UCDAVIS
ELECTRICAL AND COMPUTER ENGINEERING

Final project: Speaker recognition system

Howard Kao [hkao@ucdavis.edu]

William Orozco [worozco@ucdavis.edu]

EEC 201 - Winter Quarter 2021

Agenda

1. Description of the project
2. Overview of the code
3. System Design
   a. Data Preprocessing
   b. Feature Extraction
   c. Clustering and codebook
4. Results

Overview of the project

Speaker recognition system

Classifications: Identification and verification

Goal: Determining who is speaking.

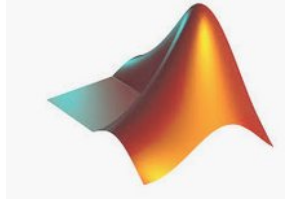
Phases: Training and testing.

Overview of the code

- Programming platform: matlab.

- **CoviDSP1.m, CoviDSP1_noise.m, CoviDSP1_notch.m:** Main script, noise added and notch. Loading files, plotting, training and testing.

- **melfb.m:** Mel filter bank.
- **mfcc_own.m:** Mel Filter Cepstrum Coefficients.
- **lbg.m:** Calculate the centroids. LBG algorithm.
- **normAudio.m:** Normalize audio.



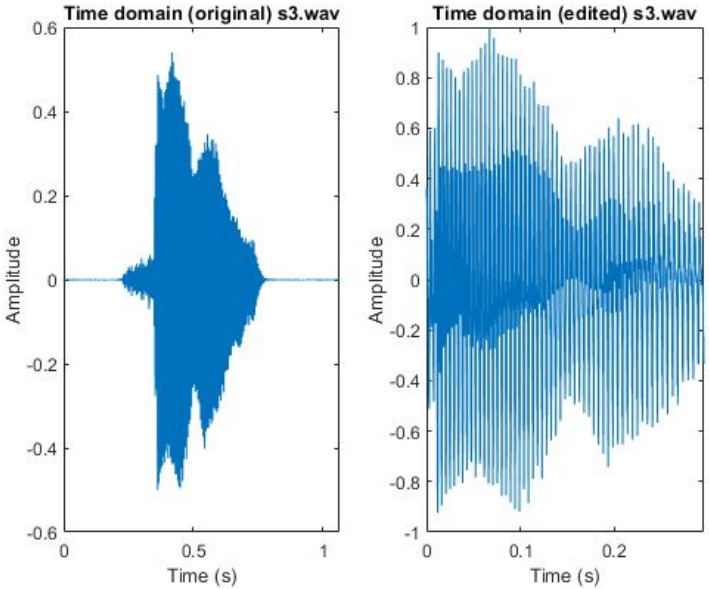
System Design

Data preprocessing

Crop quiet regions (below -10 dB), center around 0 and normalize amplitude between -1 and 1

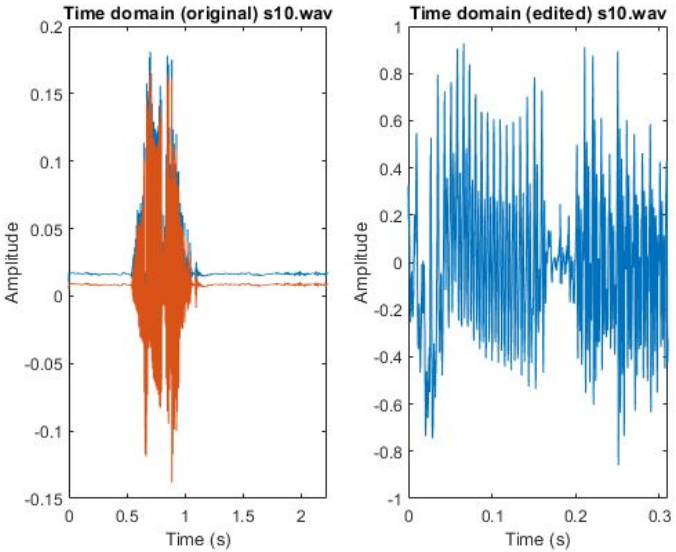
Example:

Speaker 3



Original Normalized

Speaker 10



Original Normalized

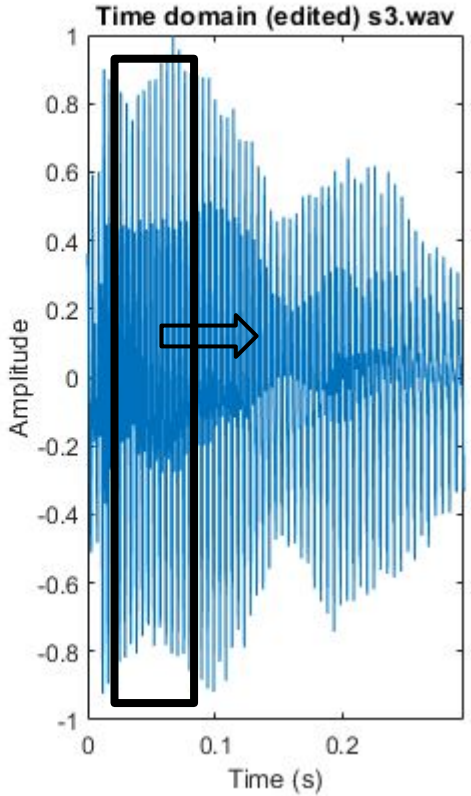
Feature extraction

Short-Time Fourier Transform

Framing, 256 samples

Overlapping, 100 samples

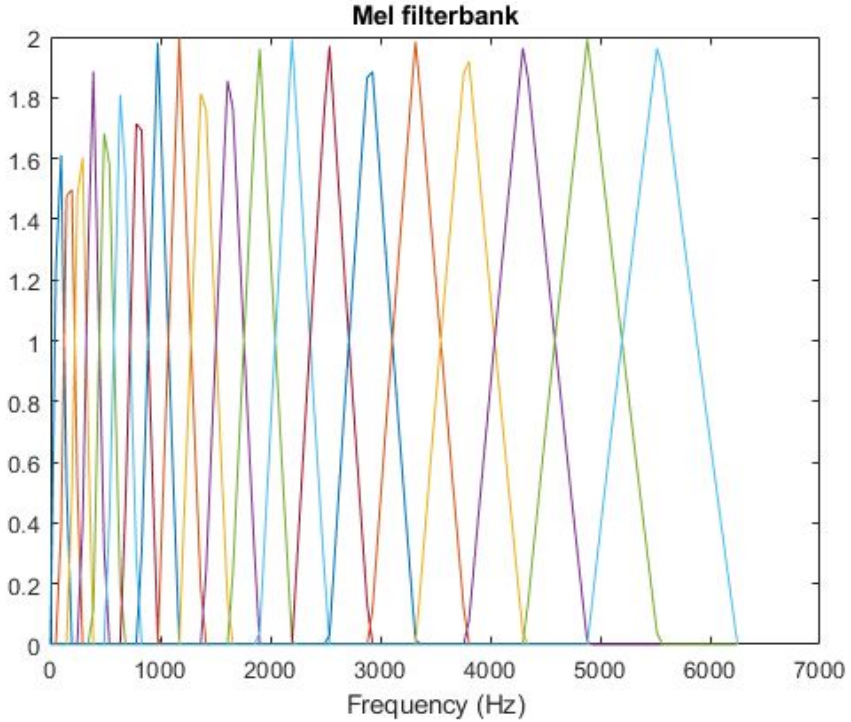
Hamming Window to minimize the spectral distortion



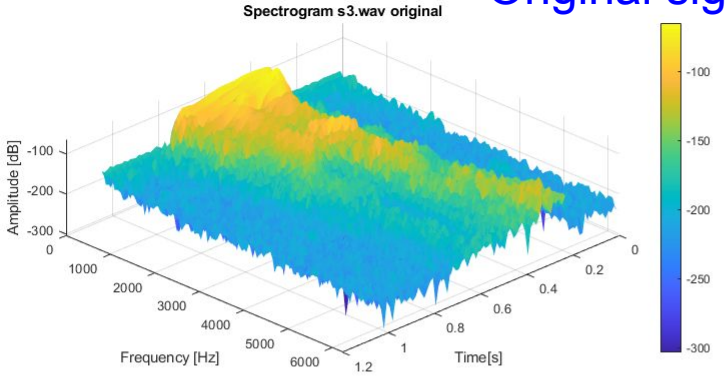
Mel Frequency wrapping

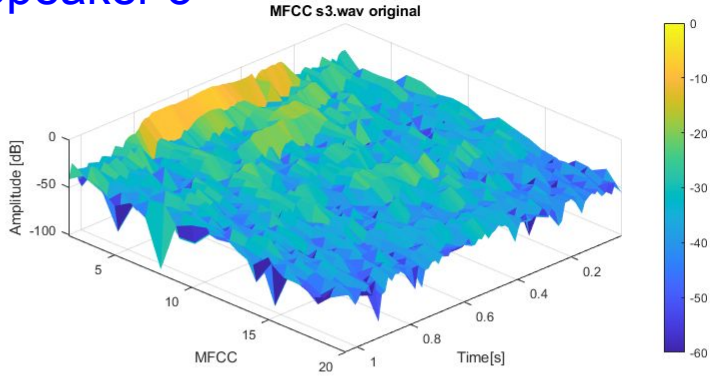
Filter bank size 20

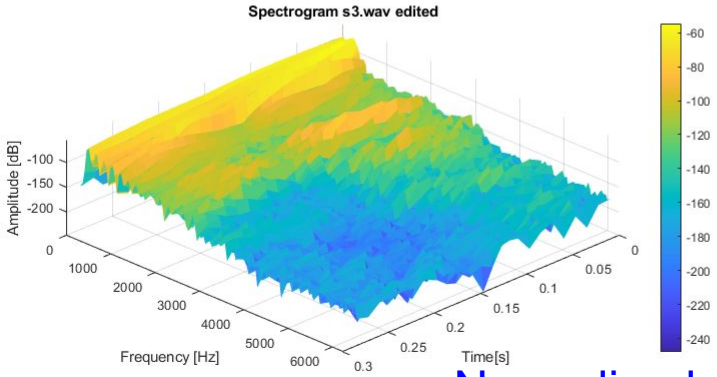
Drop first DCT coefficient

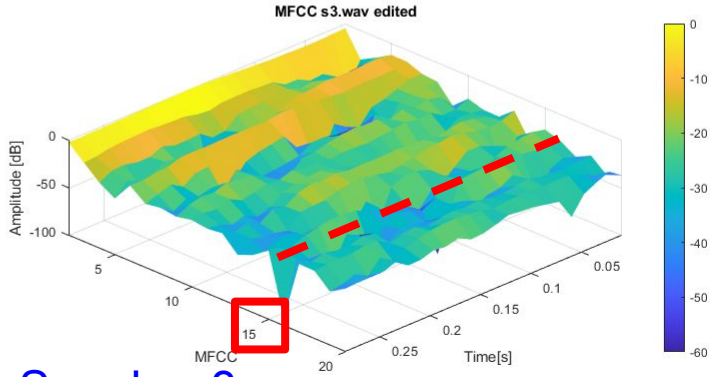


Original signal, Speaker 3



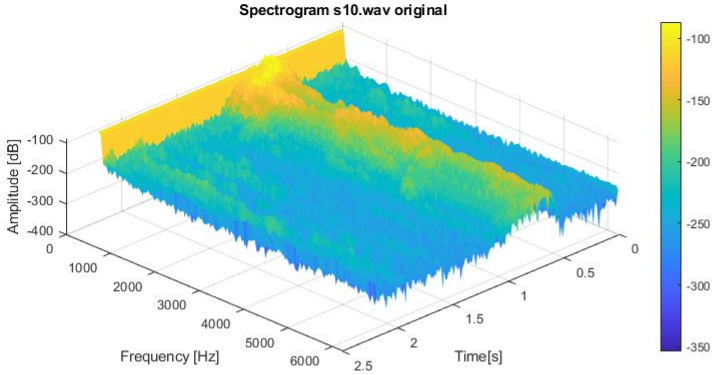


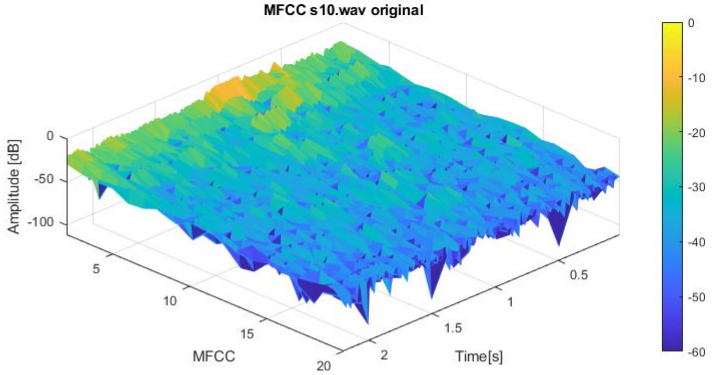


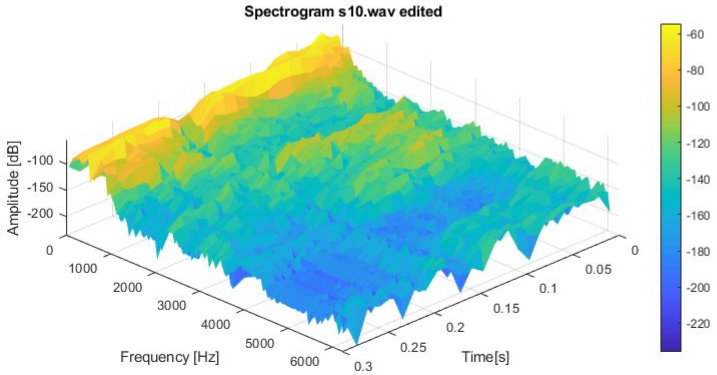


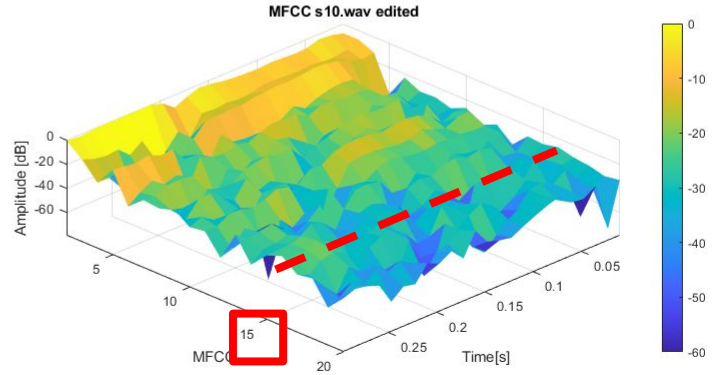
Normalized signal, Speaker 3

Original signal, Speaker 10









Normalized signal, Speaker 10

Clustering and codewords

epsilon (splitting parameter) = 0.01

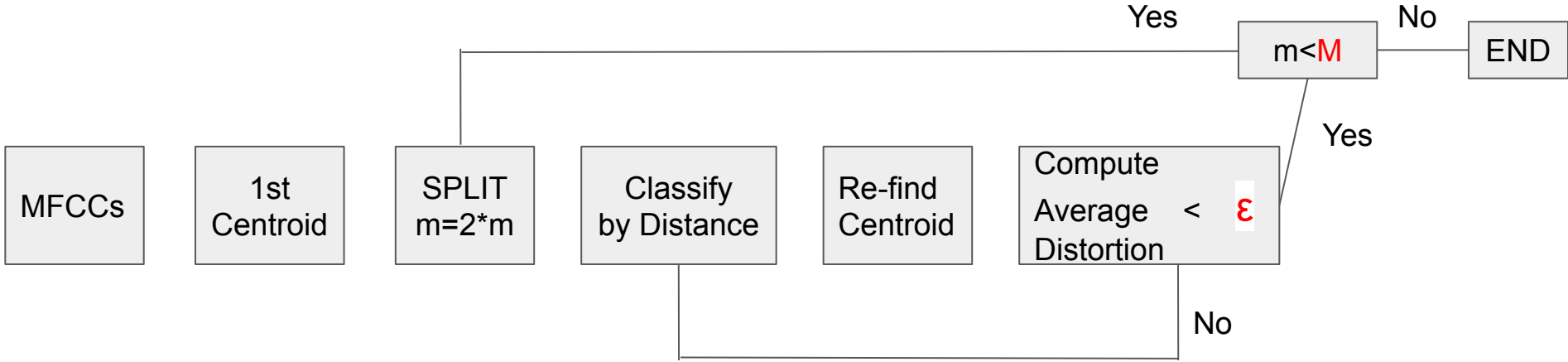
error threshold (distortion) = 0.001

distance criteria = euclidean norm (L2).

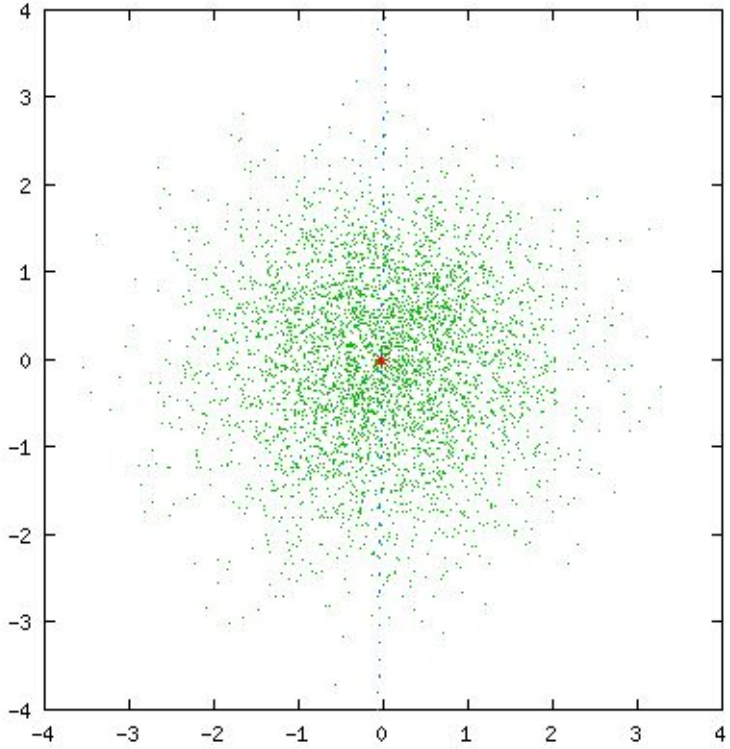
Linde Buzo Gray (LBG) algorithm

- Pick M clusters
- Decide an error threshold (ε) value - the largest distance between the data points and the centroid in a cluster.
- Decide the splitting parameter (step size)

Codebook - Data points that represent a speaker.
Clusters - Different areas of similar data points in a codebook.
Centroid - The center of a cluster.

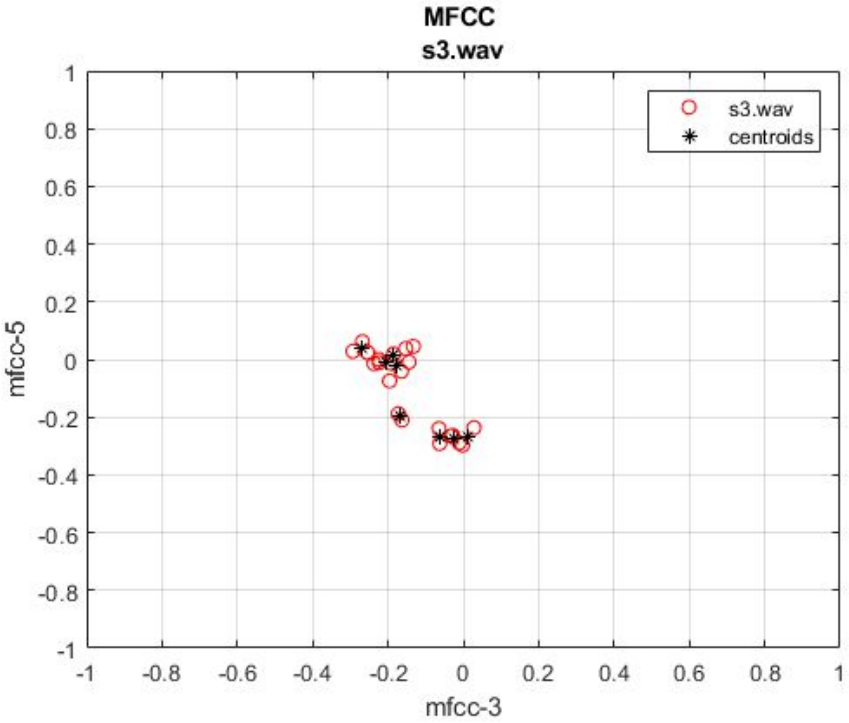


Example of LBG algorithm

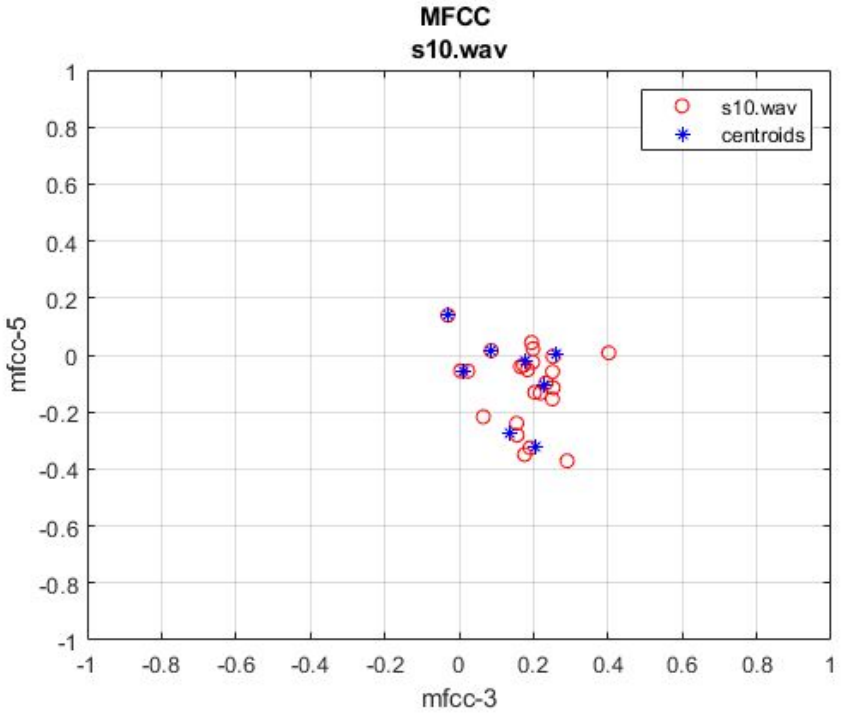


Reference: https://www.cnblogs.com/xingshansi/p/6925955.html

Speaker 3

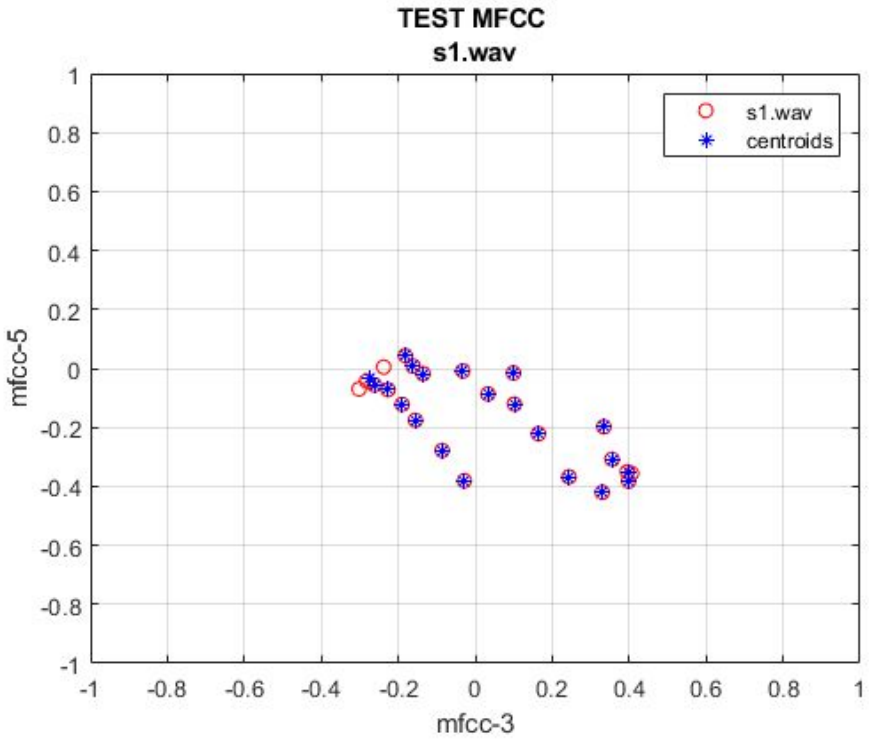


Speaker 10

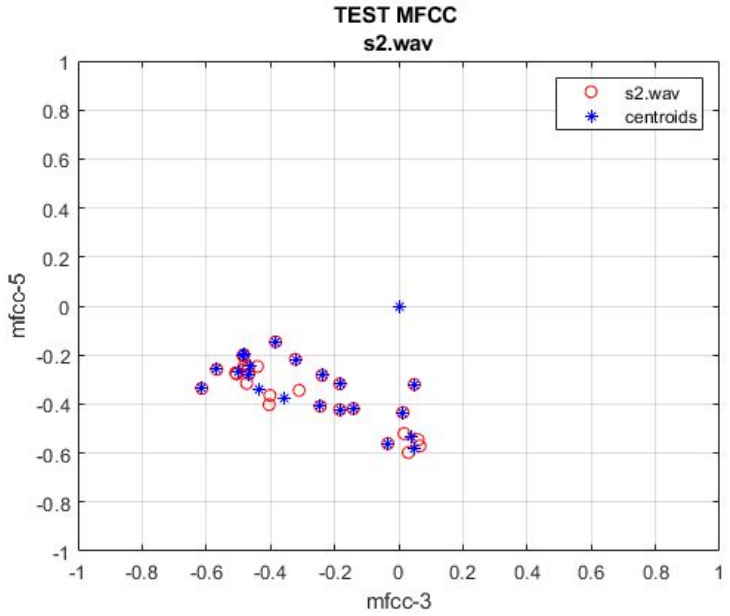


Clusters K = 8

Speaker 1



Speaker 2



Clusters K = 32

Results

Parameters:

N: Window Size.

M: Overlap

p: Size of the filterbank

K: Clusters

type_signal: Normalized or original signal

N = 256

M = 100

p = 20

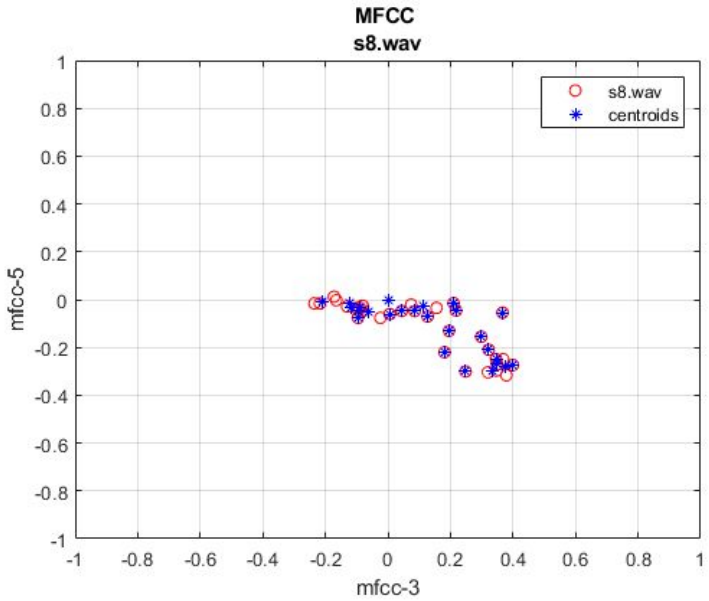
K = 32

Normalized signal

Sampling frequency 12.5 KHz

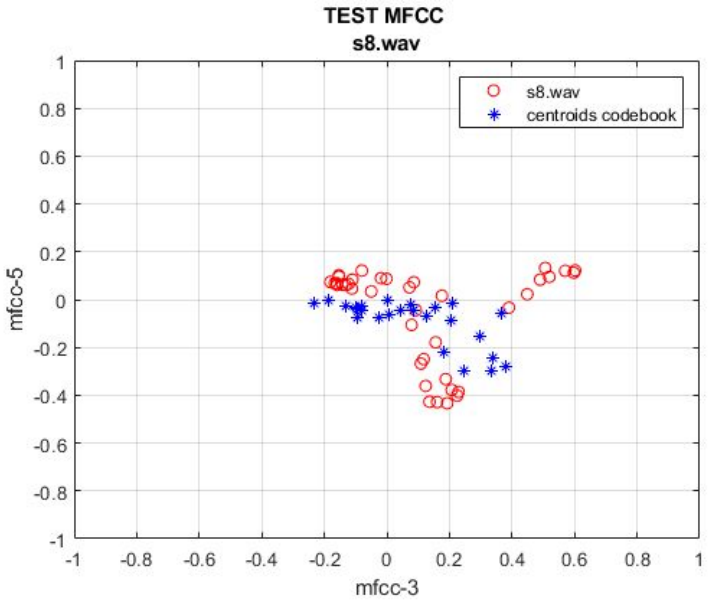
Training

Test



Speaker 8

Indexes 3 and 5



N = 256

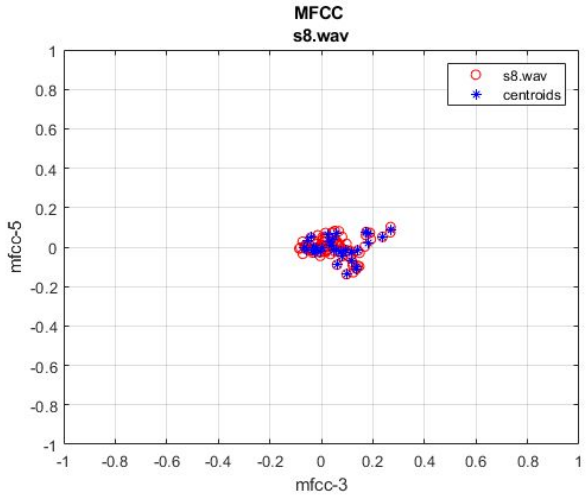
M = 100

p = 20

K = 32

Unnormalized signal

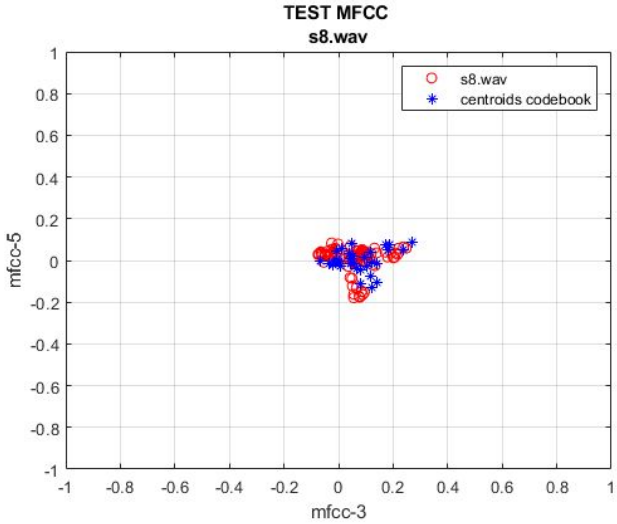
Training



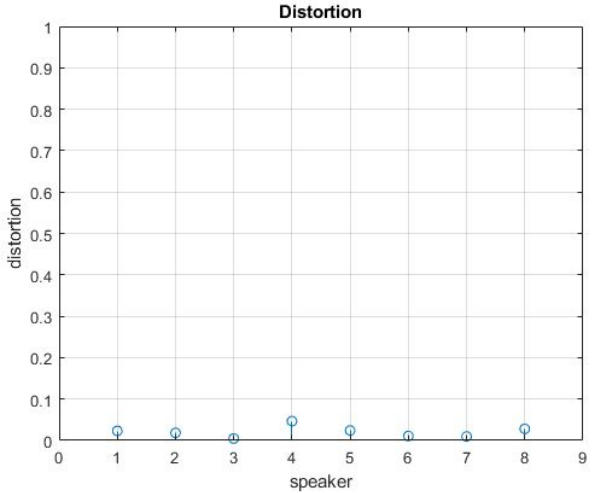
Speaker 8

Indexes 3 and 5

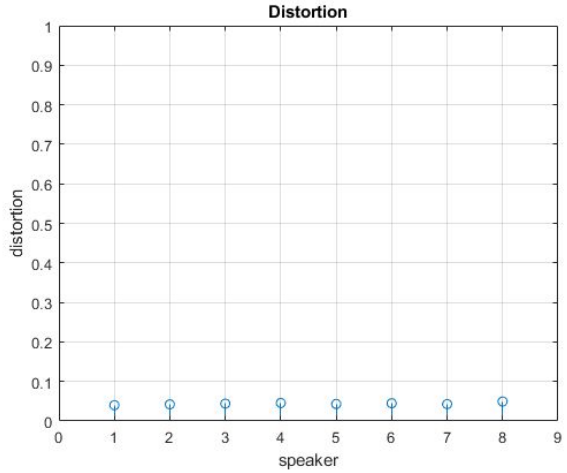
Test



Minimum error of each speaker, normalized signal



Minimum error of each speaker, unnormalized signal



Codebook error below 5%

Accuracy 100% in 8/8 speakers

Speaker	#1	#2	#3	#4	#5	#6	#7	#8
Accuracy	100%	100%	100%	100%	100%	100%	100%	100%

Let’s vary the parameters...

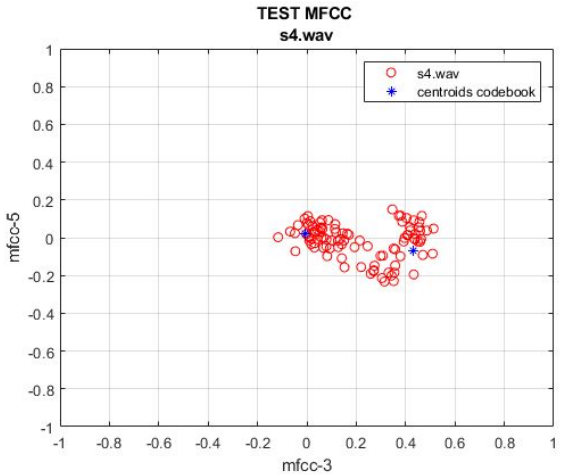
N = 100

M = 30

p = 10

K = 2

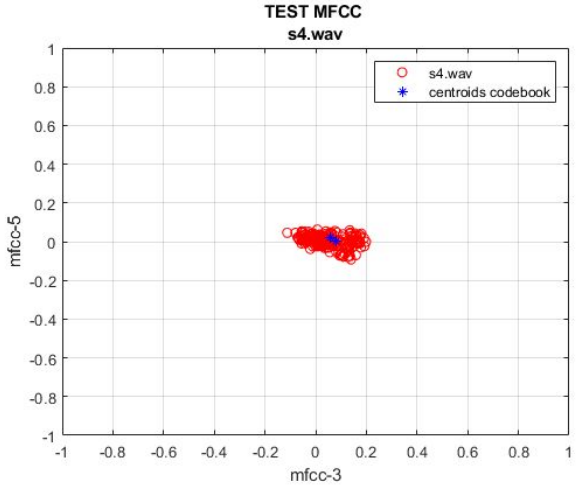
Test, normalized signal



Speaker 4

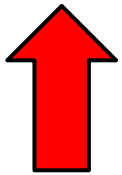
Indexes 3 and 5

Test, unnormalized signal



Accuracy, unnormalized signal. 6/8 speakers recognized.

Speaker	#1	#2	#3	#4	#5	#6	#7	#8
Accuracy	0%	100%	100%	100%	100%	0%	100%	100%



Clusters are close enough to provide false positives.

*** The normalization step is important**

Accuracy, normalized signal. 8/8 speakers recognized.

Speaker	#1	#2	#3	#4	#5	#6	#7	#8
Accuracy	100%	100%	100%	100%	100%	100%	100%	100%

Let’s add some noise...

Generate Gaussian noise using MATLAB function randn()

Speaker	S1	S2	S3	S4	S5	S6	S7	S8
Accuracy	100%	100%	100%	100%	100%	100%	100%	100%

With 0.001 as the variance of the noise (~SNR = 25 dB)

Speaker	S1	S2	S3	S4	S5	S6	S7	S8
Accuracy	100%	100%	100%	100%	100%	100%	100%	100%

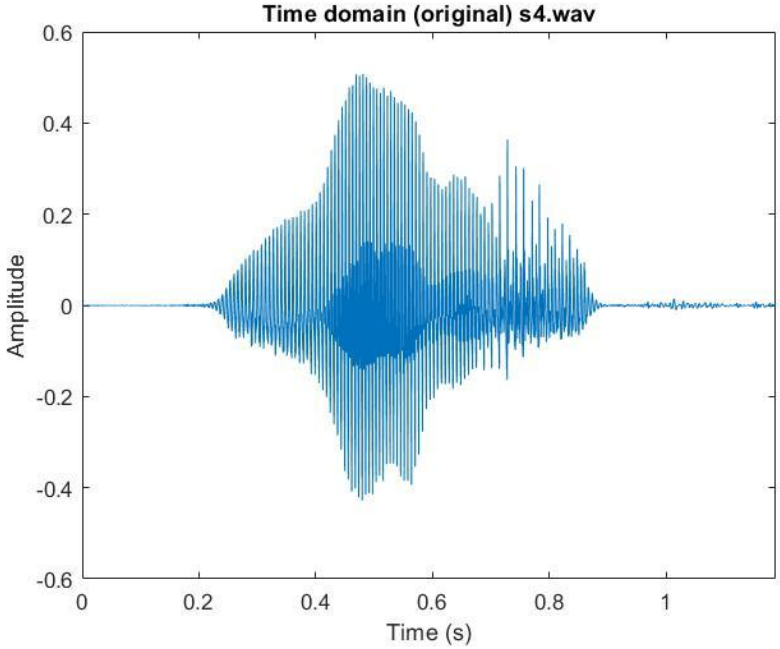
With 0.0035 as the variance of the noise (~SNR = 15 dB)

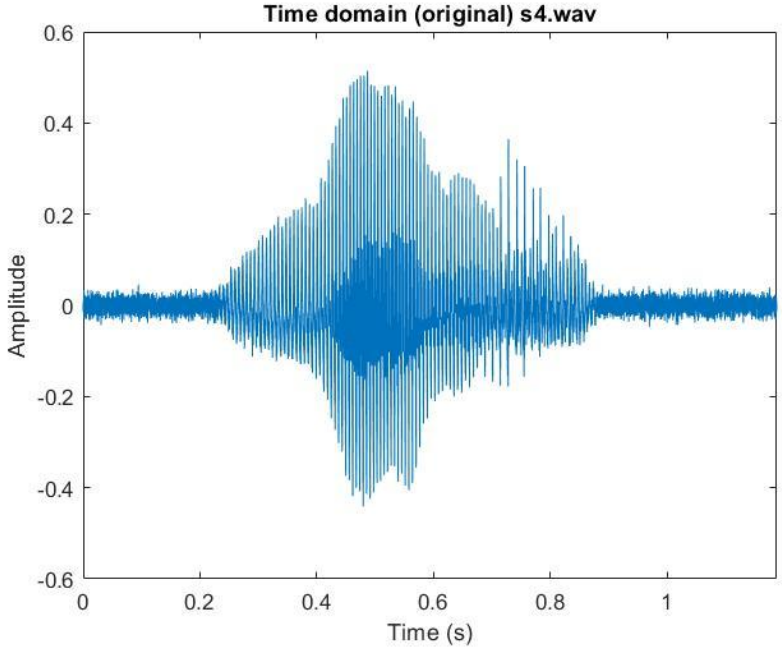
Speaker	S1	S2	S3	S4	S5	S6	S7	S8
Accuracy	100%	100%	100%	100%	100%	100%	100%	100%

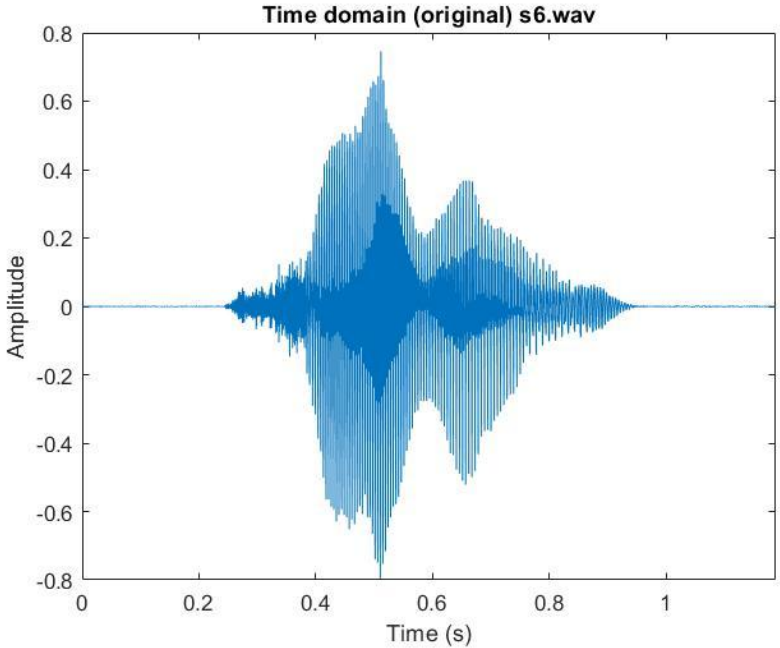
With 0.006 as the variance of the noise (~SNR = 10 dB)

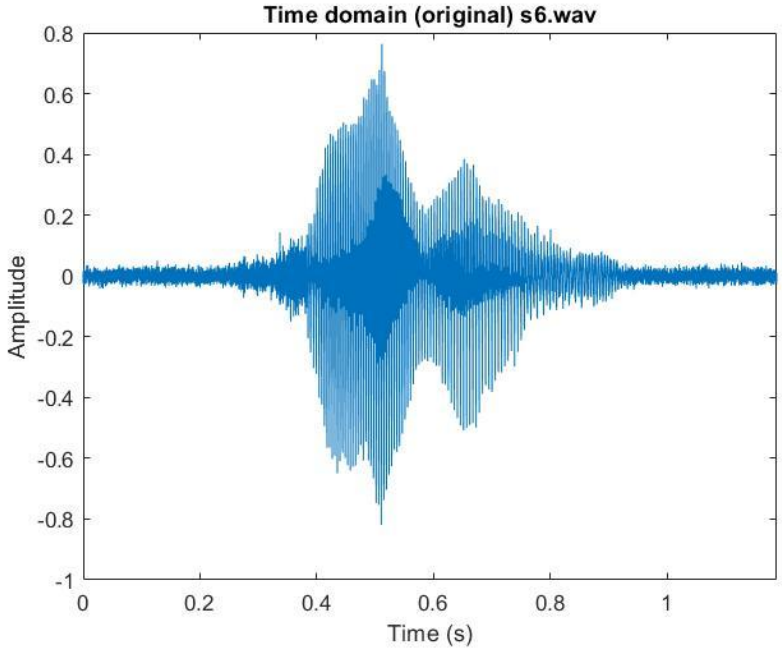
Speaker	S1	S2	S3	S4	S5	S6	S7	S8
Accuracy	100%	100%	100%	0%	100%	0%	100%	0%

With 0.013 as the variance of the noise (~SNR = 5 dB)

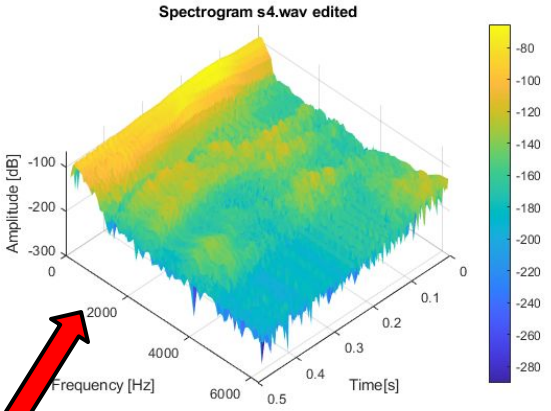

Time domain (original) s4.wav
Amplitude
Time (s)
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
0
0.2
0.4
0.6
0.8
1


Time domain (original) s4.wav
Amplitude
Time (s)
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
0
0.2
0.4
0.6
0.8
1


Time domain (original) s6.wav
Amplitude
Time (s)
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
0
0.2
0.4
0.6
0.8
1


Time domain (original) s6.wav
Amplitude
Time (s)
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
0.2
0.4
0.6
0.8
1

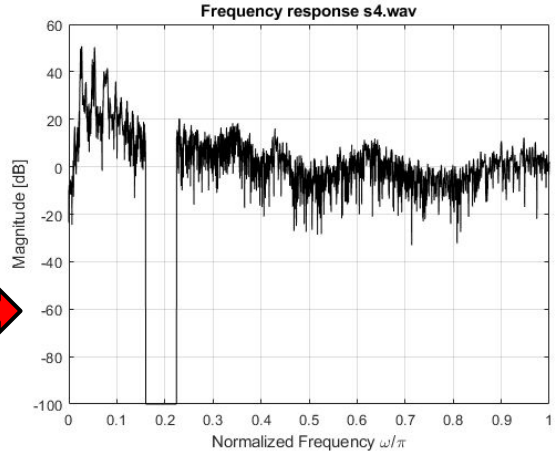
Let’s remove some frequencies...

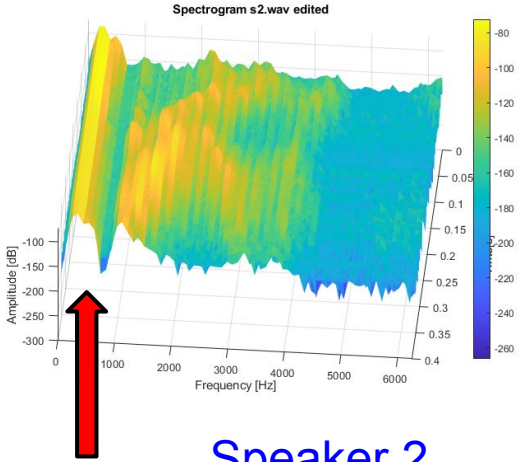


Speaker 4

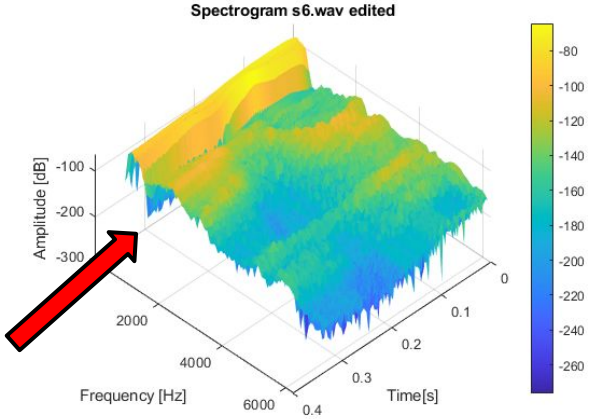
Notch filter centered at
1000 Hz, bandwidth 400 Hz

Irrelevant frequency
content was removed



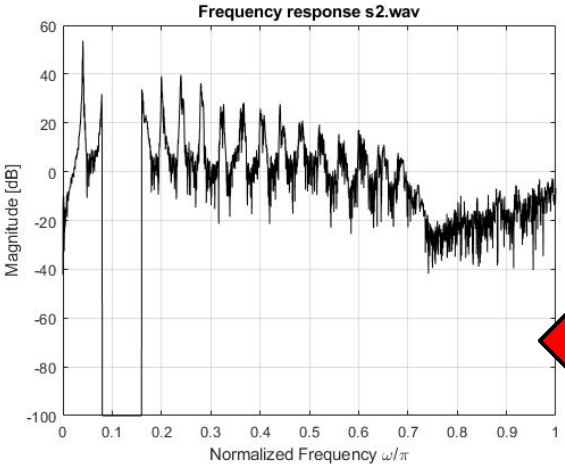


Speaker 2

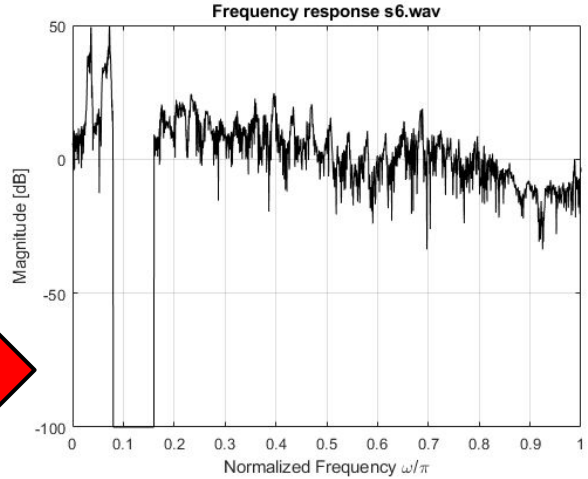


Speaker 6

Notch filter centered at 500 Hz, bandwidth 500 Hz



Relevant frequency content was removed.



Accuracy, normalized signal, notch filter centered at 1KHz, bandwidth 400 Hz. 8/8 speakers.

Speaker	#1	#2	#3	#4	#5	#6	#7	#8
Accuracy	100%	100%	100%	100%	100%	100%	100%	100%

*** The location of the frequency content of each speaker is important for recognition purposes**

Speaker	#1	#2	#3	#4	#5	#6	#7	#8
Accuracy	100%	0%	100%	100%	100%	0%	100%	100%

Accuracy, notch filter centered at 500 Hz, bandwidth 500 Hz. 6/8 speakers

Thank you