Parametric Speech Coding

Doruk Resmi
Department of Electrical-Electronics Engineering
Koç University
Istanbul, Turkey
dresmi13@ku.edu.tr

Mustafa Başaran
Department of Electrical-Electronics Engineering
Koç University
Istanbul, Turkey
mbasaran13@ku.edu.tr

Abstract—**In this project, an implementation for data compression of any given speech signal into a more suitable form for data transmission and storage without having a significant data loss; and reconstruction of the original signal was done through the methods of Linear Predictive Coding and vector quantization.**

Keywords—speech coding, parametrization, quantization

I. Introduction

Discretization of analog signals due to prevalent use of binary logic in computer architecture always posed as a problem since the invention of computers. With discretization comes with the problem of data storage and data transmission since compared to their analog counterparts, binary systems require more ‘bits’ of information to represent the same analog data, which drive scientists and engineers to come up with efficient ways of compressing the digitalized data.

Our project aims to implement a way of lossless compression technique specialized for compressing speech signals, namely Linear Predictive Coding. Moreover, the speech or the audio signal is further compressed by quantization, which is a lossy compression technique, according to the specifications given in the project description.

II. Phase 1: Linear Predictive Coding

A. Specifications

Our implementation takes a sound file of WAV format of unknown length and sampling, within the same directory of the source code. The length of the sound file or the sampling frequency of the sound file does not affect the implementation, namely, our implementation can take various sampling rates and various sound files of variable length and return the desired output. Yet, the library we used for this project, the MTR library, a library of speech sounds consists of sound files of generally few seconds with sampling frequency of 8 kHz.

B. Theory

The human sound production mechanisms can be modelled as excitation generator located at the throat, creating impulse train or white noise according to the sound to be produced, where impulse train is representing the vowels and the white noise is representing the consonants. These sounds are then ‘shaped’ throughout the folds of the vocal tract, namely lips, tongue, palate, pharynx etc. in accordance with the sound aimed to be produced in the context of the speech.

The vocal tract itself can be approximated as an open ended tube with varying thickness and length, shaping itself throughout the speech in order to generate unique harmonics of each phoneme, by suppressing or amplifying various bands or regions of the frequency spectra of the excitation signal, resulting with proper speech sounds. These mentioned harmonics are called as formants, and each phoneme has its own unique formant, meaning that each phoneme has a characteristic distribution of its frequency spectra.

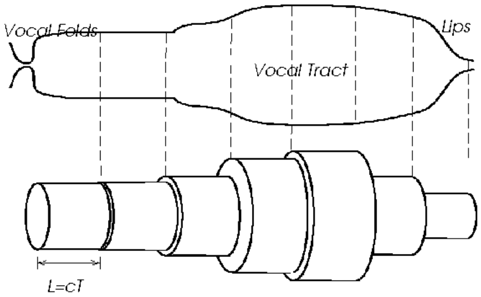


Figure 1: Lossless Tube Model of Vocal Tract

This suppressing and amplifying behavior of the vocal tract can be modelled by an all pole linear finite impulse response filter. For our project, we were given the task of approximating the vocal track as a time varying 10-pole FIR filter.

Our motivation for these models is to extract the related coefficients of the 10-pole FIR filter from the speech signal by a technique called Linear Predictive Coding, to deconstruct the speech signal into the excitation signal as it is generated at the throat by applying the inverse filter. In order to achieve that, first we need to find the parameters of the said filter.

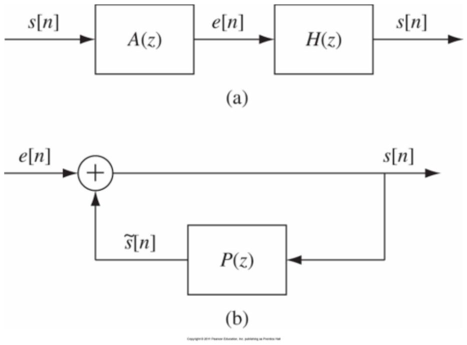


Figure 2

From our knowledge of speech production mechanisms obtained throughout the course, we know that each instance of speech is related to the next or the previous instances of the same speech. By mustering the said fact, Linear Predictive Coding predicts the next instance of the speech by a weighted combination of the previous instances, capturing the characteristics of the vocal tract at the moment of the speech production.

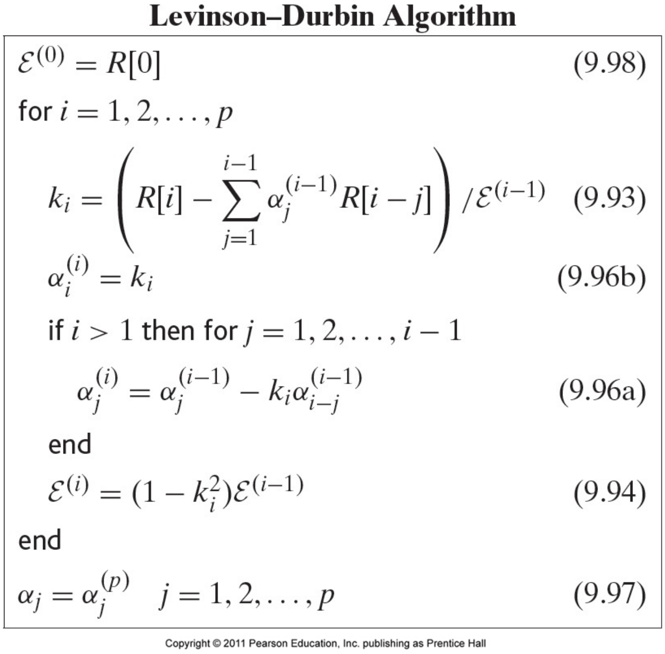
$$s[n] = \sum_{k=1}^{p} \alpha_k s[n-k]$$

Figure 3: The basic assumption of Linear Predictive Coding

C. Implementation: Encoder

The encoder of our implementation follows the steps:

- Framing the speech signal into two milliseconds frames, which corresponds to a 160-sample frame for 8 kHz sampling rate, with 50% frame shift.
- Each frame is windowed by Hanning window of equal size.
- Calculating a 10 step autocorrelation, and retaining the values to obtain an autocorrelation vector, where autocorrelation vector is generated without any use of built-in MATLAB functions.
- Calculation of reflection coefficients from autocorrelation vector by Levinson-Durbin algorithm.
- Calculation of filter coefficients/parameters from reflection coefficient.
- Extraction of excitation signal by applying the filter to the speech signal.
- Returning the user PARCOR coefficients, excitation signal and gain of the filter.



Levinson–Durbin Algorithm

```
$\mathcal{E}^{(0)} = R[0]$                                                    (9.98)
for $i = 1, 2, \ldots, p$
    $k_i = \left( R[i] - \sum_{j=1}^{i-1} \alpha_j^{(i-1)} R[i-j] \right) / \mathcal{E}^{(i-1)}$    (9.93)
    $\alpha_i^{(i)} = k_i$                                                    (9.96b)
    if $i > 1$ then for $j = 1, 2, \ldots, i-1$
        $\alpha_j^{(i)} = \alpha_j^{(i-1)} - k_i \alpha_{i-j}^{(i-1)}$                  (9.96a)
    end
    $\mathcal{E}^{(i)} = (1 - k_i^2) \mathcal{E}^{(i-1)}$                                (9.94)
end
$\alpha_j = \alpha_j^{(p)} \quad j = 1, 2, \ldots, p$                                 (9.97)
```

Figure 4: Pseudocode of Levinson-Durbin Algorithm

D. Implementation: Decoder

The encoder of our implementation follows the steps:

- Takes excitation signal, PARCOR coefficients and gain values as input.
- Constructs a filter by using PARCOR coefficients and gain values, which is the inverse filter that was implemented in the encoder.
- Applies the recently constructed filter to the excitation signal.

- Returns reconstructed signal, which supposed to be the same as speech signal that we used at the encoder, without any loss

E. Results

When we listen to the both original and reconstructed signals, there isn’t any detectable difference present between these two signals. Our segmental signal-to-noise ratio (segSNR) calculations are between 38 and 47 for various speech files. These values show that our implementation is fairly good, yet not perfect. Since the filter we applied to extract the excitation signal and the inverse of the same filter which we used to generate speech signal from excitation signal is linear, there has to be no data loss. However, application of Hanning windows to the each frame leads to an unavoidable attenuation, which explains the loss of information. Even though we preferred Hamming window throughout the course, upon experimentation with various windows, such as Bartlett window, Hamming window and other well-known windows, Hanning window returned the best results, in terms of both sound quality and segSNR. Therefore, we adopted the usage of Hanning window in our calculations.

III. PHASE 2: QUANTIZATION

To make the excitation signal and PARCOR coefficients suitable for data transmission and data storage, we have to compress the data available in them by quantization process. By following the steps of quantization in the inverse order, we are able to reconstruct the original signal, within a range of acceptable data losses.

A. Specifications

In this part of the project, we were asked to implement four different kinds of codec, each having a slightly different specification from the other three. In general, these are composed of 240 or 480 bits per frame of excitation signal; and 20 or 40 bits for 10th order PARCOR coefficients, meaning each PARCOR coefficient gets two or four bits. Both of the quantizations are done by vector quantization.

B. Theory

The idea behind the vector quantization is to group sets of data in a way that each subgroup is consisted of similar data points, or namely clusters. The number of the subgroups are chosen as necessary, depending on the application.

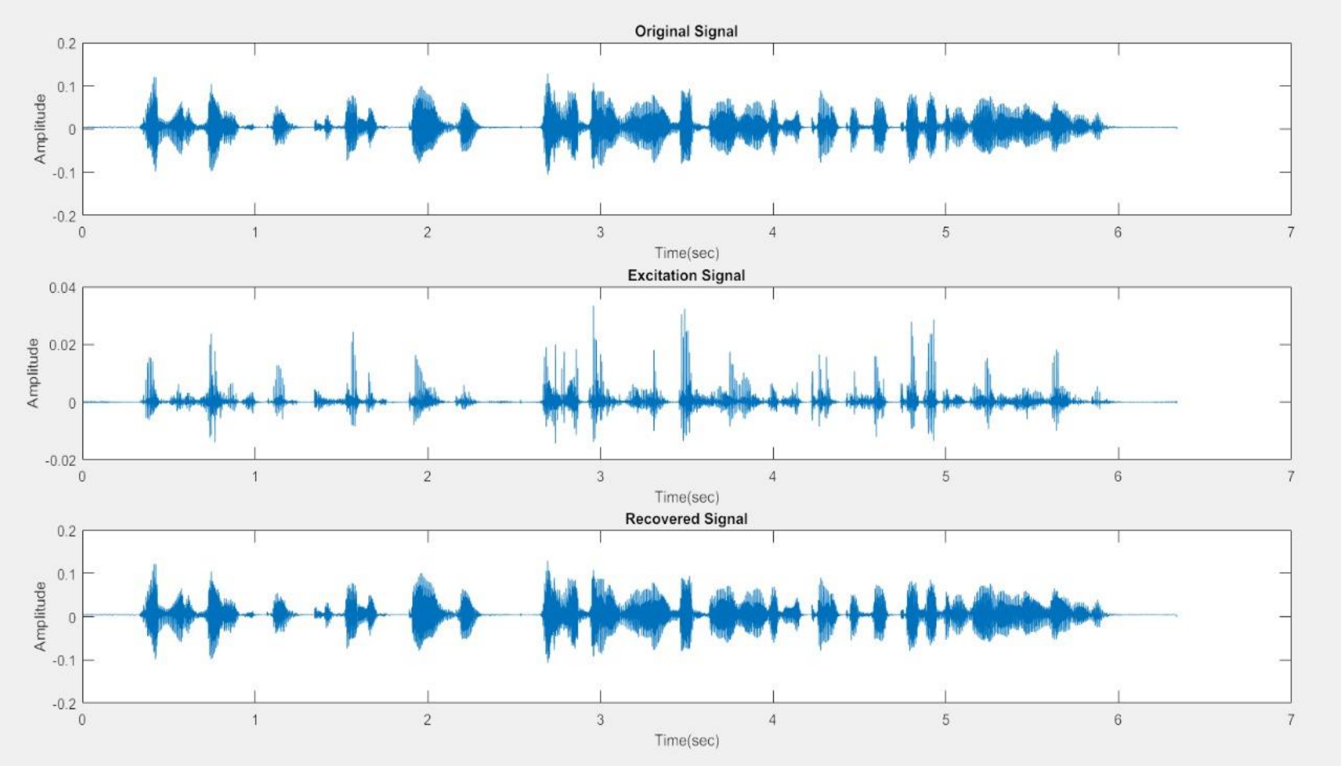


Figure 5: A visual comparison of Original, Excitation and Decoded signal, without quantization

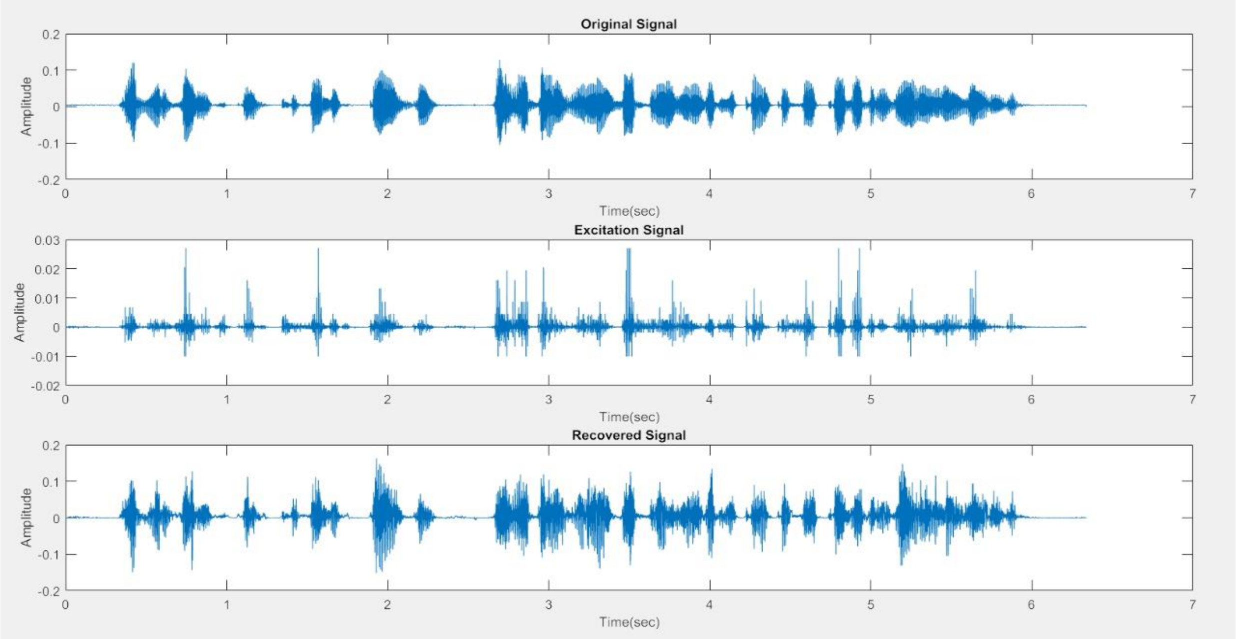


Figure 6: A visual comparison of Original, Excitation and Decoded signal, with quantization with Codec 1

In our project, we employed k-means clustering algorithm, which iteratively ‘trains’ to locate points called centroids, the points where the data set tend to cluster around, and divides the region into sub-regions corresponding to a Voronoi diagram.

By using the codebook generated prior to the application, the de-quantization or decoding process happens by referring to the codebook for the assigned outcome of each combination of bits allocated.

C. *Implementation: Quantization of PARCOR Parameters*

For PARCOR Parameters, we used 1-D clustering. To generate the codebook the total of PARCOR Parameters are made use of, with the purpose of creating a more definitively and accurately defined boundaries. For 2-bit quantization, the data of PARCOR Parameters are divided into four clusters; whereas for 4-bit quantization, the data of PARCOR Parameters are divided into 16 clusters. K-means clustering method is used to divide the data to their respective groups, with the help of built-in MATLAB function kmeans().

D. *Implementation: Quantization of Excitation Signal*

For excitation signal, we were assigned to use 480 bits or 240 bits for each segment of the excitation signal, depending on the codec. With 160 samples per frame at hand, it is not possible to represent 160 samples with 240 bits where each sample having 1.5 bits, which clearly not possible. To overcome this problem, we grouped each consecutive sample into groups of two(X[n]=(x[2n] , x[2n+1])), again assuming that each sample of a speech signal is related to the other samples in vicinity, thus reducing sample to 80 data point each having 2 different values. After this modification, each group of samples has 3 or 6 bits for representation, meaning that 8 or 64 point clustering will be implemented in our design.

Similar to quantization of PARCOR coefficients, the whole data points are used to generate the codebook more robustly. So when generating centroids for PARCOR coefficients, ten times number of frames data is used. Also when generating centroids for excitations signal eighty times number of frames data is used. Therefore, same codebook is used for all quantization and dequantization process.

E. *Results*

After the embedding of vector quantization into Encoder/Decoder, our system obtained significant amount of noise. The quality of the sound is decreased, where the sound of the speaker became much noisy. Moreover, our segSNR numbers plummeted to the range of 4 to 6.

This sharp change in quality is definitely caused by quantization process, since during the implementation of the project the encoder and decoder parts of the code were kept intact, and the only difference being the quantization. To further addressing this problem, we speculated that our bit allocation requirements are significantly low to properly discretize the samples. As a side note, we would like to add that with our experimentation with various numbers for bit allocation, we observed that the signals quantized by 480 bit 1-D clustering (3 bit per sample) were better at quality and segSNR compared to 480 bit 2-D clustering (6 bit per a couple of samples.)

IV. Conclusion

After comparing the four different codecs we were expected to implement, we can say that as the codec number goes up (from 1 to 4), the quality of sound diminished. Yet, we have significant amount of noise and disturbances in every single implementation of codec, regardless of the codec number, highly probable due to lack of bits allocated to each frame of excitation signal. Apart from that, we can say that the rest of our project works properly.

Acknowledgment

We as a group, would like to personally thank Engin Erzin for presenting us the great opportunity to learn such an interesting object, for creating the suitable environment in which a curious and inquisitive mind thrive and for conveying the vast information this subject features effectively and clearly. Likewise, we would like to express our gratitude to our dear teaching assistant Mehmet Ali Tuğtekin Turan for his academic support, for his effort and enthusiasm in answering our numerous questions, and for his labor in organization of various events throughout the year to make sure our learning and course evaluation processes are less problematic.

References

[1] L.Rabiner and C. Rader, *Digital signal processing*. New York: IEEE Press, 1972.

[2] L. Rabiner and R. Schafer, "Introduction to Digital Speech Processing", Foundations and Trends® in Signal Processing, vol. 1, no. 12, pp. 1-194, 2007.

[3] E. Erzin, "ELEC-COMP 404/504: Digital Speech and Audio Processing Lecture Slides", Koç University, 2017.

[4] http://www2.engr.arizona.edu/~sail/Past_Research/siddharth/tract.gif. 2017.