Voice Processing

Marchioro Thomas

Project 3 – Vector Quantization within LPC

1 Dataset creation

I have run the script `get_train_data.m` over the whole `train` folder of TIMIT. The script extracts the transformed reflection coefficients g_i from all the `.wav` files in the folder and subfolders and keeps track of the maximum gain $G_{\max}$. The created dataset is then saved in `train_data.mat`.

2 Scalar Quantization

The recorded value of $G_{\max}$ from the training data is employed to build a uniform scalar quantizer, which divides the range $[0, G_{\max}]$ into $L = 2^b$ equal ranges, where b is the number of bits employed for the scalar quantization. Each input gain G_j in the range $[k\Delta, (k+1)\Delta)$ is mapped into the center of the interval

$$G_q^{(k)} = \frac{2k+1}{2}\Delta,\ k = 1, \ldots, L.$$

3 Vector Quantization

In order to build the vector quantizer, I have run my implementation of the LBG algorithm, contained in the `trainVQ.m` script, on the g_i's samples extracted from the training dataset, which are vectors of size `OrderLPC`. The LBG algorithm initializes the codebook as a unique cluster, with the barycenter of the dataset as unique codeword, iteratively applies a 2-means clustering to the partitions created. Therefore, for b bits, the algorithm converges in b iterations and creates a codebook of 2^b codewords, which is much faster than a 2^b-means clustering, when b is large. Nonetheless, for large values of b, it often happens that one partition created by the LBG algorithm in an intermediate iteration contains only one sample, meaning that at the next iteration the 2-means clustering cannot further split such partition. In that case, I decided to simply skip to the next partitions (the algorithm still runs until 2^b partitions are created). In order to visualize the performance of the algorithm, I reduced the dimensionality of the dataset to 2 and plotted the resulting clusters and, as one can see from Figure 1, a good partitioning of the data is obtained.

4 Transformation of the reflection coefficients

The coefficients a_i of the vocal tract filter $H(z)$ are usually not uniformly distributed on their vector space, therefore a transformation of the reflection coefficients is more suitable for being partitioned and quantized. Since the transformation is computed as

$$g_i = \log_2\left(\frac{1-k_i}{1+k_i}\right),$$

the k_i's can be easily retrieved by applying the inverse transformation, which is found as

$$\frac{1-k_i}{1+k_i} = 2^{g_i} \Rightarrow \frac{1-k_i+1-1}{1+k_i} = 2^{g_i} \Rightarrow \frac{2}{1+k_i} - 1 = 2^{g_i}$$
$$\Rightarrow k_i = \frac{2}{1+2^{g_i}} - 1 \Rightarrow k_i = \frac{1-2^{g_i}}{1+2^{g_i}}.$$

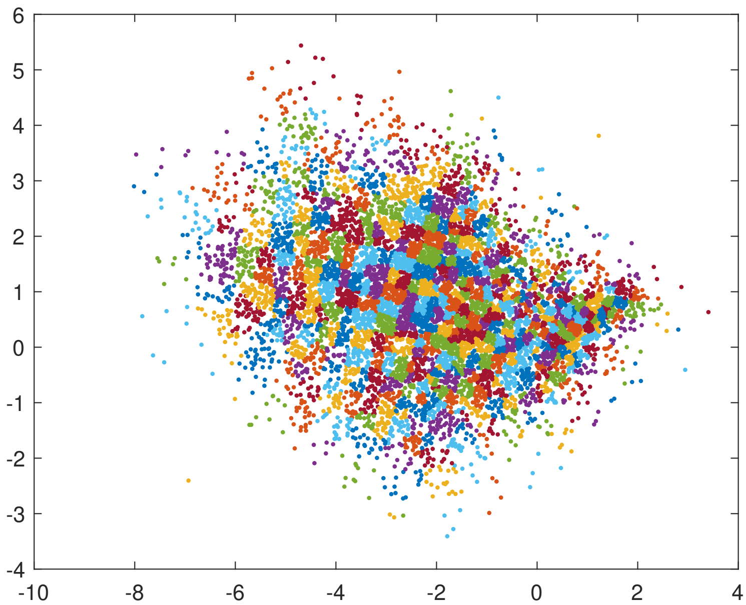

Figure 1: Example of clusters obtained running `trainVQ.m` on a 2D dataset.

The a_i's are the coefficients of the vocal tract filter's transfer function, while the k_i's are the relative IIR lattice coefficients, hence a_i's can be computed using the `latc2tf` function of MATLAB.

5 Encoding and decoding processes

The encoding procedure is done frame-by-frame: from each frame, the gain and the g_i's coefficients are extracted, quantized and converted into the corresponding indices of the codebooks. Then, the codewords and the excitation are passed to the decoder, which finds the relative quantized gain and reflection coefficients in the codebook and employs them to filter the excitation and obtaining back the original signal. Indeed, keeping the original excitation at the decoder side goes against the whole point of the source coding process, but using an "artificial" excitation (like the robot or the whispering excitation, as in Project 1) leaded to poor results. I applied the LPC coding scheme with a filter of order 6, using 6 bits to quantize the gain and 13 bits to quantize the g_i's. The overall reconstruction is quite good, as one can see from the comparison with the original signal in both time and frequency domains (Figures 2 and 3). Since I used the original excitation, better quality is achieved with a lower order of the LPC filter, but if a different excitation is employed (or if even excitation is quantized) a higher order LPC is more likely to achieve better results.

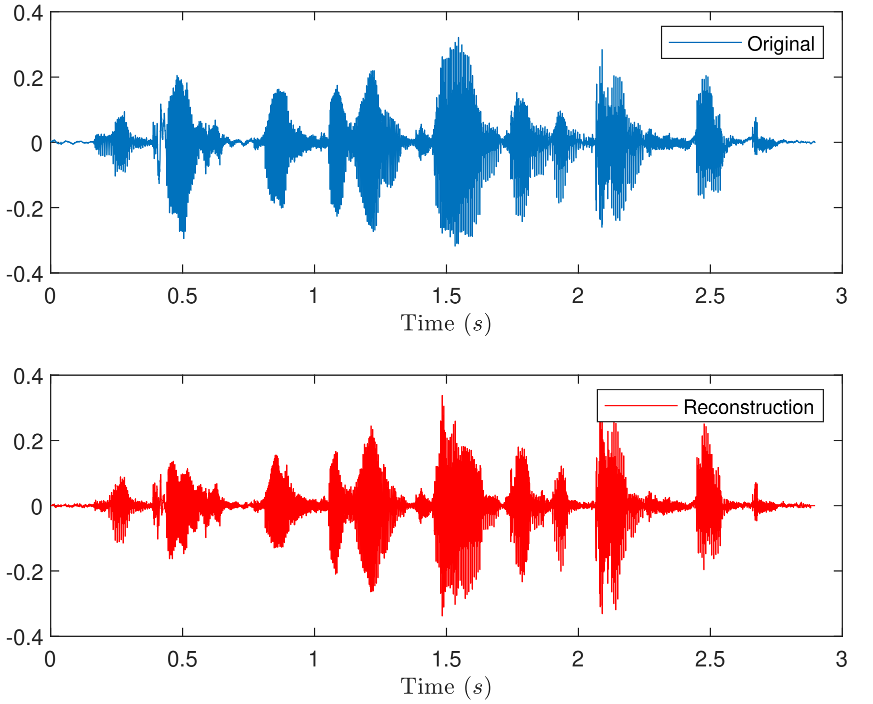

Figure 2: Comparison of the original and reconstructed signals in the time domain.

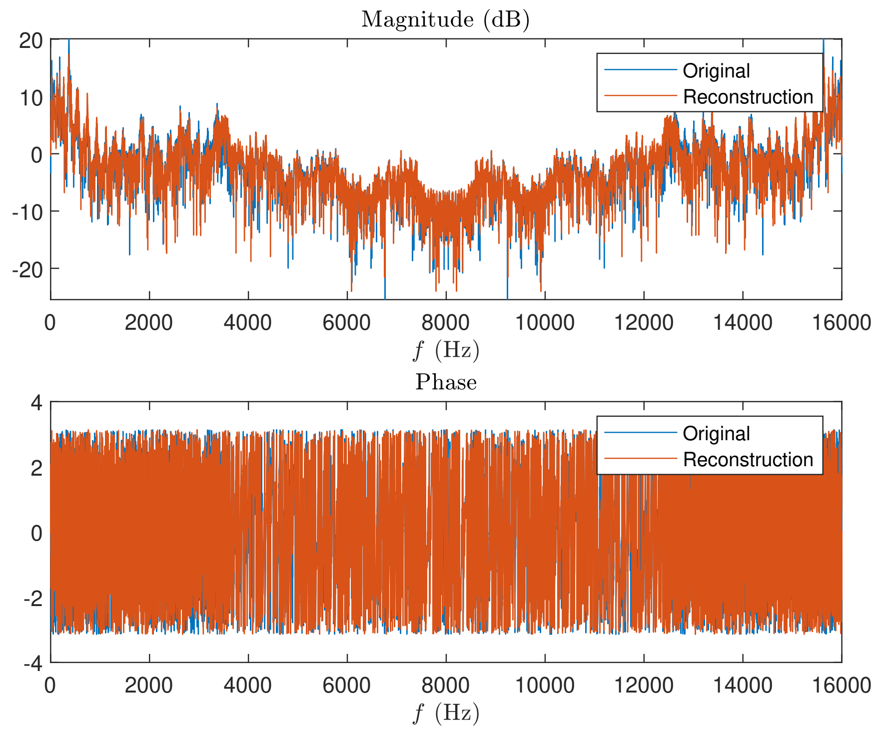

Figure 3: Comparison of the original and reconstructed signals in the frequency domain.