CS342 Neural Network Class Project on Neuroevolution

Title: Acoustic Signal Source Localization using backpropagation network

Yul Young Park, Dec. 05 2004

1 Goal: To achieve acoustic signal source localization capability using 2 receivers.

2 Overall System:

The most conventional source localization methods utilize an array receiver and adaptive filtering method with certain amount of correctness. In this project, 2 receivers and backpropagation network(supervised learing) were used to achieve the similar or even better correctness with conventional digital signal processing techniques. The current results looked unsatisfactory, but there remained several factors to be improved. In addition, originally, neuroevolution was planned to be utilized, but the supervised learning of backpropagation method took the more time expected. So, neuroevolution method will be another future work for the better result. The brief block diagram is as followings

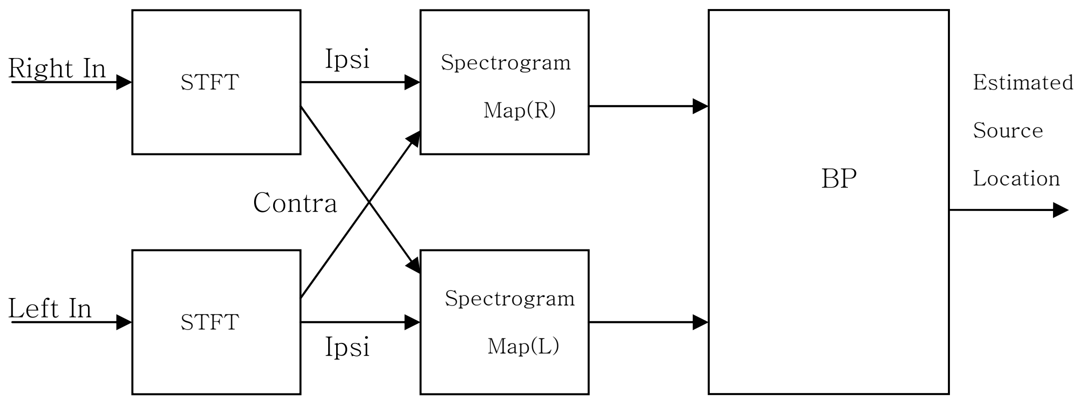


Fig.1 Overall Block Diagram

* STFT (Short-Time Fourier Transform), BP(Backpropagation)

** Biological Analogy: R/L ear -> Superior Olivary Complex -> Auditory Cortex

3 Detail Block Description:

3.1 Input signal: SigJenny software by NaTCH inc. was used to generate various frequency tone burst signals in PC, and JBL 2425J horn driver was used to generate acoustic signal. Then, 2 microphones inside the artificial head(KEMAR) were used to get the left and right ear signals separately. Finally, User build multi-channel receiver power amplifier and NI PCI-4472 multi-channel simultaneous sampling analog device card was used for data sampling and recording.

Fig. 2 Data recording Environment: (From Left)
Artificial Head, Horn Driver, and User build multi-channel amplifier

3.2 STFT: For source localization, time and frequency domain information are required at the same time. Thus, 512-point FFT was done with 51 points window. The 95% of window was overlapped (48points) to provide 72us time domain resolution and the frequency domain resolution was 86Hz (=22050Hz/256pts). About detail input signal structure, refer to Appendix 1 or run the program attached.

3.3 Spectogram map: The simple algebraic map was generated by subtracting STFT of the right ear from that of the right ear. Here, the inhibition of the other side ear was assumed. The resulting data became the input to the backpropagation network. Since, the total number of training data was too big, only 17 evenly spaced frequency points were selected for each source location case. The final combined 136x217 block were input to the backpropagation network. Thus, total 136 patterns from 8 source location categories were used to train the network.

3.4 Backpropagation network: The 2-layer backpropagation network was implemented and used. 217 input unit, 217 hidden layer unit, and 8 output unit composed the network. The least mean square learning rule, momentum=0.1, learning rate=0.9, and maximum trail number =10000 were used. As a data representation, the continuous value input data representation and bipolar distributed net output were obtained by bipolar sigmoid activation function.

4 Training Data:

4.1 Signal Type: sinusoidal toneburst of various frequency were used in training and testing. Generally, the 3 periods of sinusoidal tone burst signals were recorded at 44100Hz sampling frequency. Then, the 15.873msec duration of signals (total 700 samples) were selected from each recording. The 500Hz sine tone burst signal was used as the training signals, and 1k,1.5k,2k, and 4k tone burst signals were used as testing signal. Then, the STFT of each signal was computed and the difference between the left and right signal was input to the network(136 patterns, 217 input units)

4.2 Receiver: 2 microphone inside the artificial head was used to collect input signals

4.3 Space Configuration: 8 locations in the anechoic chamber

8 locations =>[Left, Right, Middle, Front, Back, Half, High, Low]

left,low = [1,-1,-1,-1,-l, 1,-l, 1]
left,high = [1,-1,-1,-1,-l, 1, l,-1]
right,low = [-1, 1,-1,-1,-l, 1,-l, 1]
right,high = [-1, 1,-1,-1,-l, 1, l,-1]
back,low = [-1,-1, 1,-1, l,-1,-l, 1]
back,high = [-1,-1, 1,-1, l,-1, l,-1]
front,low = [-1,-1, 1, 1,-l,-1,-l, 1]
front,high = [-1,-1, 1, 1,-l,-1, l,-1]

*** 'Middle' means b/w Left&Right, and 'Half' means b/w Front&Back. Those signals were used as redundant information on purpose to resolve the conflicting output

5 Test Data: 1k,1.5k,2k, and 4k tone burst signals were used as testing signals, and the same data processing was done on those signals before they were presented to the trained network. The total test signals are 85 signals: Left/High 4k tone burst 17 , Left/High 2k tone burst 17, Right/High 1.5k tone burst 17, Back/High 1.5k tone burst 17, and Front/High 1.5k tone burst 17.

6 Evaluation Measure: Not only conventional Adaptive filtering method results was not available, but the output itself wasn't satisfactory in given time. Thus, the network was tested only by the testing data.

7 Results: one test result data is presented. Please, refer to Appendix 3, or run the

codes attached. For Left/High location of 2k/4k sine tone burst, the distinction between Left/Right/Back/Front was unsuccessful(2/34 hits), but High/Low distinction gave good success(30/34). However, for Right/High case, the hit rate was reverse. The hit rate of Left/Right/Back/Front was 17/17 , but High/Low hit rate was 0/17., For Back/High 1.5k tone burst case, the results looked be put in the middle. the hit rate of Left/Right/Back/Front was 4/17, and the hit rate for High/Low distinction was 5/17. Finally, for Front/High 1.5k tone burst case, the hit rate of Left/Right/Back/Front was 11/17, but the hit rate for High/Low distinction gave 13/17 success. The Middle and Half output were used to compensate the confliction between Left/Right or Front/Back.

8 Discussion:

In some test case, the High/Low distinction could be done successfully, and the other case, the vice versa. Of course, some cases were between the two extremes. So, if the interaural time difference(ITD) and interaural level difference (ILD)were separately trained, it might reduce the above error. Then, the superior olivary complex need to be divided into tow parts: MSO(Medial SO) and LSO(Lateral SO). Then, MSO would be implemented for the ITD distinction and LSO for the ILD distinction. This might reduce the complexity because input data matrix will be divided into two for each network resulting less computation.

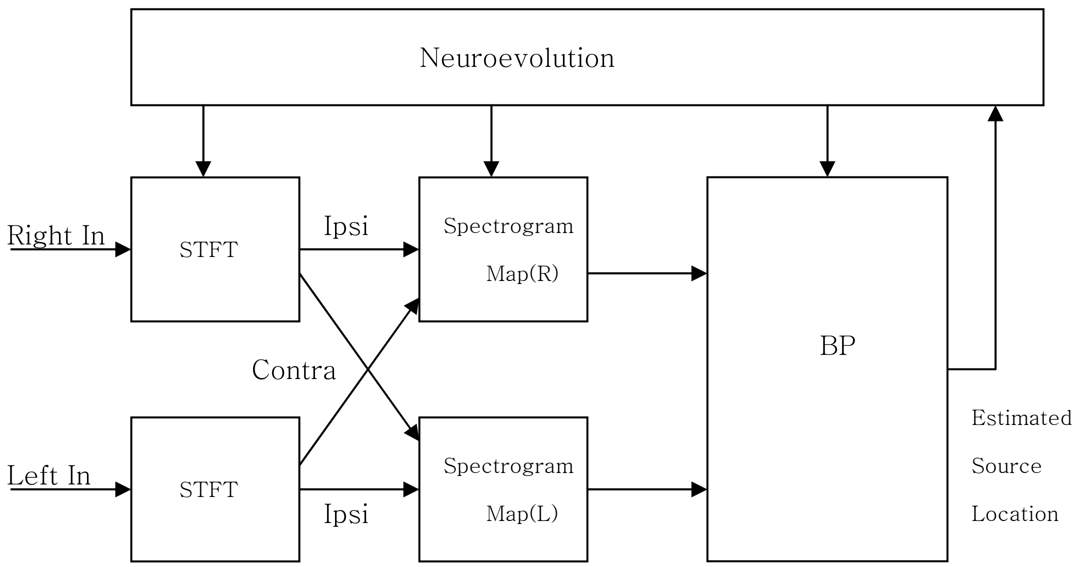
Also, since the input data was represented by continuous value, maximum MSE setting required empirical data. If a certain threshold was set for the input data like 1 for excitation, 0 for idle, and -1 for inhibition, the maximum MSE can be easily found, and this was the case when the backpropagation network was trained for XOR logic. However, this might require more consideration on the better type of activation function.

Finally, the artificial head size was 12cm, and the expected ITD of left lower source location case was about 600usec and this could be observed from the recorded data. Overall, when the cone of confusion is considered, even for the human, the source localization may be difficult. Of course, the memory and comparison with previously presented data and head movement may help to do this job easier

9 Future Works:

9.1 Data Representation: Even though the STFT which has the data point evenly spaced in time and frequency domain was used here for the input data transformation, the other type of input data transformation like wavelet would be more suitable for the cognitive model to reflect psychoacoustic respects. In addition, non-linear or probabilistic spectrogram map not the simple subtraction might help to give better result. For example, the spiking neuronal model would be used in the spectrogram map generation

9.2 Neuroevolution: Neuroevolutioin can be utilized to adjust various data representation parameter and the network parameters including the network architecture itself. Then, the neuroevolution controls may be considered as efferent signals



9.3 Others: Devise a method to extract the features among data which explain precedent effect, echo-canceling in real situation.

10 Special Thanks: entire data collection was done by the help of Soonkwon Paik(Ph.D Candidate in ECE Dept), and also he provided valuable comments on the data recoding techniques.

11 References:

- http://nn.cs.utexas.edu/
- Kishan Mehrotra et al, Elements of Artificial Neural Networks, 1st ed.
- Laurene Fausett, Fundamentals of Neural Networks, 1st ed.

- J. O. Pickles, An Introduction to the Physiology of Hearing, 2nd ed.
- B. C.J. Moore, An Introduction to the Ppsychology of Hearing, 5th ed.
- A.V.Oppenheim and R.W.Schafer, Discrete-time Signal Processing, 2nd ed.

Appendix 1. Input Signal Representation (Selected)

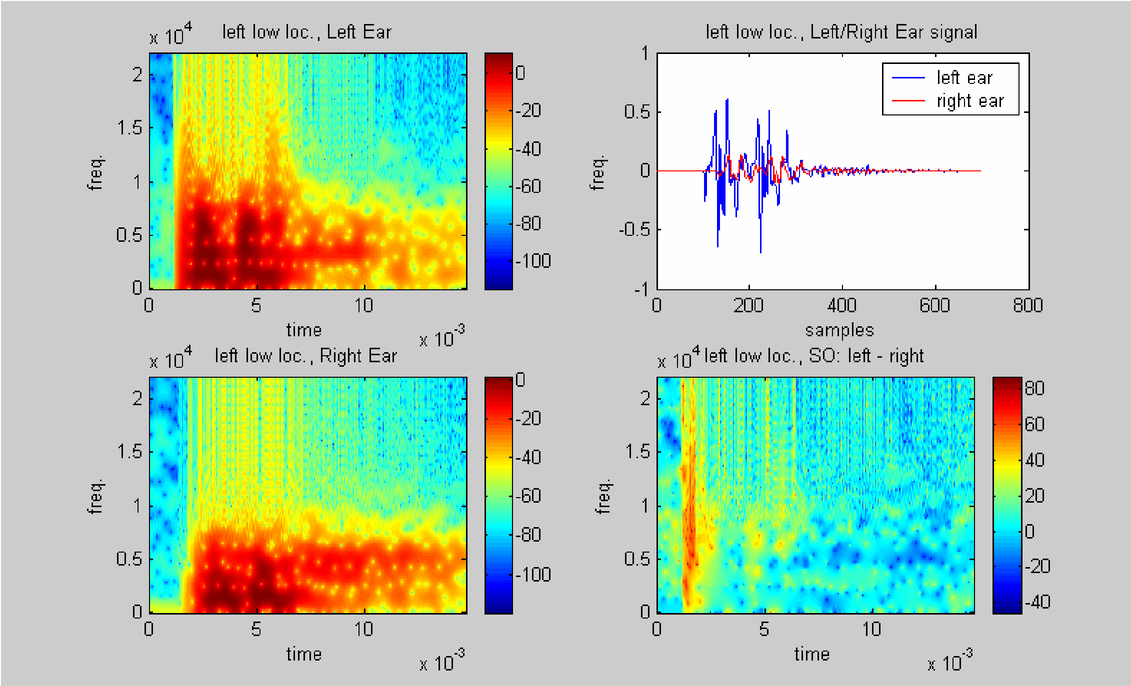


Figure A.1 input signal location = Left, Low (a)STFT of the left ear, (b) STFT of the right ear, (c) Subtraction (b) from (a), (d) time domain signal (counterclockwise)

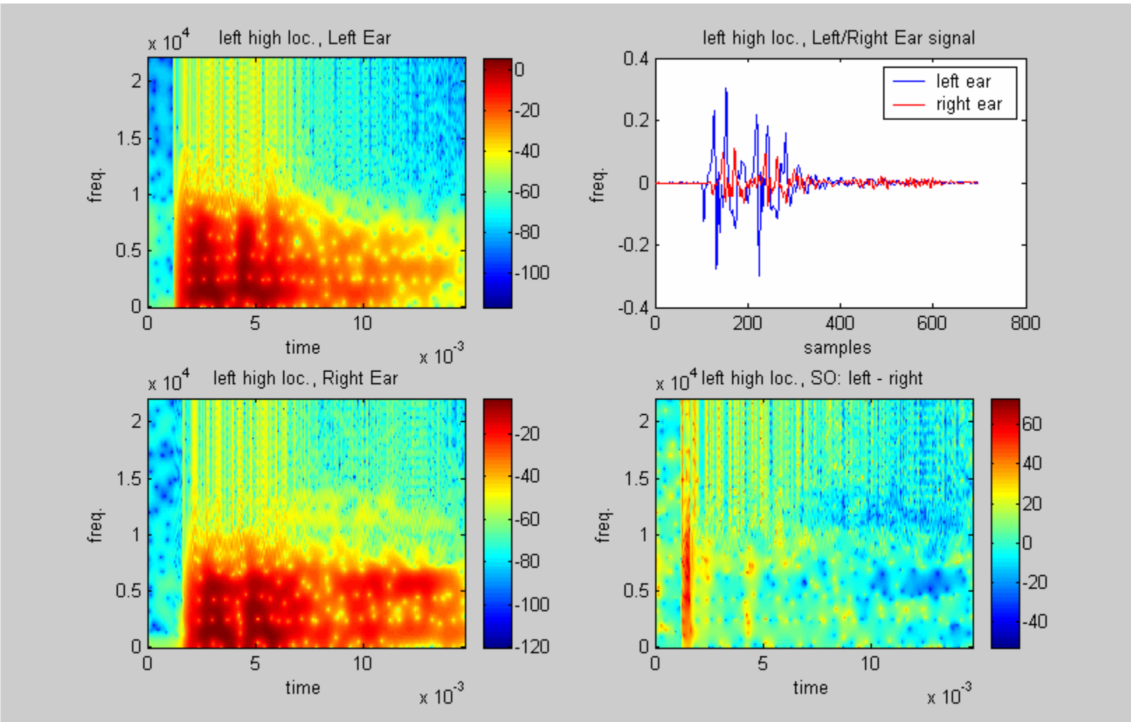


Figure A.2 input signal location = Left, High (a)STFT of the left ear, (b) STFT of the right ear, (c) Subtraction (b) from (a), (d) time domain signal (counterclockwise)

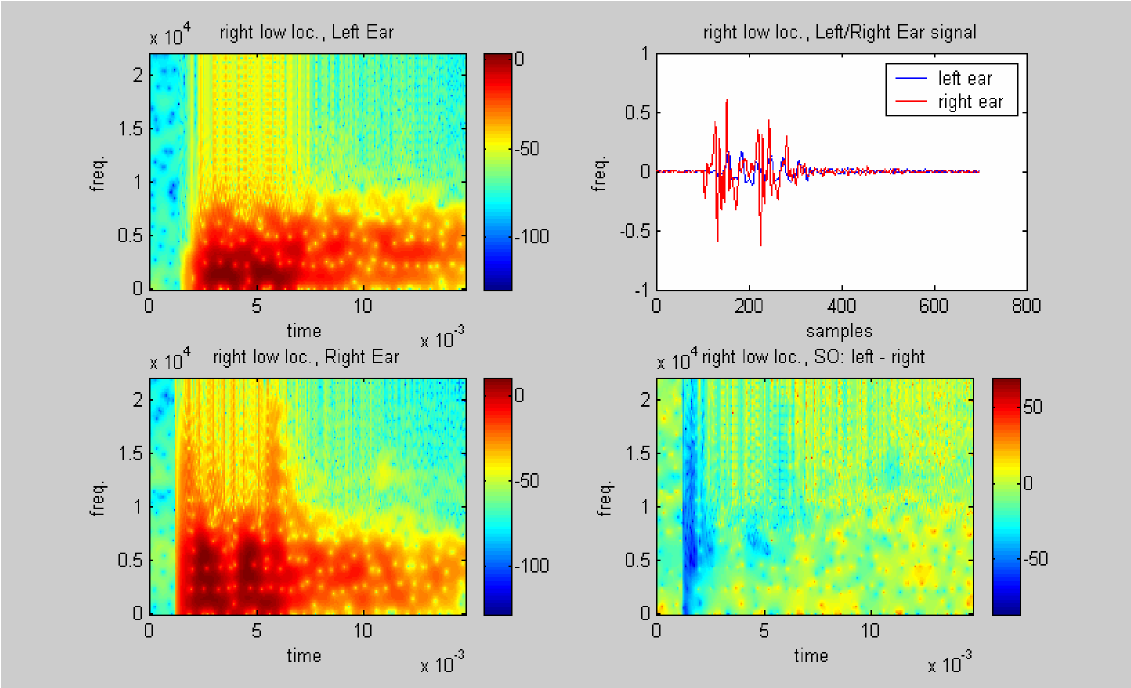


Figure A.3 input signal location = Right, Low (a)STFT of the left ear, (b) STFT of the right ear, (c) Subtraction (b) from (a), (d) time domain signal (counterclockwise)

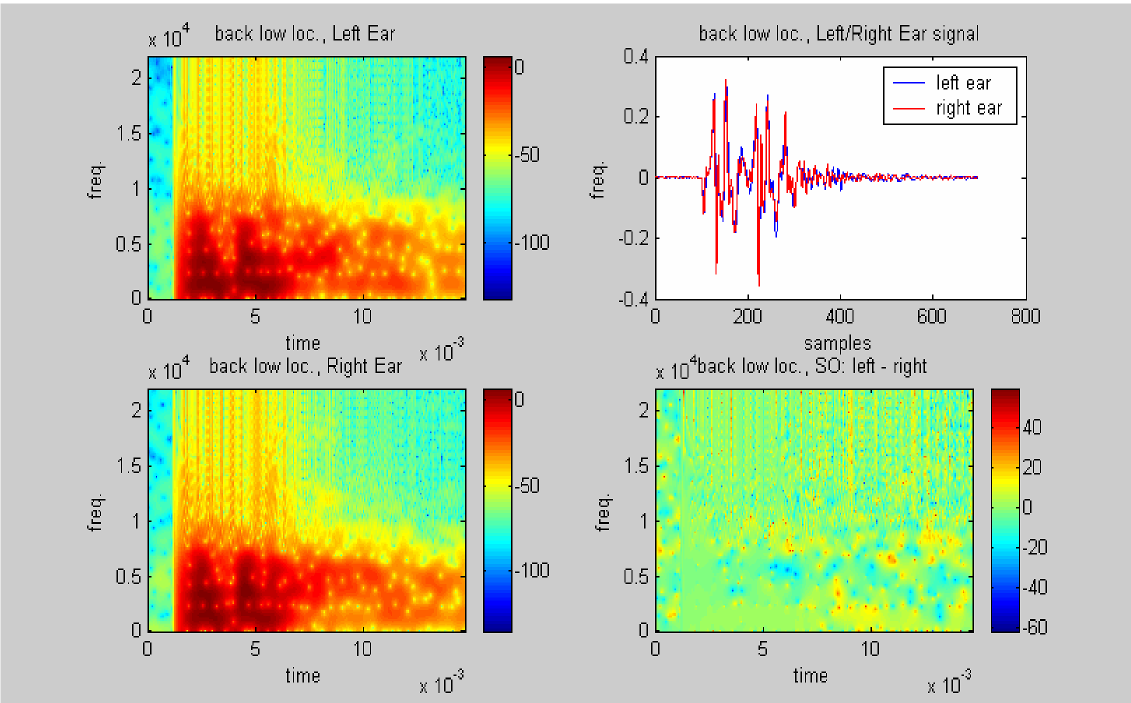


Figure A.4 input signal location = Back, Low (a)STFT of the left ear, (b) STFT of the right ear, (c) Subtraction (b) from (a), (d) time domain signal (counterclockwise)

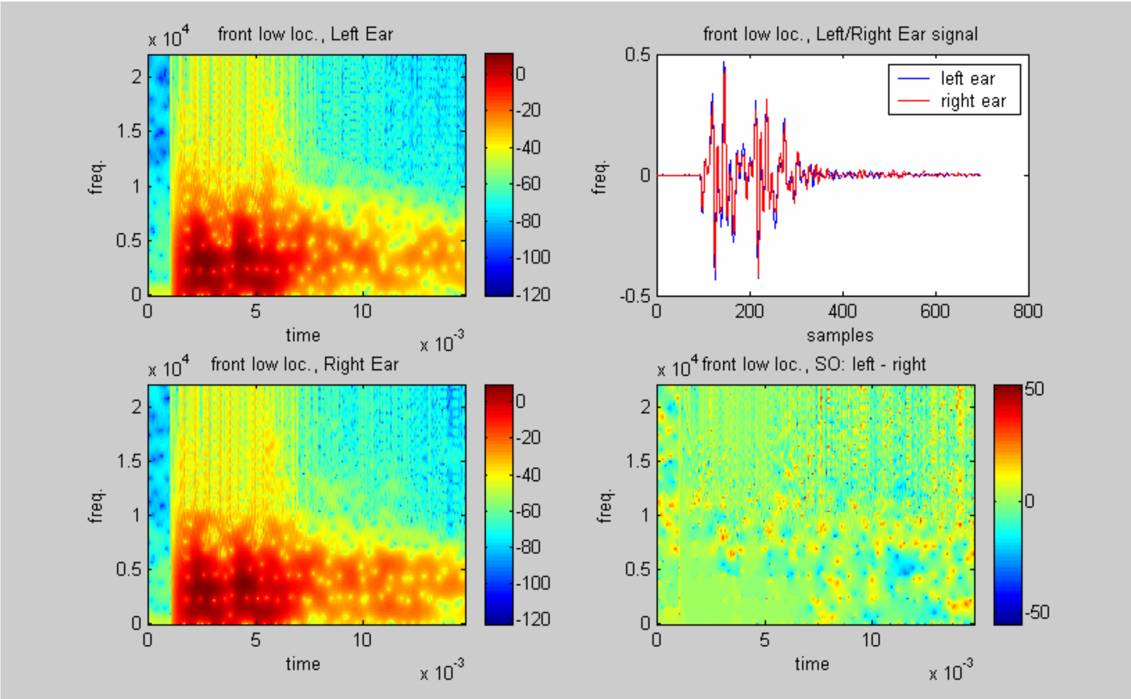


Figure A.5 input signal location = Front, Low (a)STFT of the left ear, (b) STFT of the right ear, (c) Subtraction (b) from (a), (d) time domain signal (counterclockwise)

Appendix 2. Test Signal Representation (Selected)

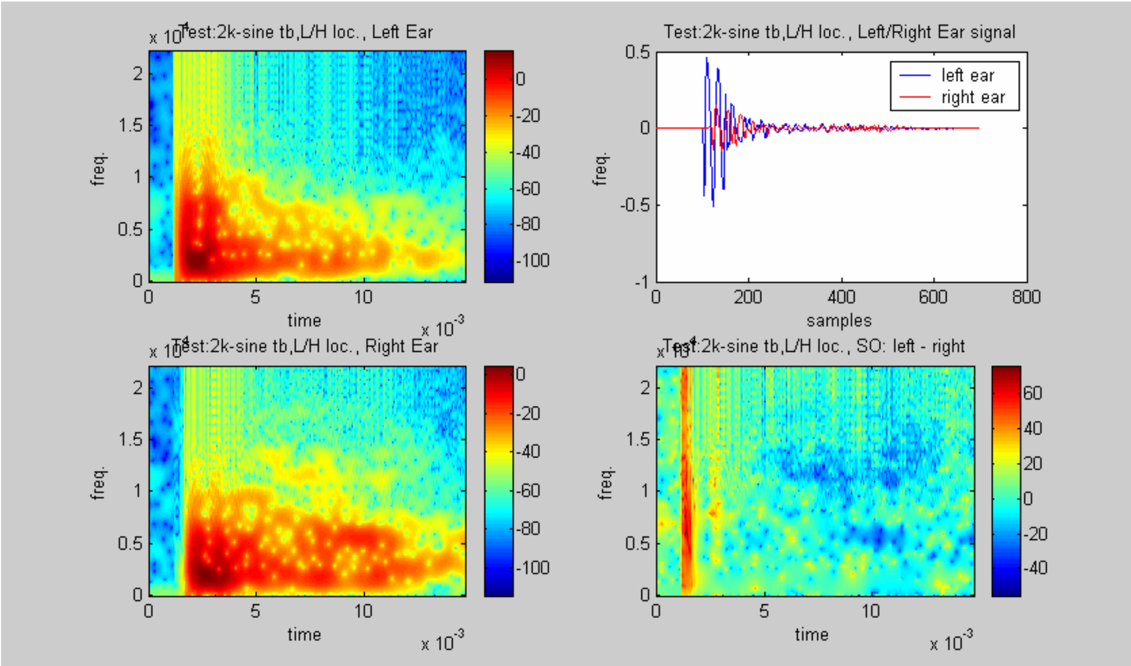


Figure A.6 test signal, 2kHz sine Left/High location (a)STFT of the left ear, (b) STFT of the right ear, (c) Subtraction (b) from (a), (d) time domain signal (counterclockwise)

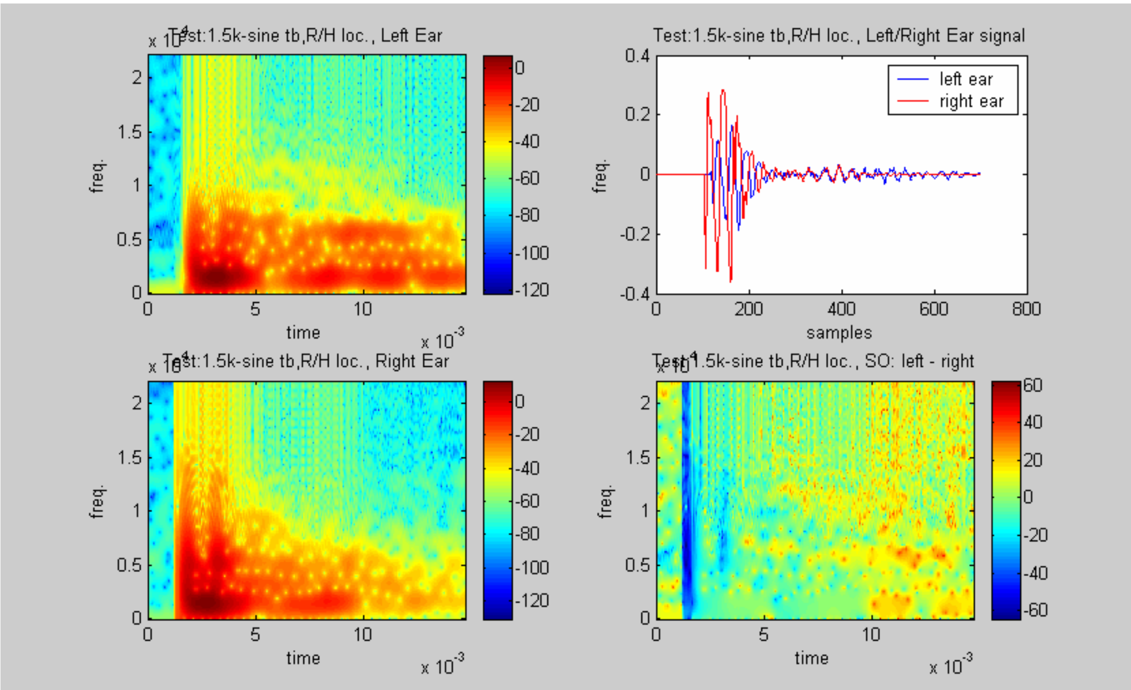


Figure A.7 test signal 1.5kHz sine, Right/High location (a)STFT of the left ear, (b) STFT of the right ear, (c) Subtraction (b) from (a), (d) time domain signal (counterclockwise)

Appendix 3. Test Result

=> test output

[Left, Right, Middle, Front, Back, Half, High, Low]

0	0	1	1	0	0	1	0
1	0	1	1	0	0	1	0
1	0	1	1	0	0	1	0
1	0	1	1	0	0	1	0
1	0	1	1	0	0	1	1
1	0	1	1	0	0	1	0
1	0	1	1	0	0	1	0
1	0	1	1	0	0	1	0
1	0	1	1	0	0	1	0
1	0	1	1	0	0	1	0
1	0	1	1	0	0	1	0
1	0	1	0	0	0	1	1
1	0	1	1	0	0	1	0
1	0	1	1	0	0	1	0
1	0	1	1	0	0	1	0
1	0	1	0	0	0	1	0
1	0	1	1	0	0	1	1
1	0	1	1	0	0	1	0
1	0	1	1	0	0	1	0
1	0	1	1	0	0	1	0
1	0	1	1	0	0	1	0
1	0	1	1	0	0	1	0
1	0	1	1	0	0	1	0
1	0	1	1	0	0	1	0
1	0	1	1	0	0	1	0
1	0	1	1	0	0	1	0
1	0	1	1	0	0	1	0
1	0	1	1	0	0	1	1
1	0	1	1	0	0	1	0
1	0	1	1	0	0	1	0
1	0	1	1	0	0	1	0
1	0	1	1	0	0	1	0
1	0	1	1	0	0	1	0

1	0	1	1	0	0	1	0
0	1	0	0	1	1	0	1
0	1	0	0	1	1	0	0
0	1	0	0	1	1	0	1
0	1	0	0	1	1	0	1
0	1	0	0	1	1	0	0
0	1	0	0	1	1	0	1
0	1	0	0	1	1	0	1
0	1	0	0	1	1	0	1
0	1	0	0	1	1	0	1
0	1	0	0	1	1	0	1
0	1	0	0	1	1	0	1
0	1	0	0	1	1	0	1
0	1	0	0	1	1	0	0
0	1	0	1	1	1	0	0
0	1	0	0	1	1	0	0
0	1	0	0	1	1	0	0
0	1	0	1	1	1	0	0
1	0	1	1	0	0	1	0
1	0	1	1	0	0	1	0
1	0	1	1	0	0	0	1
1	1	0	1	0	1	0	0
1	1	0	0	1	1	0	1
0	0	1	0	1	0	1	0
1	0	1	1	0	0	0	0
1	1	0	1	1	1	0	0
0	1	0	1	0	0	1	1
0	1	0	0	1	1	0	1
0	0	0	0	1	0	0	1
0	0	0	0	1	0	0	1
1	0	1	0	1	0	1	0
1	0	1	1	0	0	0	0
1	0	1	1	0	0	0	0
1	1	0	1	1	1	1	0
1	0	1	1	0	0	0	1
1	1	1	1	1	1	1	0

0	0	1	1	0	0	1	0
0	0	1	1	0	0	1	0
0	0	1	1	0	0	1	0
1	0	0	0	0	0	0	1
1	0	1	1	0	0	1	0
0	0	1	1	0	0	1	0
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	0	1
1	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
1	0	1	1	0	0	1	0
1	0	1	1	0	0	1	0
0	0	1	1	1	0	1	0
1	0	1	1	0	0	1	0
0	0	1	1	0	0	1	0

=> expected output

[Left, Right, Middle, Front, Back, Half, High, Low]

1	0	0	0	0	1	1	0
1	0	0	0	0	1	1	0
1	0	0	0	0	1	1	0
1	0	0	0	0	1	1	0
1	0	0	0	0	1	1	0
1	0	0	0	0	1	1	0
1	0	0	0	0	1	1	0
1	0	0	0	0	1	1	0
1	0	0	0	0	1	1	0
1	0	0	0	0	1	1	0
1	0	0	0	0	1	1	0
1	0	0	0	0	1	1	0
1	0	0	0	0	1	1	0
1	0	0	0	0	1	1	0
1	0	0	0	0	1	1	0
1	0	0	0	0	1	1	0

1	0	0	0	0	1	1	0
1	0	0	0	0	1	1	0
1	0	0	0	0	1	1	0
1	0	0	0	0	1	1	0
1	0	0	0	0	1	1	0
1	0	0	0	0	1	1	0
1	0	0	0	0	1	1	0
1	0	0	0	0	1	1	0
1	0	0	0	0	1	1	0
1	0	0	0	0	1	1	0
1	0	0	0	0	1	1	0
1	0	0	0	0	1	1	0
1	0	0	0	0	1	1	0
1	0	0	0	0	1	1	0
1	0	0	0	0	1	1	0
1	0	0	0	0	1	1	0
1	0	0	0	0	1	1	0
1	0	0	0	0	1	1	0
0	1	0	0	0	1	1	0
0	1	0	0	0	1	1	0
0	1	0	0	0	1	1	0
0	1	0	0	0	1	1	0
0	1	0	0	0	1	1	0
0	1	0	0	0	1	1	0
0	1	0	0	0	1	1	0
0	1	0	0	0	1	1	0
0	1	0	0	0	1	1	0
0	1	0	0	0	1	1	0
0	1	0	0	0	1	1	0
0	1	0	0	0	1	1	0
0	1	0	0	0	1	1	0
0	1	0	0	0	1	1	0
0	1	0	0	0	1	1	0
0	1	0	0	0	1	1	0
0	1	0	0	0	1	1	0
0	0	1	0	1	0	1	0

0	0	1	0	1	0	1	0
0	0	1	0	1	0	1	0
0	0	1	0	1	0	1	0
0	0	1	0	1	0	1	0
0	0	1	0	1	0	1	0
0	0	1	0	1	0	1	0
0	0	1	0	1	0	1	0
0	0	1	0	1	0	1	0
0	0	1	0	1	0	1	0
0	0	1	0	1	0	1	0
0	0	1	0	1	0	1	0
0	0	1	0	1	0	1	0
0	0	1	0	1	0	1	0
0	0	1	0	1	0	1	0
0	0	1	0	1	0	1	0
0	0	1	0	1	0	1	0
0	0	1	1	0	0	1	0
0	0	1	1	0	0	1	0
0	0	1	1	0	0	1	0
0	0	1	1	0	0	1	0
0	0	1	1	0	0	1	0
0	0	1	1	0	0	1	0
0	0	1	1	0	0	1	0
0	0	1	1	0	0	1	0
0	0	1	1	0	0	1	0
0	0	1	1	0	0	1	0
0	0	1	1	0	0	1	0
0	0	1	1	0	0	1	0
0	0	1	1	0	0	1	0
0	0	1	1	0	0	1	0
0	0	1	1	0	0	1	0
0	0	1	1	0	0	1	0
0	0	1	1	0	0	1	0

Appendix 4. Attached Codes

Code Files:

nn_main.m: main module. Type ‘nn_main’ in the prompt. It takes for a while to finish the job.

nn_training.m: to train network

nn_test.m: to test network

bipolar_sig.m: activation function

nn_backpropagation.m: 2-layer backpropagation network

nn_plot.m: to plot various results

testdata.m: to load test data

trainingdata.m: to load training data

Data Files:

nn_result.mat: include final network weights, all train/test data, and test results

*****.mat: the recorded signal. For the detail, refer to ‘trainingdata.m’ and ‘testdata.m’