Speaker Recognition

Feature Extraction

The features of each of the training speakers must be extracted using a MFCC processor. The received speech signal must be divided into frames, windowed, then converted to the frequency domain using FFT. Mel-frequency wrapping involves processing the signal using mel-spaced filter banks that scale linearly at low frequencies and logarithmically at higher values in order to emphasize the frequencies that are important to human hearing. By doing so, cepstrum coefficients from the mel spectrum of each frame can be obtained to be later used in vector quantization.

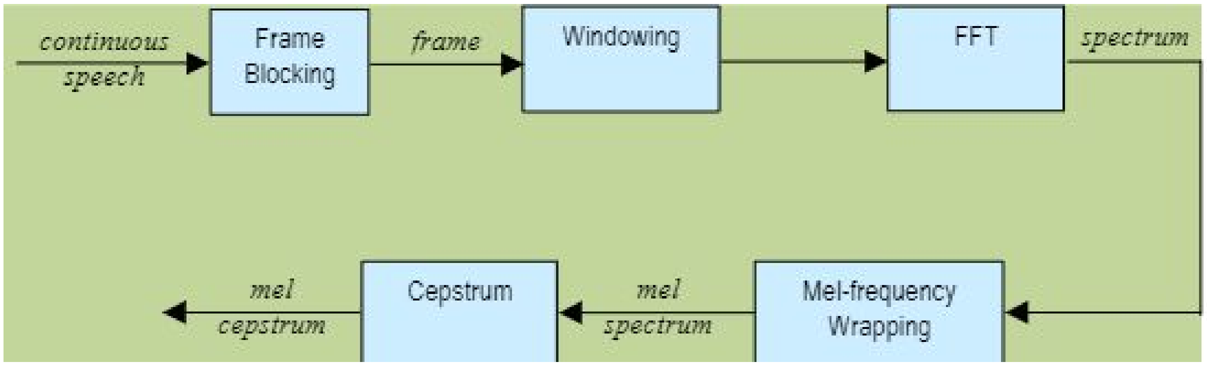


Figure 1: Block Diagram of MFCC extraction

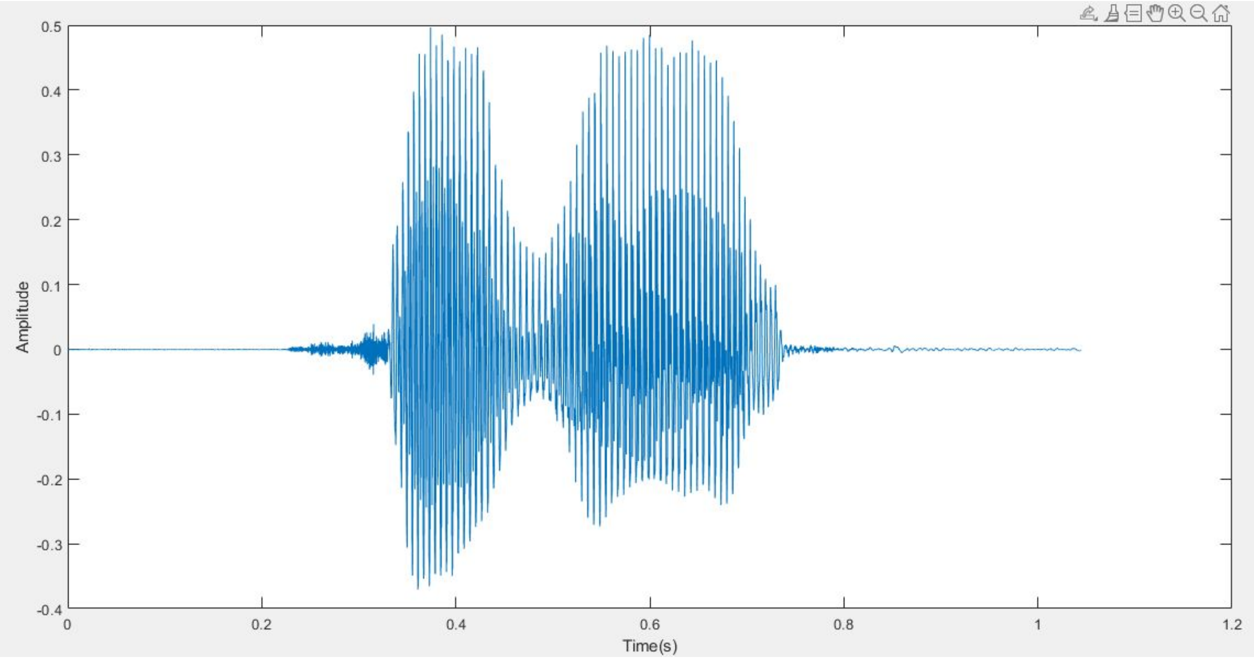


Figure 2: Speech Signal of Speaker 1 Unmodified

Frame Blocking

Each frame consists of N = 256 samples with each frame starting M = 100 samples after the other for an overlap of 156. The sampling frequency used was 12500 Hz, so each frames equates to 0.02048 seconds

Windowing

Each frame was processed through a hamming window of equal size to the number of samples in the frame to reduce spectral distortion. The hamming window used the form

$$w(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right), \quad 0 \le n \le N-1$$

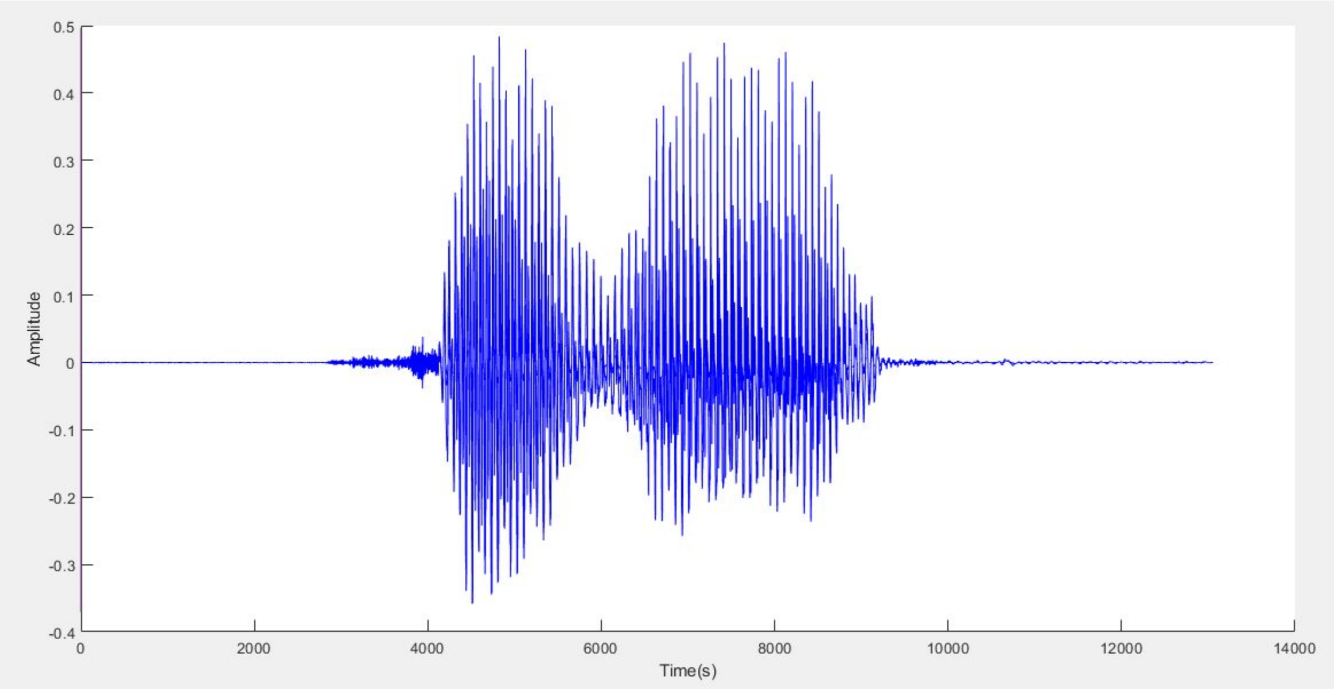


Figure 3: Speech Signal of Speaker 1 Framed and Windowed with N = 256

FFT

The FFT of the signal is taken to get the periodogram.

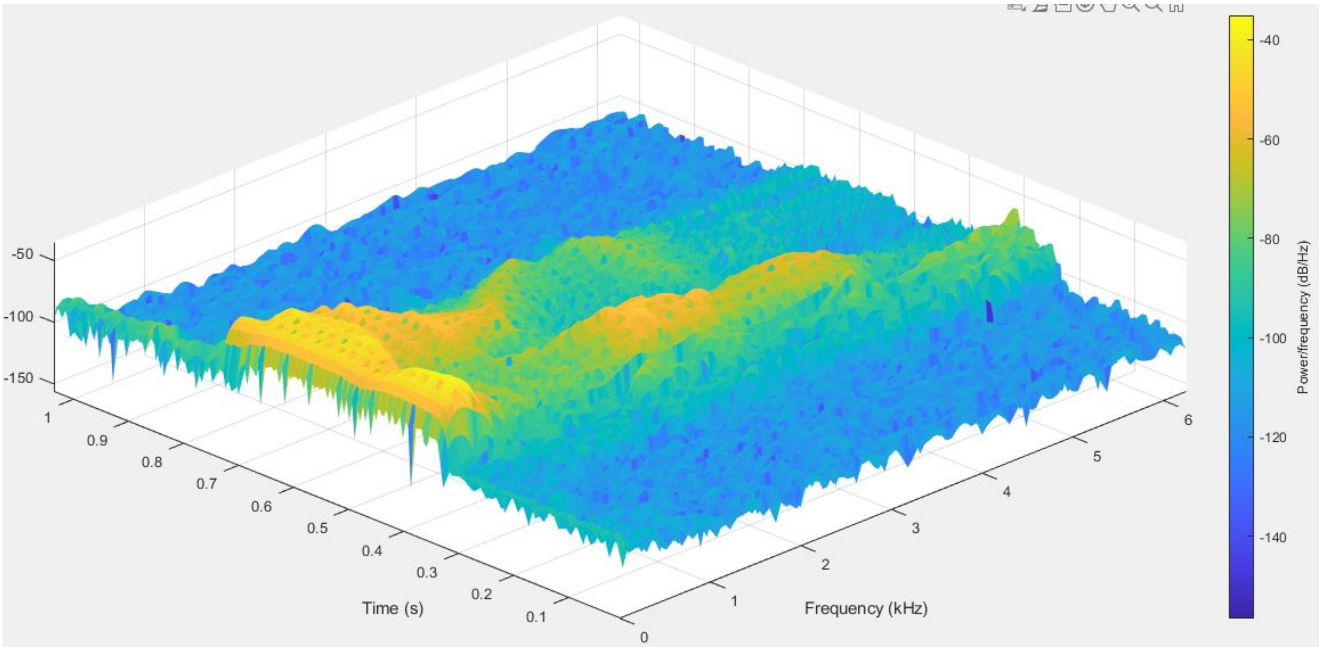


Figure 4: Periodogram of Speaker 1, N = 128, M = 42

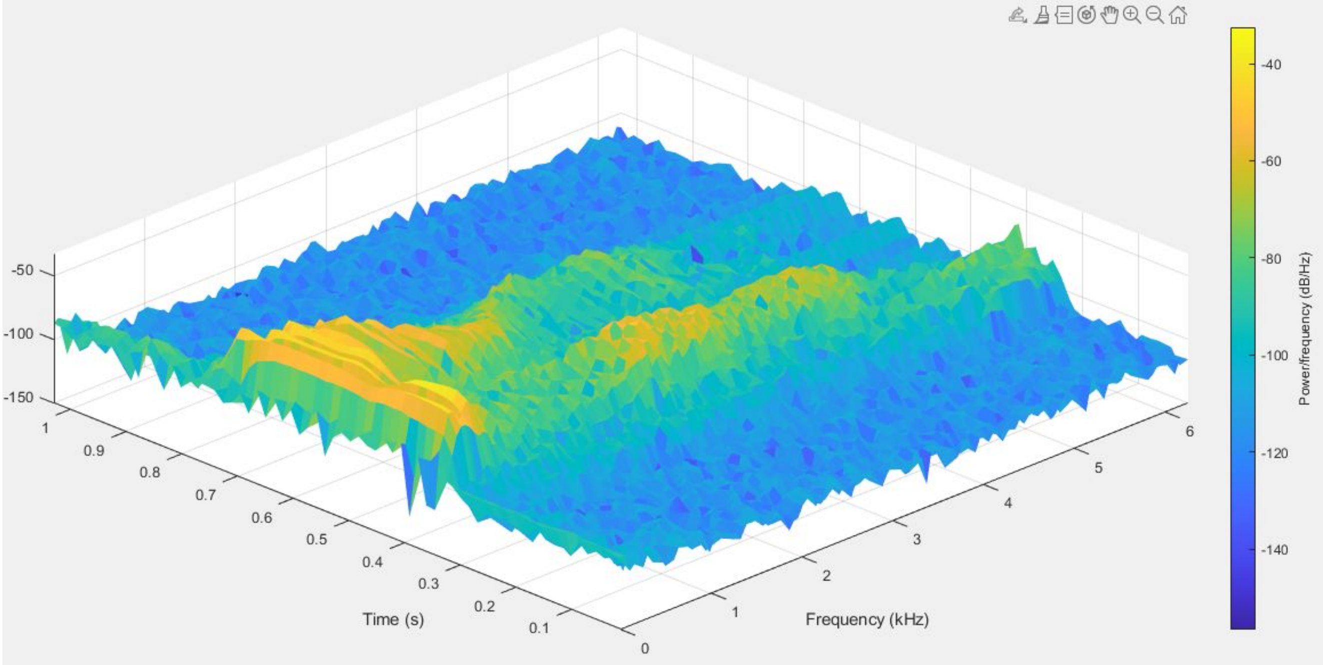


Figure 5: Periodogram of Speaker 1, N = 256, M = 85

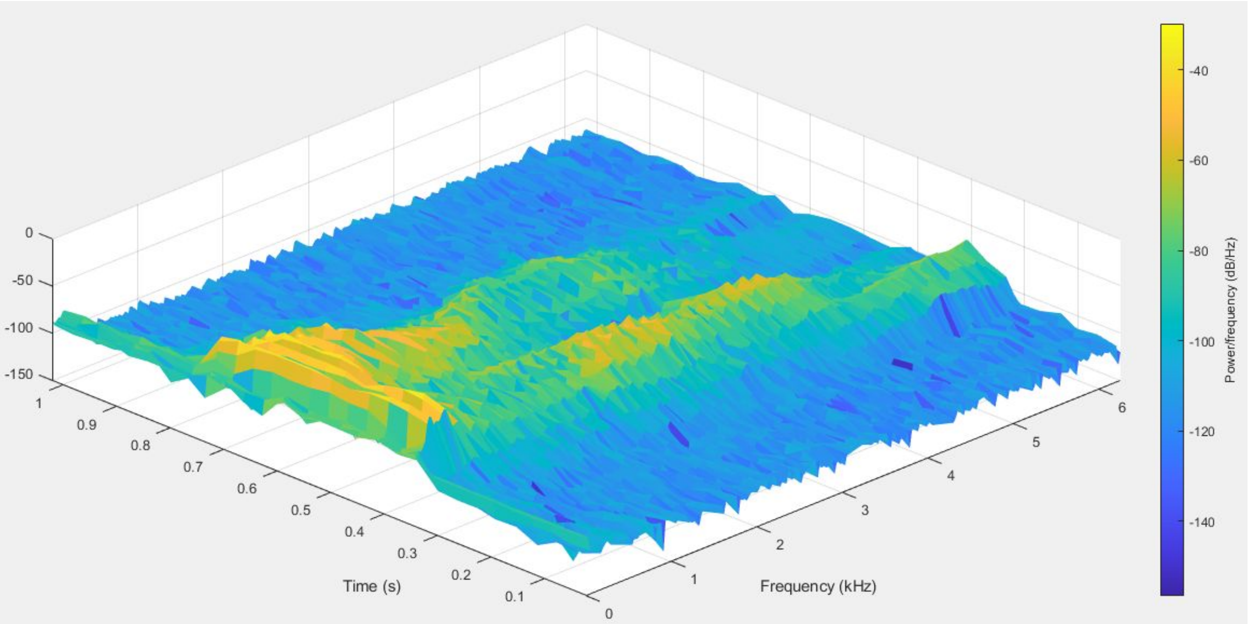


Figure 6: Periodogram of Speaker 1, N = 512, M = 170

The frequencies that contain most of the power/frequency are located below 1 kHz which reveals the dominant frequencies in human speech, thus those frequencies are emphasized by the mel-spaced filter banks.

Mel Frequency Wrapping

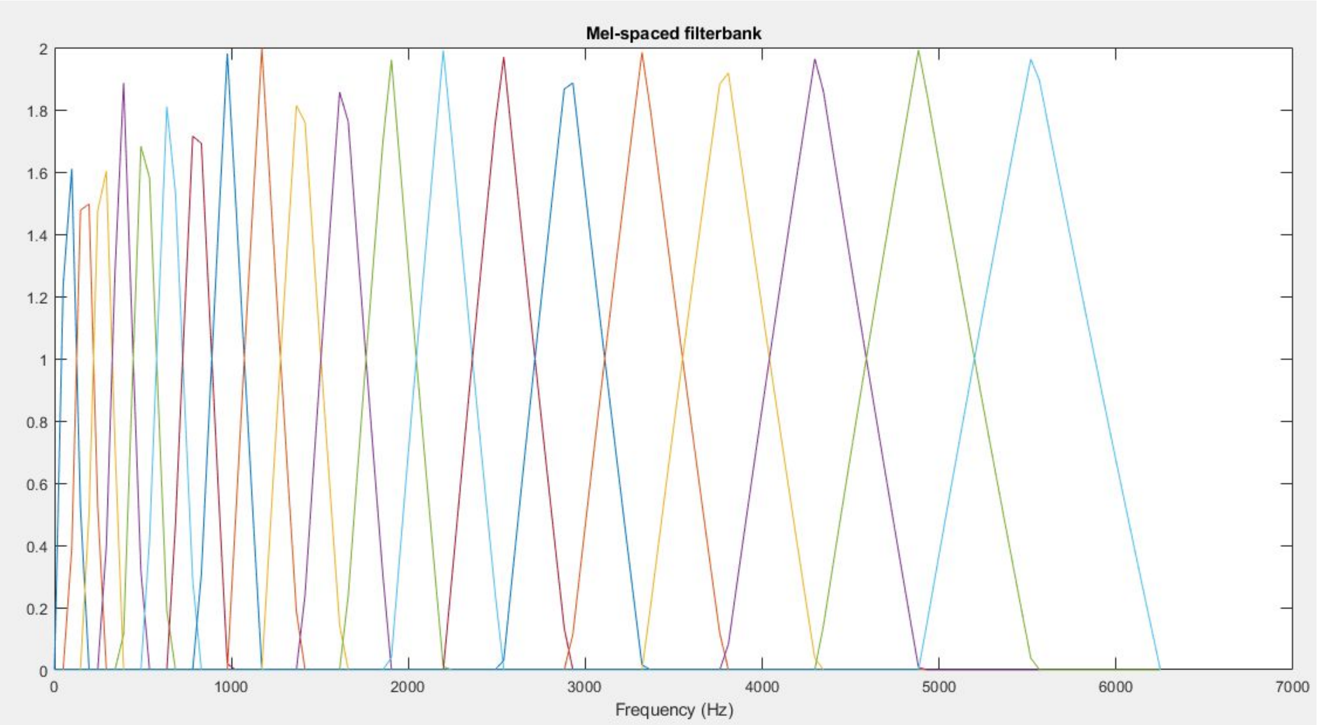


Figure 5: Mel-spaced Filter Bank Responses

20 filter banks were used resulting in 20 mel spectrum coefficients for each frame.

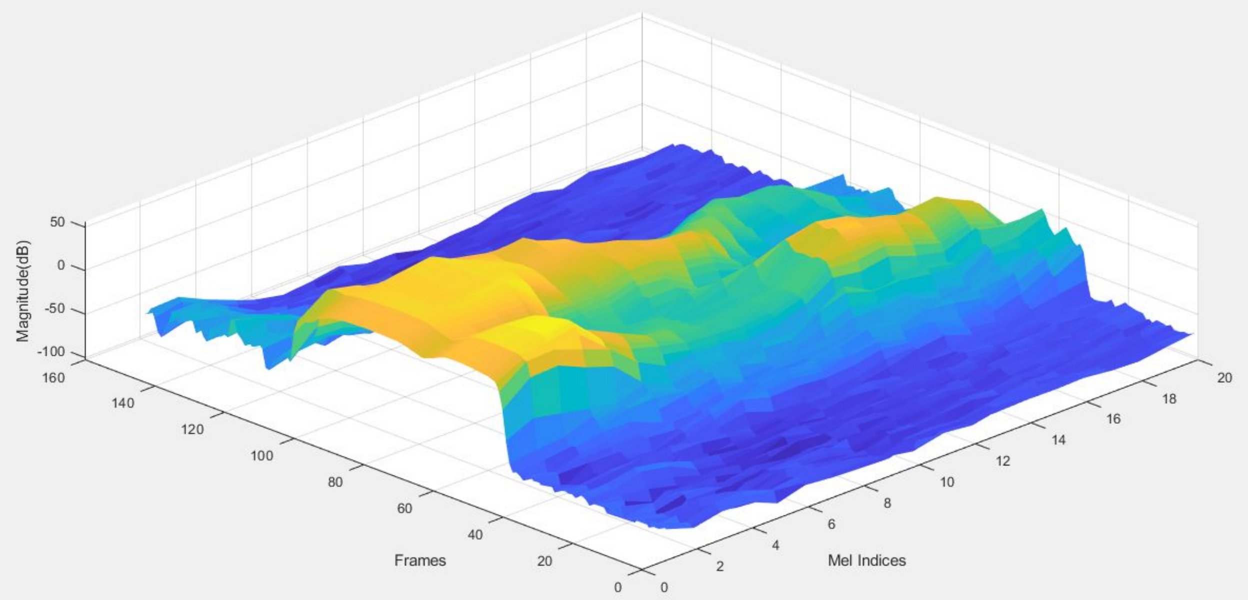


Figure : Periodogram of Speaker 1 after Mel Frequency Wrapping, N = 256, M = 85

Each filter bank produces one coefficient for a total of 20 per frame which results in a smoother periodogram.

Cepstrum

In order to turn the mel spectrum into the mel cepstrum, the DCT of the natural log of the mel spectrum was taken for the purpose of converting it to the time domain as shown below. Sk are the coefficients of the mel spectrum.

$$\tilde{c}_n = \sum_{k=1}^{K} (\log \tilde{S}_k) \cos\left[n\left(k-\frac{1}{2}\right)\frac{\pi}{K}\right], \qquad n = 0,1,\ldots,K\text{-}1$$

The result is 20 MFCC's for each frame.

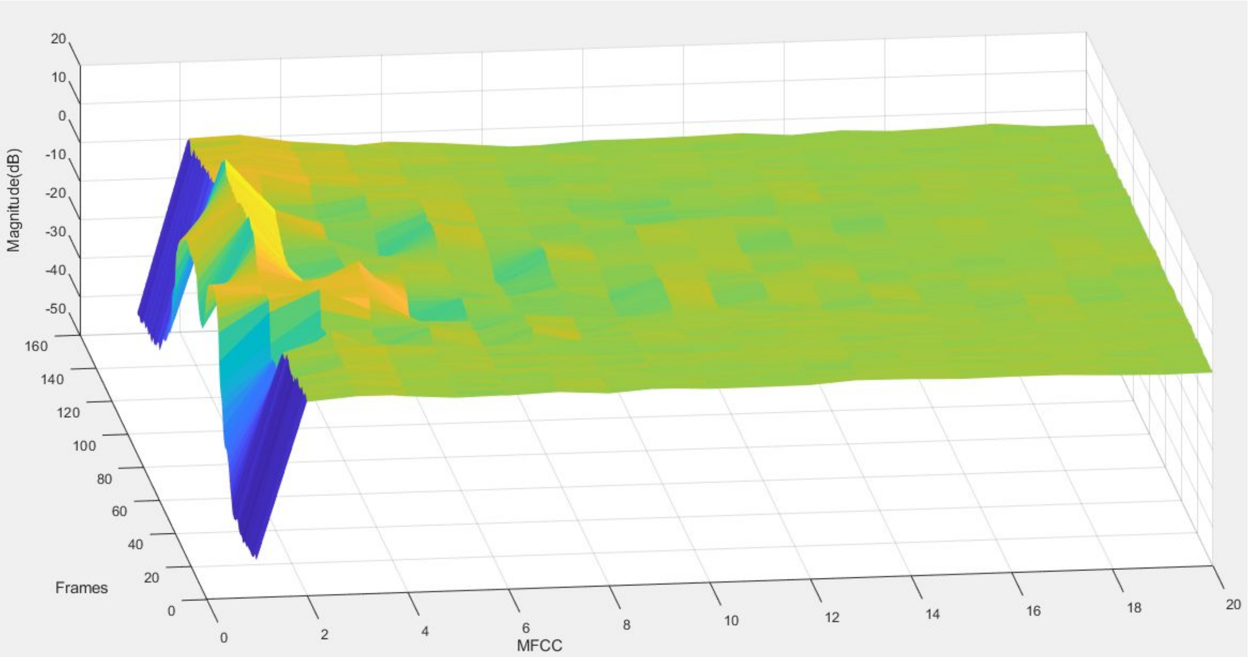


Figure : Periodogram of Speaker 1 MFCC, N = 256, M = 85

Vector Quantization

In order to match the extracted features to the test speaker, the training vectors from the MFCC must be used to create a VQ codebook for each speaker. The LBG algorithm was used to find the centroids and determine if the distortion between the centroid and samples is sufficiently small. The splitting parameter, e = 0.01, determines the position where each portion of the centroid divides, then the distance between the centroid and the closer codeword is used to update the centroid. This continues until the distortion is smaller than the splitting parameter and a codebook of size M is formed. Because each centroid splits into two for each iteration, the codebook will always be a size of the power of 2. A codebook of size 32 was used to increase the robustness of the system as one of size 16 had some errors in detecting the speaker.

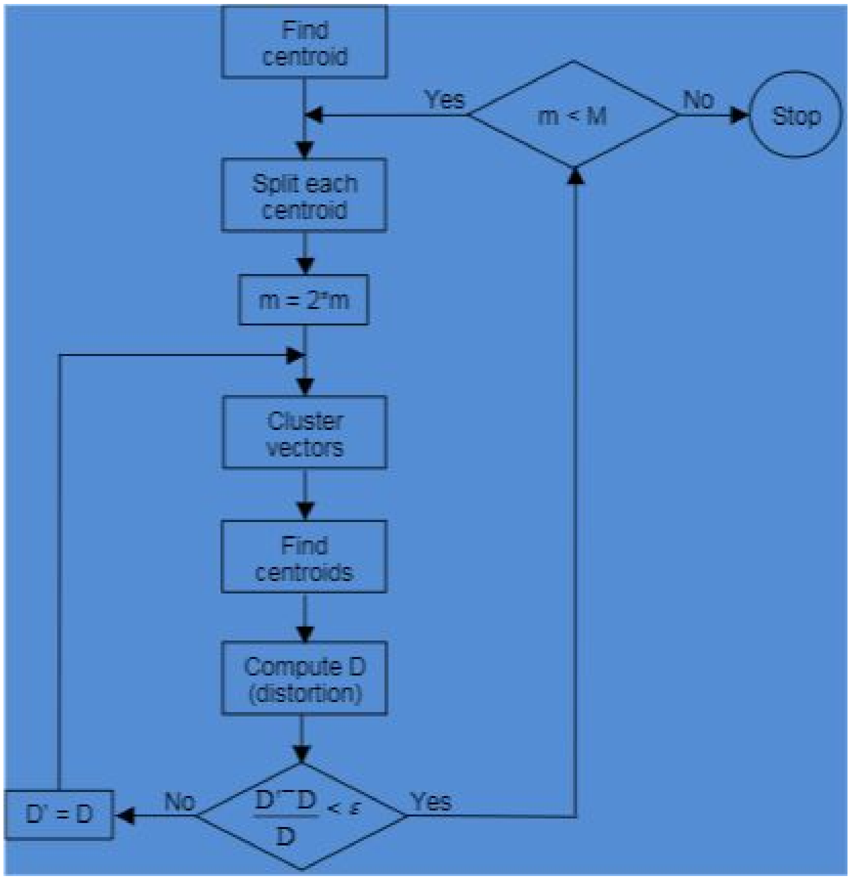


Figure 5: Flow Chart of LBG Algorithm

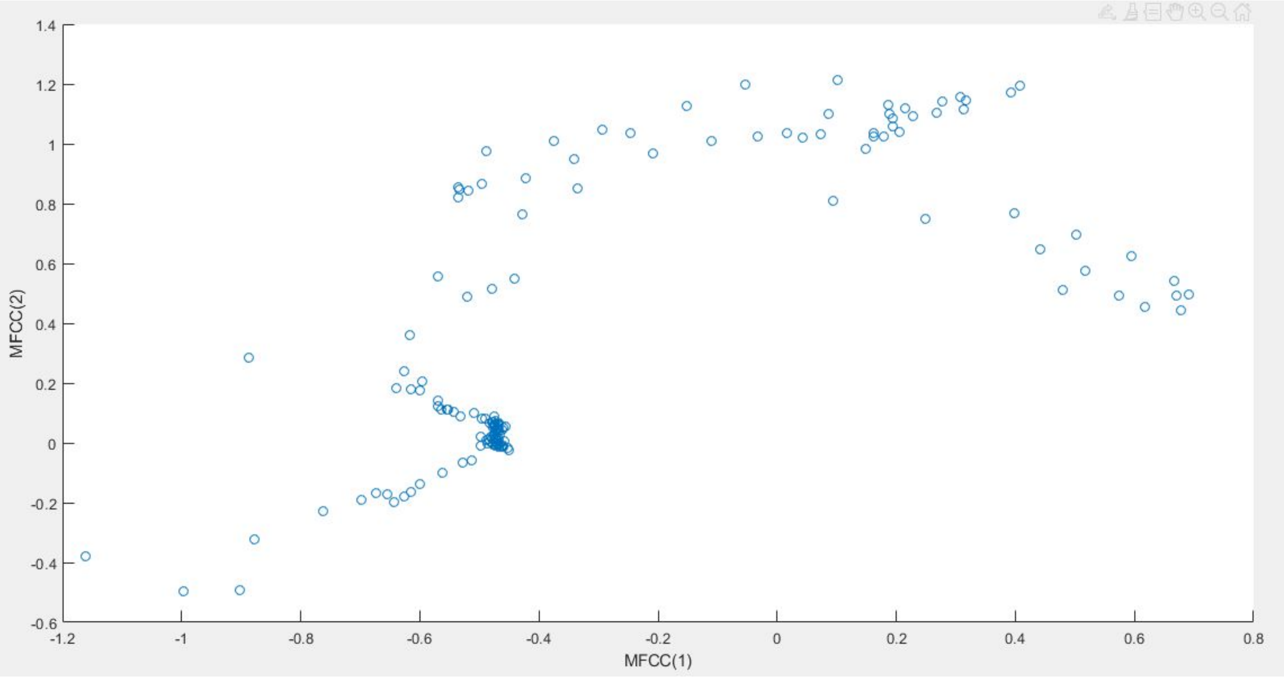


Figure : Clustering of MFCC's 1 and 2 of Speaker 1 before Vector Quantization

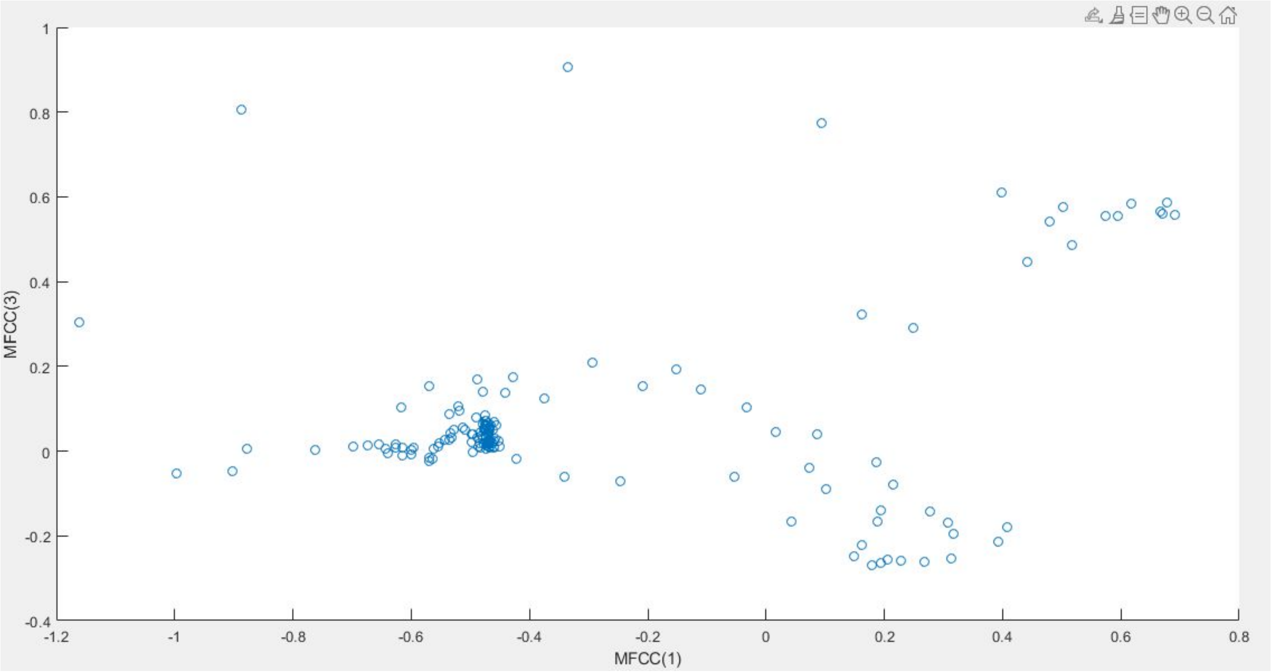


Figure : Clustering of MFCC's 1 and 3 of Speaker 1 before Vector Quantization

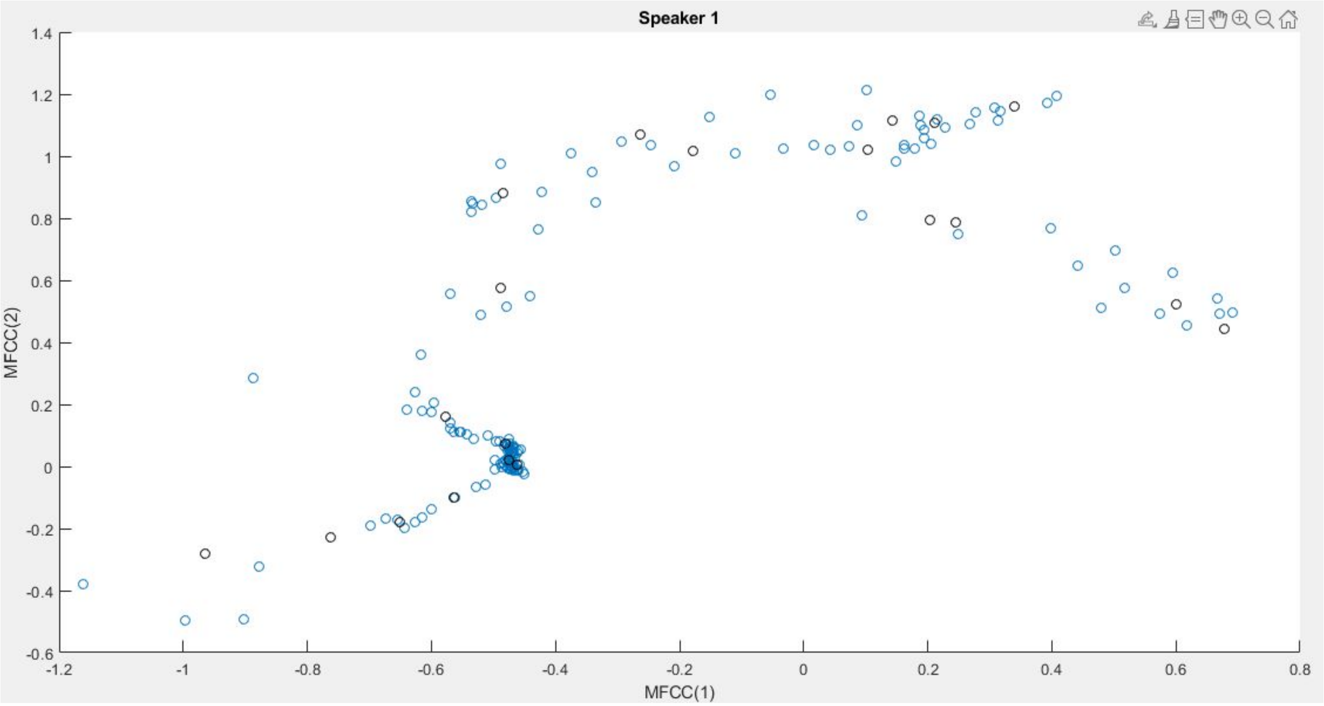


Figure : Clustering of MFCC’s 1 and 2 of Speaker 1, Centroids(Black), Samples(Blue)

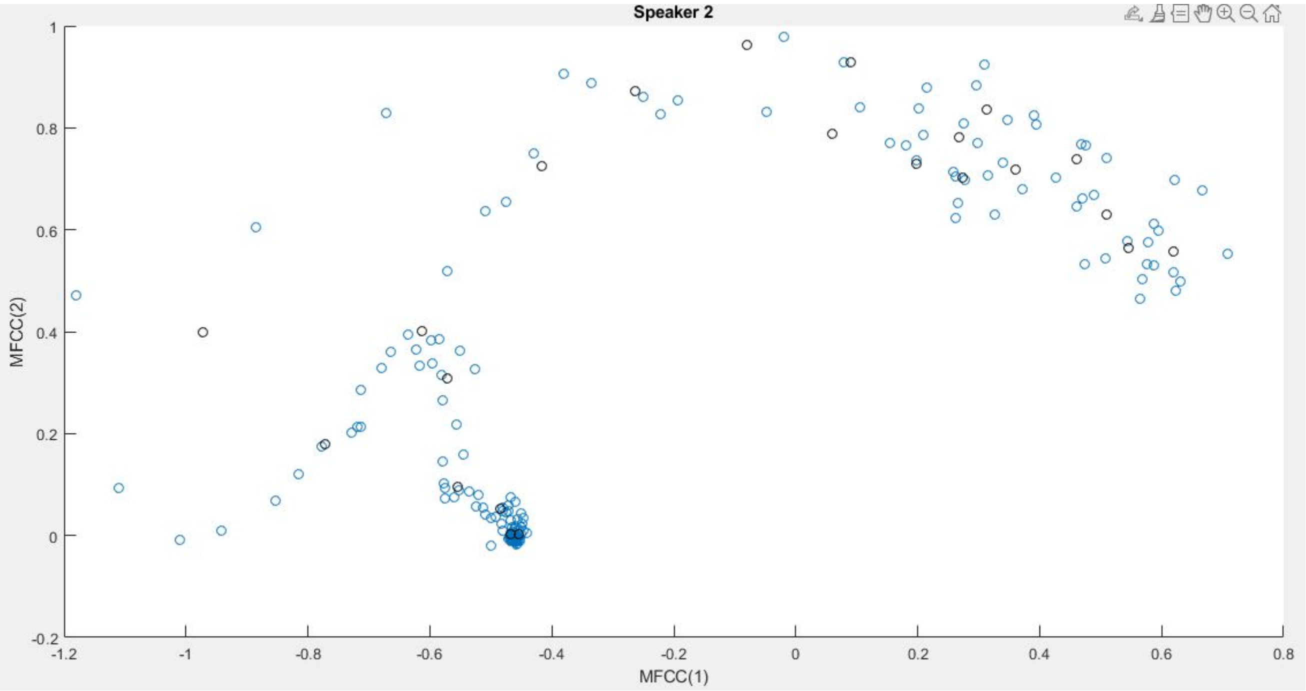
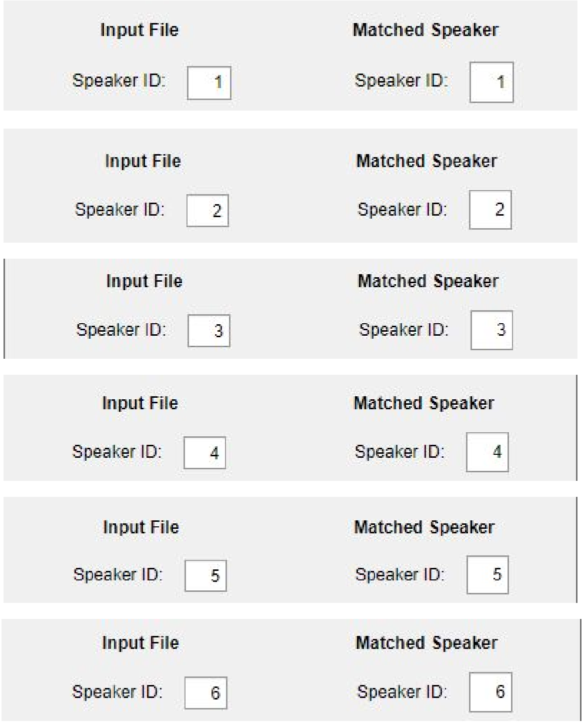


Figure : Clustering of MFCC’s 1 and 2 of Speaker 2, Centroids(Black), Samples(Blue)

Testing

After the codebooks of each training speaker is initialized, the codebook of the testing speaker is taken. Whoever's training data has the lowest distortion with the testing data, the speaker is matched. Using a GUI, the results can be quickly matched.



Input File Speaker ID: 1
Matched Speaker Speaker ID: 1

Input File Speaker ID: 2
Matched Speaker Speaker ID: 2

Input File Speaker ID: 3
Matched Speaker Speaker ID: 3

Input File Speaker ID: 4
Matched Speaker Speaker ID: 4

Input File Speaker ID: 5
Matched Speaker Speaker ID: 5

Input File Speaker ID: 6
Matched Speaker Speaker ID: 6

Figure 6: Matching of Testing and Training Speakers

The corresponding testing speaker has the same ID as the training speaker which matches with what is shown above.

Notch Filter

In order to test the robustness of the code, a notch filter at specific frequencies was used.

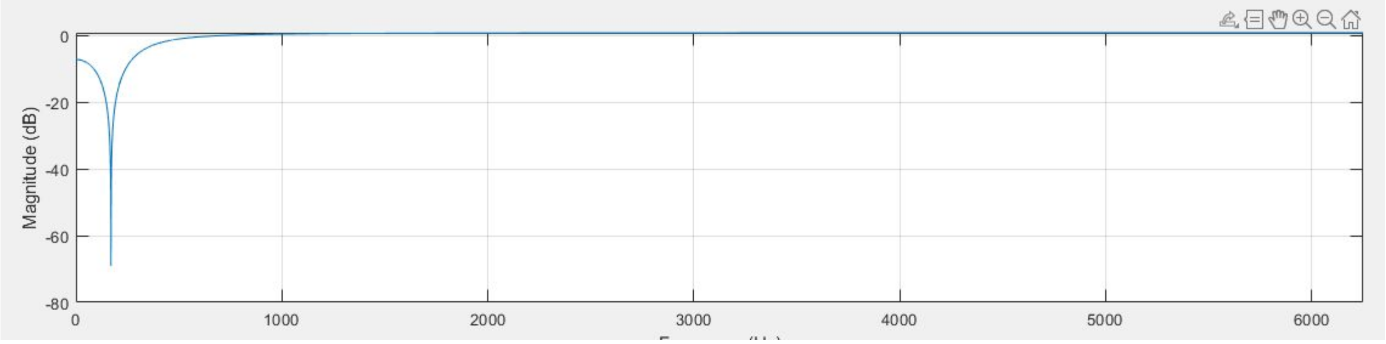


Figure 7: Notch Filter with Null Frequency at 170 Hz

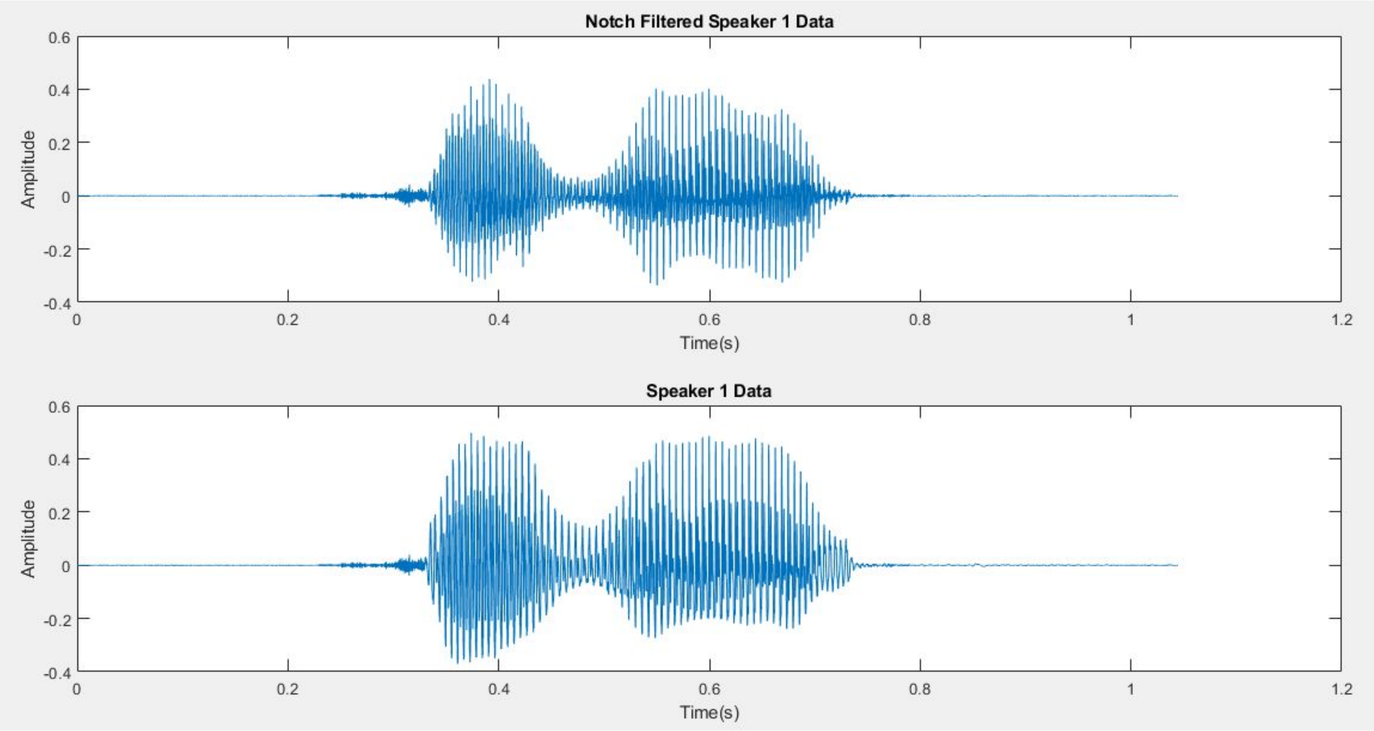
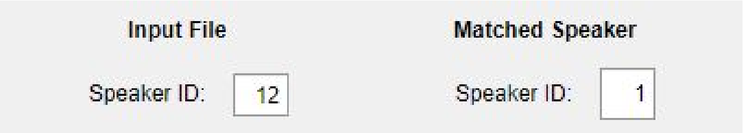


Figure 8: Notch Filtered vs. Unmodified Data of Speaker 1



Input File		Matched Speaker	
Speaker ID:	12	Speaker ID:	1

The correct training speaker can still be identified when the notch filter null frequency is far enough from the dominant frequencies.

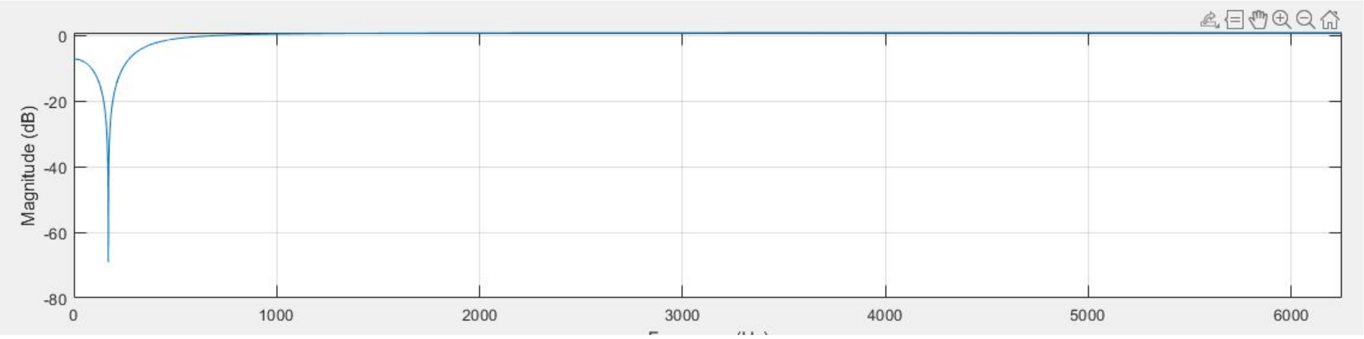


Figure 9: Notch Filter with Null Frequency at 157 Hz

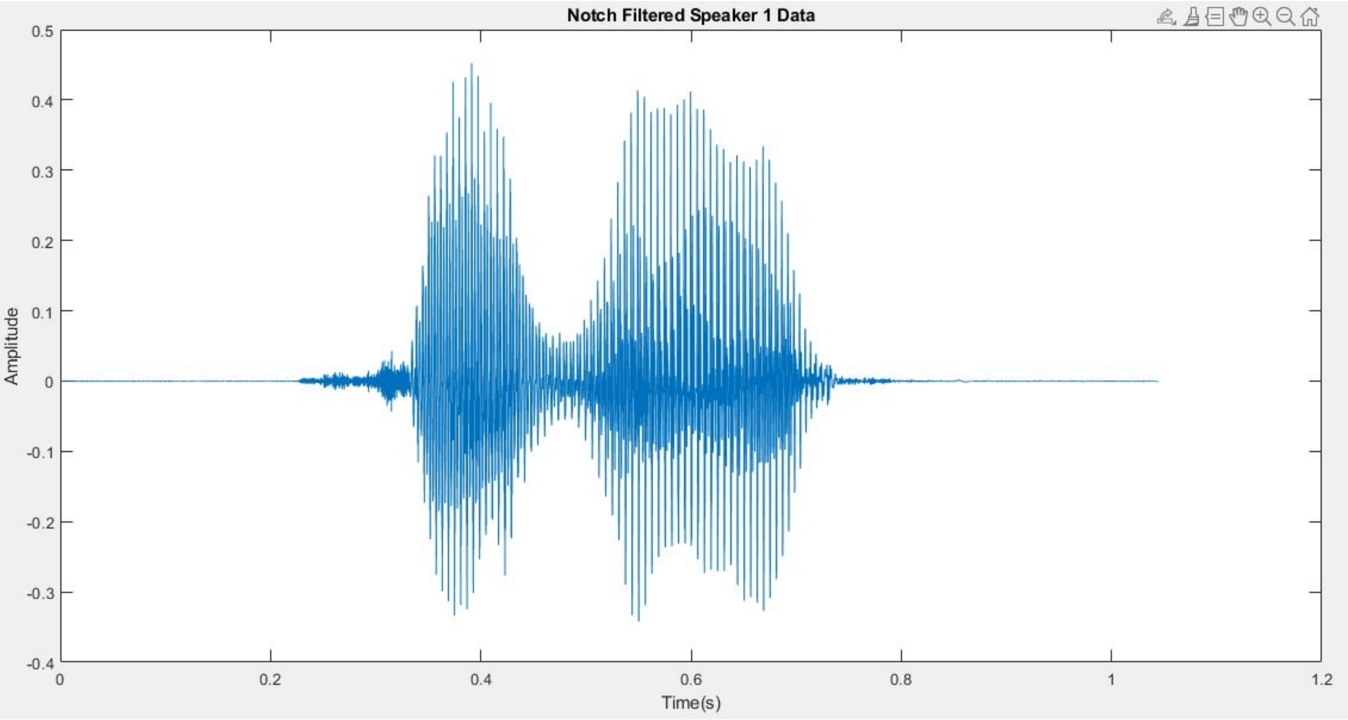
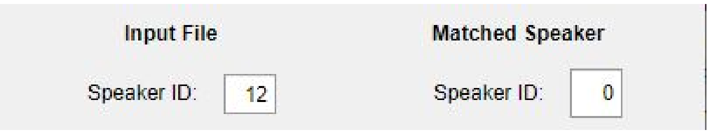


Figure 10: Notch Filtered Data of Speaker 1



Input File

Speaker ID: 12

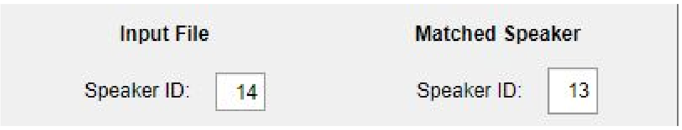
Matched Speaker

Speaker ID: 0

157 Hz is a more dominant frequency where more of the power in the signal is located, so when the notch filter was applied to the signal the testing data could not be matched to the training data as shown by the 0 under the Match Speaker.

Database

Additional voices were received from the Speech Accent Archives to further test the robustness of the code.

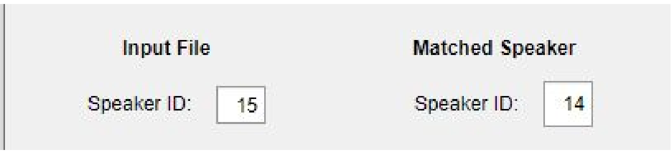


Input File

Speaker ID: 14

Matched Speaker

Speaker ID: 13



Input File

Speaker ID: 15

Matched Speaker

Speaker ID: 14

From the database the appropriate testing speaker matches to the training speaker.

References

[1] Professor Ding, "Speaker Recognition System", 2021

[2] Y. Linde, A. Buzo & R. Gray, “An algorithm for vector quantizer design”, IEEE Transactions on Commu
28, pp.84-95, 1980.

[3] Y. Tiwari, “MFCC and its applications in speaker recognition”, 10 Feb 2010