# CS578 Speech Processing
# Laboratory 3
# Vector Quantization & LPC Coding

George Manos
csd4333@csd.uoc.gr

Alexandros Angelakis
csd4334@csd.uoc.gr

16 January 2023

## 1 Implementation Details

First of all, we do the analysis on each speech signal in our train set by storing all of their LPC analysis gains and the corresponding companding reflection coefficients. The order of the LPC was set to 10. Using them, we then build the scalar and vector quantizers. After building them, we can check the quantization results over to the training set of the scalar quantizer, to see how the quantization levels were applied to the gains of training sample over all frames. Finally, we perform the Analysis-Quantization-Synthesis function to every speech signal in the chosen test data set with the quantized LPC values we've found, and compare the original signal with the synthesized one, both by plotting them and listening to them (with a 3 seconds delay in between), printing the computed Mean Squared Error as well.

## 2 Scalar Quantization

For the Scalar Quantization, we implemented a simple uniform quantizer. As we know, this quantizer is going to be used for the gains of each speech signal and these gains are always positive. Thus, we considered our quantization boundaries: $0 \leq x[n] \leq 4\sigma_x$, where $\sigma_x$ is the standard deviation of all the gains in the train data set. Then for new gain values, the uniform quantizer will return the quantized value by searching the first decision level where $G \leq x_i$, and then take the mean between this decision level and its previous ($\hat{G} = \frac{x_i + x_{i-1}}{2}$, $0 \leq i \leq L$). If there is a gain that is larger than $4\sigma$, we clip it to the maximum quantization value (essentially to $\frac{x_L + x_{L-1}}{2} = 4\sigma_x - \frac{\Delta}{2}$, where $\Delta$ is the quantization step). The quantizer is applied on the original gain value as resulting from the analysis step, and the quantized gain value is used on the synthesis step. Again, as noted from the project description, we commented out the lines where the signal is normalized according to the energy of the analyzed signal.

### 2.1 Results

Our scalar quantization works pretty well for our data sets as the gains over all frames for most testing speech signals are uniformly quantized over many quantization levels. Our first implementation was using the decision levels in the range $min_G \leq x[n] \leq max_G$, where $min_G$ and $max_G$ are the minimum of all the gains and the maximum of all the gains of our train data set respectively. This solution was really sensitive to outliers (really high and really low gain values respectively) and would not distribute the gain values of our testing speech samples over to all quantization levels, but they would usually limit themselves into only a couple specific ones. Especially fewer quantization levels, e.g. 4, we can see that the data is only quantized in 2 levels, as the $max_G$ value of all training data is much higher than the $max_G$ value of that specific point.

Using the $4\sigma_x$ as our max range sounded as an even better solution for our quantizer, as this utilizes the sparsity of the gains of our training data. We can also visually see a significant improvement of the quantization, as the gain values are distributed over all 4 quantization levels, but also a better fit on some points (e.g. comparing figure 2 and 4 at index 160)

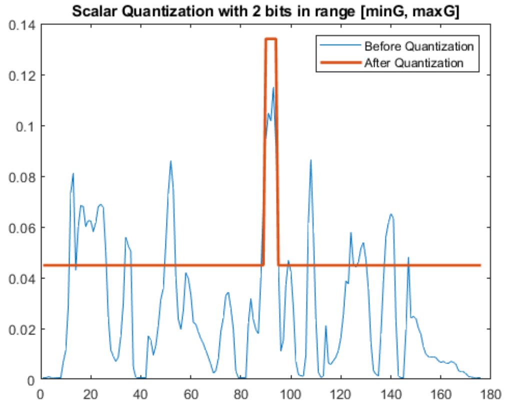

Figure 1: Uniform quantization in range $min_G \leq x[n] \leq max_G$ on sx276.wav

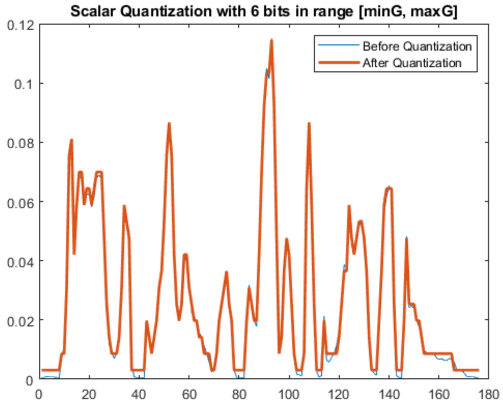

Figure 2: Uniform quantization in range $min_G \leq x[n] \leq max_G$ on sx276.wav

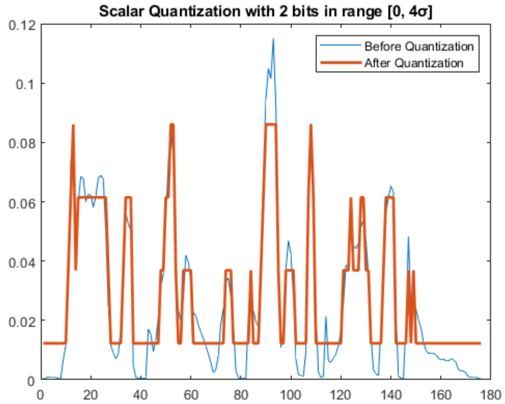

Figure 3: Uniform quantization in range $0 \leq x[n] \leq 4\sigma_x$ on sx276.wav

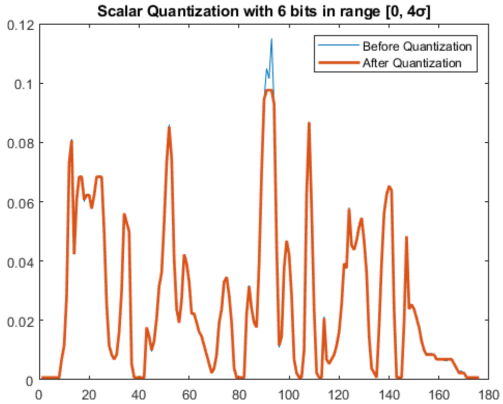

Figure 4: Uniform quantization in range $0 \leq x[n] \leq 4\sigma_x$ on sx276.wav

On this part, we apply only scalar quantization before synthesizing our signal (essentially without VQ, for experimentation purposes). Listening at the synthesized signal with quantized gains, we can definitely hear a difference. Its quality is significantly worse, having a robotic feeling, although we can still understand what the speaker is saying.

Let's take a look at the speech signals. We can see that the energy from the voiced frames has been reduced, and the energy of the unvoiced has been increased. By computing their energy, we can see that a lot of energy from the original signal has been lost. For example, the energy of the original signal is $E_{original}$ = 8.928014 and the energy of the synthesized signal is $E_{synthesized}$ = 0.024409.

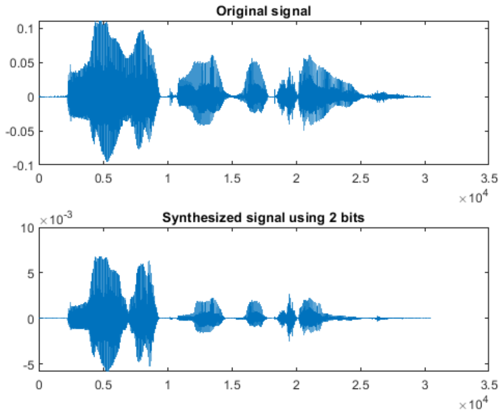

Figure 5: Synthesis using only scalar quantization on gains on sx391.wav (2-bits)

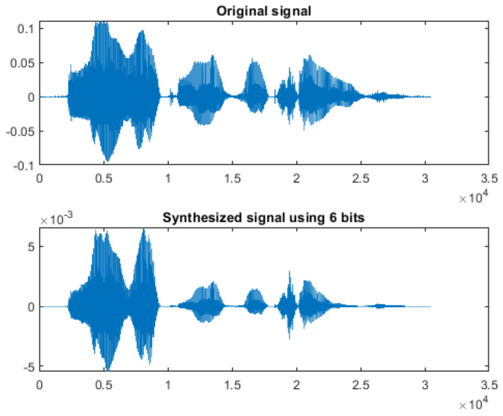

Figure 6: Synthesis using only scalar quantization on gains on sx391.wav (6-bits)

By increasing the number of bits to 12, there is no such difference visually by looking at the signals' plots, but acoustically, the quality is significantly better. That robotic feeling we mentioned before is gone and the quality of the synthesized signal is almost the same as the original one. Having more quantization levels means that the quantized values will be closer to the origianl value.

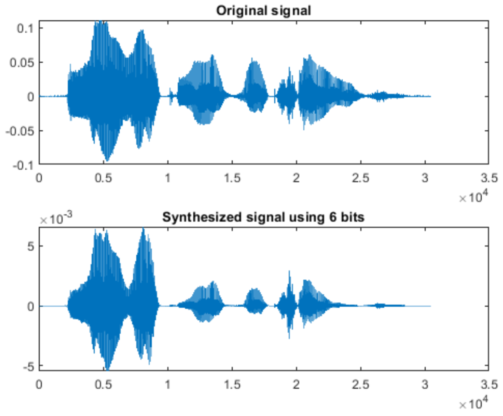

Figure 7: Synthesis using only scalar quantization on gains on sx391.wav (6-bits)

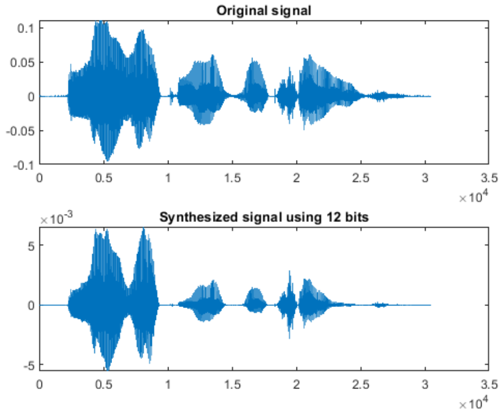

Figure 8: Synthesis using only scalar quantization on gains on sx391.wav (12-bits)

# 3 Vector Quantization (VQ)

The major part of the Vector Quantization (VQ) algorithm is to build the proper codebook (collection of codevectors that new data will be quantized to). We did that by using the LBG algorithm as described in the lectures. First of all, we initialized the first centroid as the mean vector (sized 1xOrderLPC) over all training data. Then, we split the computed centroids iteratively into 2 subcentroids and used them as initial centroids on the kmeans algorithm, realigning them to the center of each cluster. To properly split each centroid, we considered a small constant number $e = 0.05$ which will be added and subtracted from the original centroid on each dimension. The process is repeated while $k < 2^B$, where $B$ is the number of bits used for the quantization.

The constructed codebook is then used to quantize the companding reflection coefficient vector of each speech frame. The vector is matched to the codebook's codevector that has the minimum euclidean distance.

## 3.1 Vector Quantization Process

As mentioned before, in the LPC analysis we also extract the reflection coefficients $k$ of every speech frame, as resulted from the Levinson algorithm. Then, we calculate the companding reflection coefficients by using the formula $g_i = log\frac{1-k_i}{1+k_i}$. The vector quantization is applied to those $g_i$ values using the previously constructed codebook.

After the quantization, the next step is to recompute the LPC coefficients. First, we have to apply the inverse companding formula: $k_i = \frac{1-10^{g_i}}{1+10^{g_i}}$ to compute the new reflection coefficients, and then, using the method described in the lectures we compute the new LPC coefficients. The excitation signal is of course computed with the original LPC coefficients a, as resulted from the levinson algorithm, and the new LPC coefficients $\hat{a}$ are only applied on the synthesis part.

## 3.2 Results

As we expected, the number of bits play a major role in the synthesis part, especially on the quality of the synthesized speech signal. By using a low number of bits such as 2 (thus 4 quantization levels), we can hear that a lot of information has been lost from our original signal and the quality is really bad as the difference brought by the quantization process is quite big, for the reasons described above. In this case, the mean squared error (MSE) between the original and the synthesized signal for the sx96.wav file is $MSE = 0.000507$.

Using a higher number of bits such as 6 (thus 64 quantization levels), we have better results in our synthesized signal. The MSE is down to 0.000492, and although the difference may seem small[*], the audio quality of the synthesized signal is significantly improved.

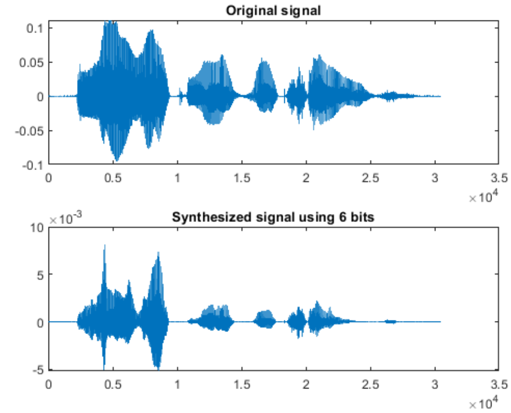

Figure 9: VQ using 6 bits on sx391.wav

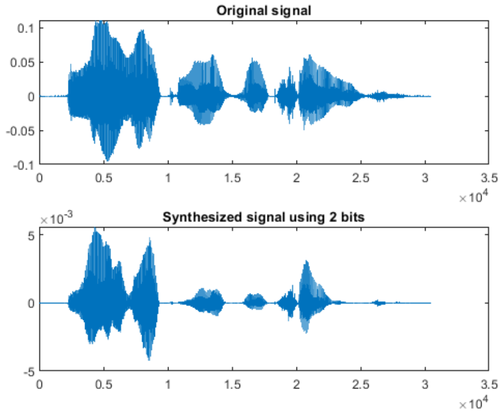

Figure 10: VQ using 2 bits on sx391.wav

By listening to the synthesized signal using only 2 bits for the quantization process, we can say that some characteristics of the speakers' voices have been lost and they sound like they are whispering, resulting in a robotic effect. As we increase the number of bits we are using for the vector quantization process, the synthesized signal sounds more and more like the original speech signal.

[*] This would also be a good moment to remind that since the signal has not been normalized, all of the numbers will look relatively small.

# 4 Experiments

Overall, we apply quantization on both the analyzed signal's Gain and LPC coefficients. Both quantizers are constructed using a training set, and are applied on a different test set. If the training set gains and LPC coefficient values were much different than the test set ones, for example if the training set included only male speakers and the test set only female ones, the reconstruction mean squared error would be significantly higher as the quantization levels would be much different from the testing input values and vectors respectively. For this reason, we must make sure that the training and test sets are both balanced and that our quantizers are fairly generalizable (e.g. the uniform scalar quantizer that uses the training set's x max value as maximum instead of the standard deviation was not generalizable as a training speech signal with a really high gain value on even 1 frame would significantly increase the difference between the quantization levels).

## 4.1 Comparing Codebooks

Increasing the number of bits for the vector quantization, we can once again notice the synthesis resluts improve. The bigger the codebook size, the less information is likely to be lost from the quantization process. The difference is not only acoustic but also visual, as on the plots we can see the reconstructed signals using 6 bits are closer to the original signal, compared to the ones using 2 bits.

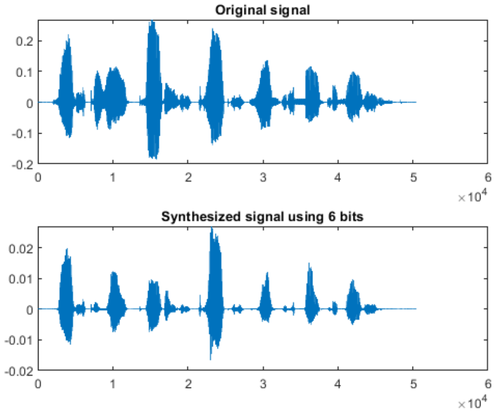

Figure 11: VQ using 6 bits with MSE: 0.000494 on sx96.wav

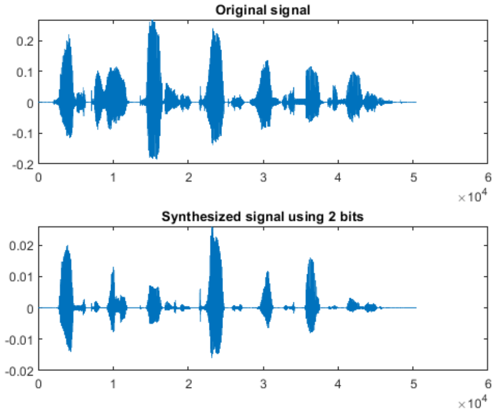

Figure 12: VQ using 2 bits with MSE: 0.000504 on sx96.wav

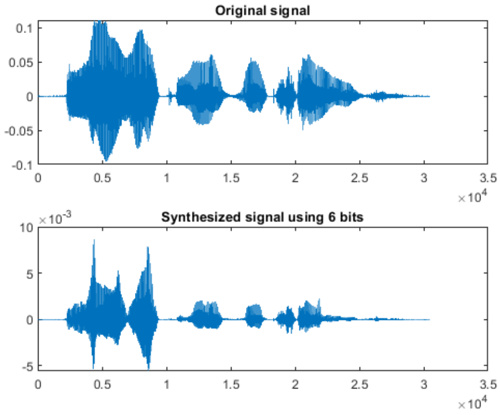

Figure 13: VQ using 6 bits with MSE: 0.000276 on sx391.wav

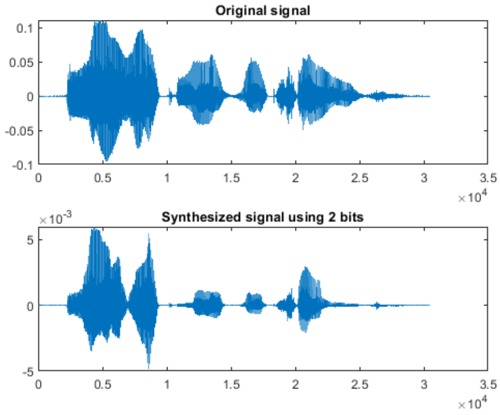

Figure 14: VQ using 2 bits with MSE: 0.000279 on sx391.wav

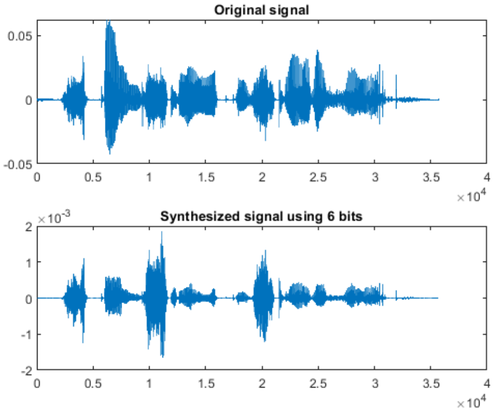

Figure 15: VQ using 6 bits with MSE: 0.000042 on sx72.wav

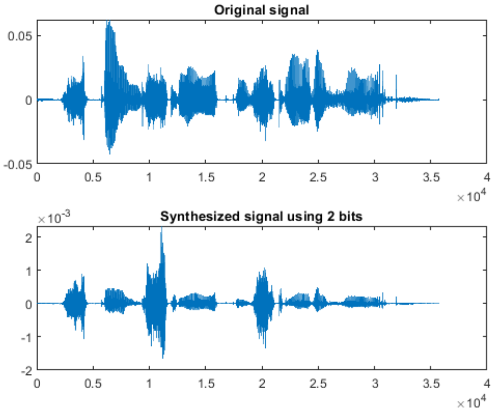

Figure 16: VQ using 2 bits with MSE: 0.000042 on sx72.wav

## 4.2 Our Voices

We also quantized our voices using 6 bits. The synthesized signals have a bad quality, especially Alexandros' voice. There is a lot of noise in our speech signals and we sound like we are whispering (we sound like pilots when they are trying to speak). The MSE of Alexandros' voice is pretty high, hence the bad audio quality, unlike George's voice that has a very low MSE, and his synthesized signal sounds a lot better. This could be due to the training set used to train our quantizers as the male voices are much more similar to George's voice, along with its charasteristics. A great idea for future analysis would be to explain those quantizers further, for example testing how characteristics such as the spoken content and the accent.

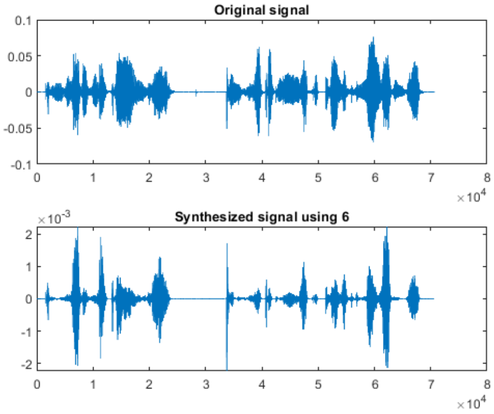

Figure 17: Quantization using 6 bits on George Manos voice with MSE = 0.000100

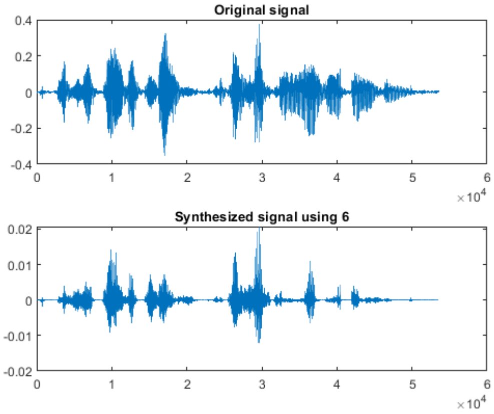

Figure 18: Quantization using 6 bits with on Alexandros Angelakis voice with MSE = 0.002249

### 4.3 LPC Order

The final part is to see how the LPC order affects our results. On the previous experiments, we used an LPC order of 10.

For this part, we will consider the number of bits as constant ($B = 6$). We noticed that when we decrease the LPC order (for example 2), the quality of the synthesized signal is almost the same as the original's one. Even the MSE between them is the lowest we have seen in our experiments, with the value of $MSE = 0.000257$, for the sx391.wav file. On the other hand, as we increase the LPC order (for example 24), the quality of the synthesized signal is getting worse, but the differences do not affect the speech quality that much. The MSE is also getting higher, with the value of $MSE = 0.000281$. Finally, we also tested our quantization with LPC order equal to 64. The result is pretty bad, since there are lot of differences between the original and the synthesized signal, both visually and aurally and their MSE is 0.000290 (the highest we've seen for the sx391.wav file). There is a lot of noise in our synthesized signal and its intelligibility is really low.

This is due to that the higher the LPC order is, the more coefficients we will quantize. As noted on Lab1 where we experimented with LPC order, since it is used both on analysis and synthesis steps, the order would not affect the results of the synthesized signal. This time however since we apply vector quantization to the coefficients, the larger the coefficients vector is, the more coefficients we will quantize. As a result, the information lost by the quantization process will be bigger since we quantize a vector of higher dimensionality, therefore increasing the difference between the original and the quantized vector.

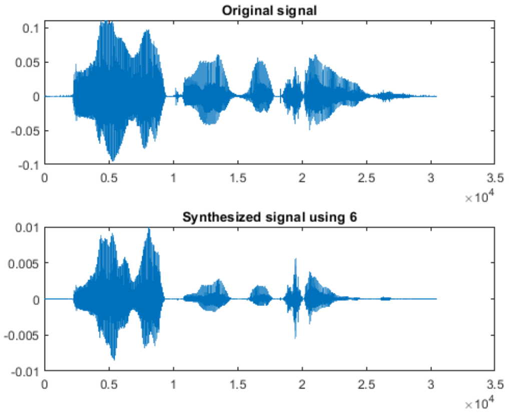

Figure 19: Quantization using 6 bits with LPC order 2 and MSE = 0.000257 on sx391.wav

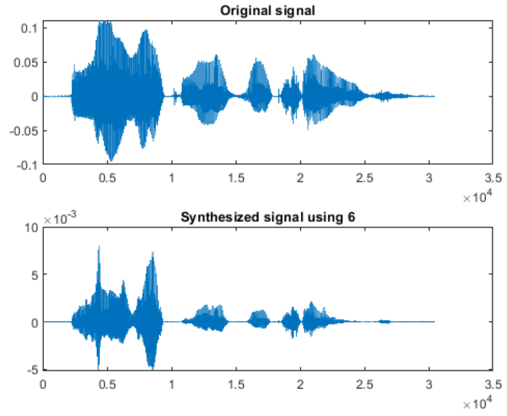

Figure 20: Quantization using 6 bits with LPC order 10 and MSE = 0.000277 on sx391.wav

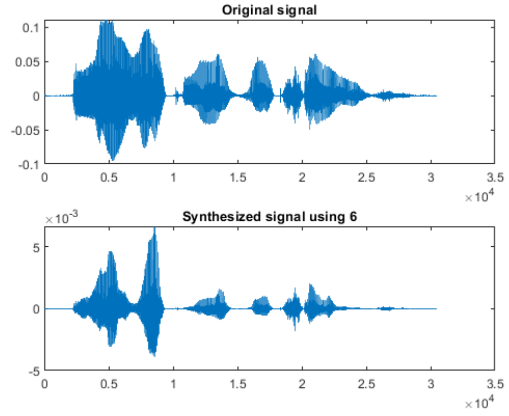

Figure 21: Quantization using 6 bits with LPC order 24 and MSE = 0.000281 on sx391.wav

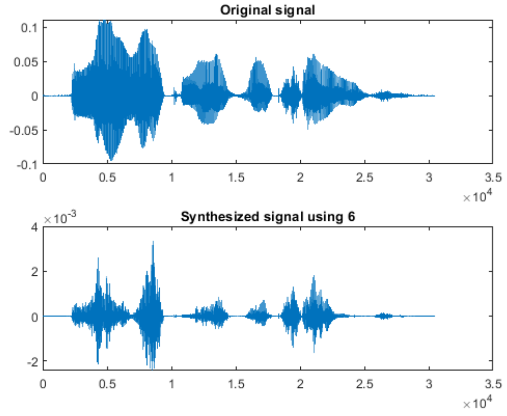

Figure 22: Quantization using 6 bits with LPC order 64 and MSE = 0.000290 on sx391.wav

## 5 Appendix

More philosophy ☺