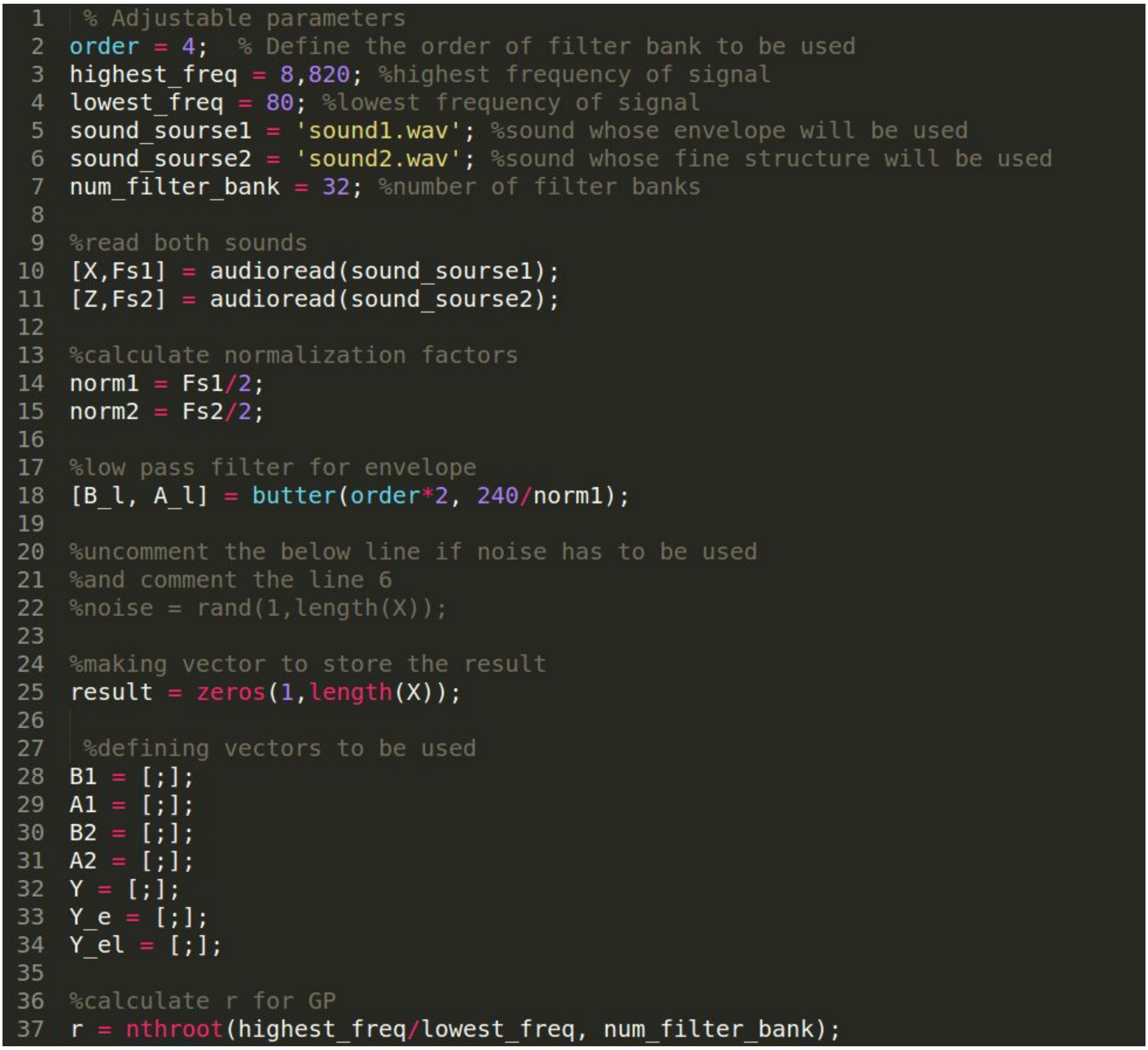
In the paper, the authors state that neurons in the auditory brainstem sensitive to features like sound envelop (slow variations in the sound) and the fine structure (higher frequency fluctuations) in the sound apart from its frequency components. So they investigate the relative perceptual importance of envelope and fine structure, by synthesizing stimuli called '**auditory chimaeras**', which have the envelope of one sound and the fine structure of another.

Synthezation of auditory chimaeras.

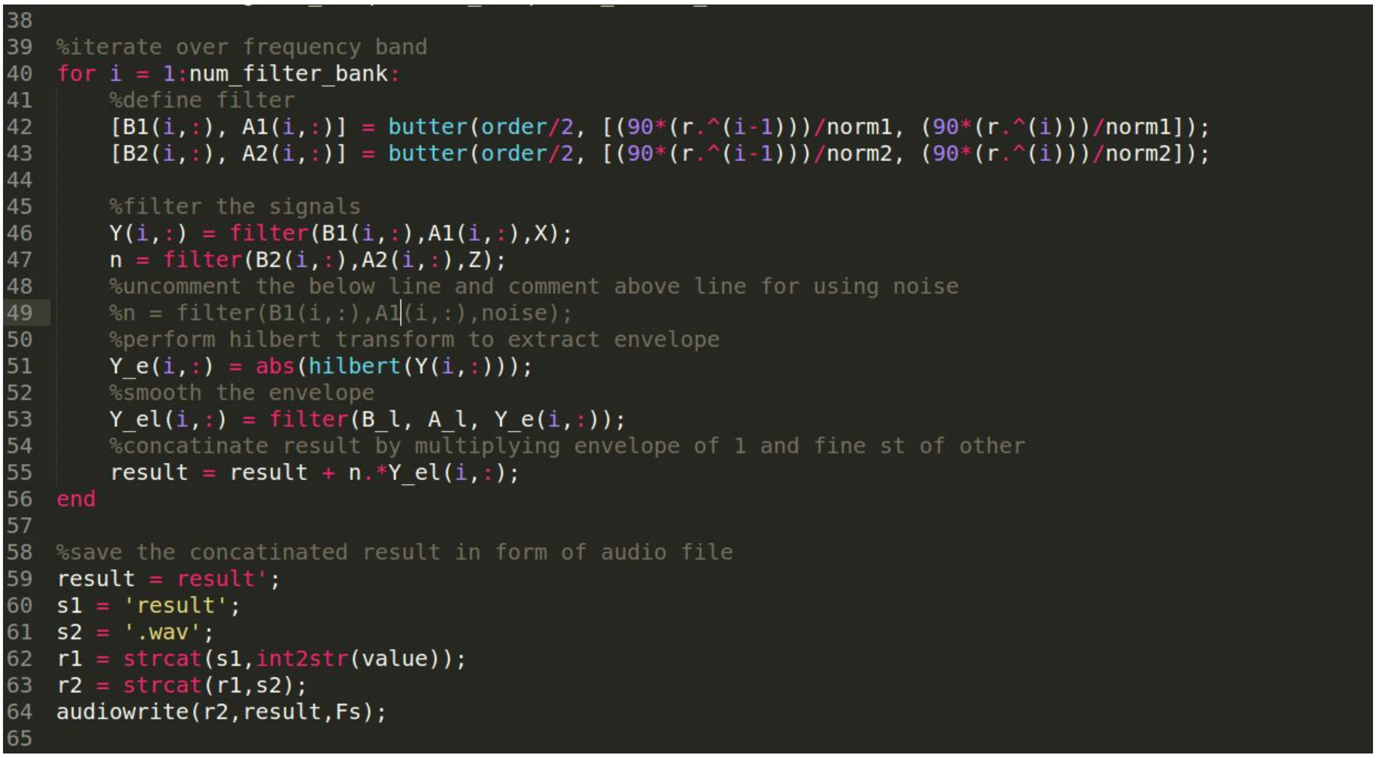
To synthesize auditory chimaeras, two sound waveforms are used as inputs.
Step1: A bank of band-pass filters is used to split each sound into 1 to 64 complementary frequency bands spanning the range 80–8,820 Hz.
Step2: The output of each filter is factored into its envelope and fine structure using the Hilbert transform.
Step3: The envelope of each filter output from the first sound is then multiplied by the fine structure of the corresponding filter output from the second sound.
Step4: These products are finally summed over all frequency bands to produce an auditory chimaera that is made up of the envelope of the first sound and the fine structure of the second sound in each band.

Note: The primary variable in this study is the number of frequency bands, which is inversely related to the width of each band.

Note: See code on next page



```
1   % Adjustable parameters
2  order = 4;  % Define the order of filter bank to be used
3  highest_freq = 8,820; %highest frequency of signal
4  lowest_freq = 80; %lowest frequency of signal
5  sound_sourse1 = 'sound1.wav'; %sound whose envelope will be used
6  sound_sourse2 = 'sound2.wav'; %sound whose fine structure will be used
7  num_filter_bank = 32; %number of filter banks
8
9  %read both sounds
10 [X,Fs1] = audioread(sound_sourse1);
11 [Z,Fs2] = audioread(sound_sourse2);
12
13 %calculate normalization factors
14 norm1 = Fs1/2;
15 norm2 = Fs2/2;
16
17 %low pass filter for envelope
18 [B_l, A_l] = butter(order*2, 240/norm1);
19
20 %uncomment the below line if noise has to be used
21 %and comment the line 6
22 %noise = rand(1,length(X));
23
24 %making vector to store the result
25 result = zeros(1,length(X));
26
27  %defining vectors to be used
28 B1 = [;];
29 A1 = [;];
30 B2 = [;];
31 A2 = [;];
32 Y = [;];
33 Y_e = [;];
34 Y_el = [;];
35
36 %calculate r for GP
37 r = nthroot(highest_freq/lowest_freq, num_filter_bank);
```



```
38
39 %iterate over frequency band
40 for i = 1:num_filter_bank:
41     %define filter
42     [B1(i,:), A1(i,:)] = butter(order/2, [(90*(r.^(i-1)))/norm1, (90*(r.^(i)))/norm1]);
43     [B2(i,:), A2(i,:)] = butter(order/2, [(90*(r.^(i-1)))/norm2, (90*(r.^(i)))/norm2]);
44
45     %filter the signals
46     Y(i,:) = filter(B1(i,:),A1(i,:),X);
47     n = filter(B2(i,:),A2(i,:),Z);
48     %uncomment the below line and comment above line for using noise
49     %n = filter(B1(i,:),A1(i,:),noise);
50     %perform hilbert transform to extract envelope
51     Y_e(i,:) = abs(hilbert(Y(i,:)));
52     %smooth the envelope
53     Y_el(i,:) = filter(B_l, A_l, Y_e(i,:));
54     %concatinate result by multiplying envelope of 1 and fine st of other
55     result = result + n.*Y_el(i,:);
56 end
57
58 %save the concatinated result in form of audio file
59 result = result';
60 s1 = 'result';
61 s2 = '.wav';
62 r1 = strcat(s1,int2str(value));
63 r2 = strcat(r1,s2);
64 audiowrite(r2,result,Fs);
65
```

Note: Code is well commented for easy understanding.
For each stimulus change see section below.

For Simulation Study 1:

Part a:

- Comment line 6, 11, 15, 43, 47 and uncomment line 22 and 49.
- Replace Sound1.wav with the HINT sentence sound filename.
- Change num_filter_bank and obtain the result.
- For using fine structure of sound just exchange noise and X in code.

Part b:

- No change needs to be done.
- Sound1.wav will be a HINT sentence whose envelope has to be used.
- Sound2.wav will be a HINT sentence whose fine structure has to be used.
- Change num_filter_bank and obtain the result.

For Simulation Study 2:

- No change needs to be done.
- Sound1.wav will be a melody whose envelope has to be used.
- Sound2.wav will be a melody sentence whose fine structure has to be used.
- Change num_filter_bank and obtain the result.

For Simulation Study 3:

- No change needs to be done.
- Sound1.wav will be a ITD sentence pointing towards the left, used for envelope.
- Sound2.wav will be a ITD sentence pointing towards the right, used for fine structure.
- Change num_filter_bank and obtain the result.

For part a, both sentences were the same while in part b, both sentences were different. Sound1.wav and sound2.wav can be interchanged to see the effect of fine structure on sound localization.
