Assignment 2 : Speech Compression and Quantization

Yann Debain
debain@kth.se

Harsha Holenarasipura Narasanna
harshahn@kth.se

March 6, 2018

Introduction

In this project, we shall examine several quantization methods, compression techniques and derive useful inferences from the observed results. Firstly, quantization methods shall be developed involving the design of uniform scalar quantizer of both mid-rise and mid-tread, followed by vector quantizer and finally, deduce performance metrics from the observed experimental SNR values. Secondly, design and implementation of the adaptive open-loop DPCM, then analyse the bit rate allocation, performance metrics and finally, arrive at the valid conclusion based on the observed results. For this purpose, the speech signal **speech8** will be used.

1 The Uniform Scalar Quantizer

In this task, we implement the simplest quantizer of all : the uniform scalar quantizer (The USQ). The USQ is entirely defined with three parameters :

- n_{bits}, the number of bits used to code one sample. $2^{n_{bits}}$ is the number of output value ;
- m, the mean of the output values ;
- xmax the maximum of the output values ;

In order to make the quantizer independent to the rate we have implemented the USQ with only a scalar division.

In this part, we have simulated a "midrise" and "midtread" quantizer by choosing m = 0 and m = 1.5 respectively.

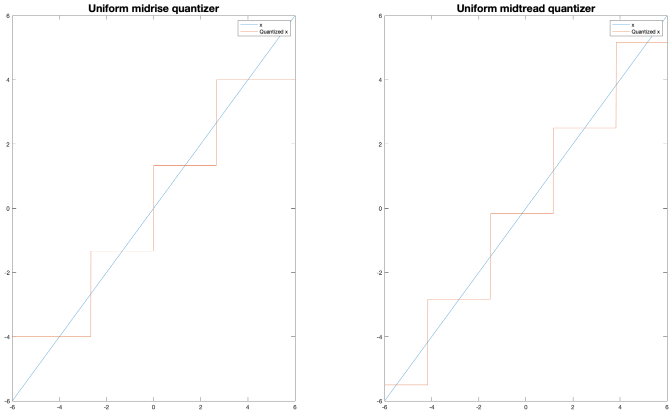

FIGURE 1 – USQ
m = 0 (left) and m = 1.5 (right)

We can see from the figures above that the results are the ones expected : The output of the quantizer is in [-xmax-m ; xmax+m] and the levels are well represented.

2 Parametric Coding of Speech

In this task, we complete our design of the vocoder from the first assignment by finding the right encoding-parameters to transmit the speech with bit rates as low as 2 kbits/second when the parameters are quantized.

The experiments have been made with the parameter alen = 256 and ulen = 128.

2.1 Quantizing the Gain

Here we will design a uniform scalar quantizer for the gain parameter such that the overload distortion is negligible.

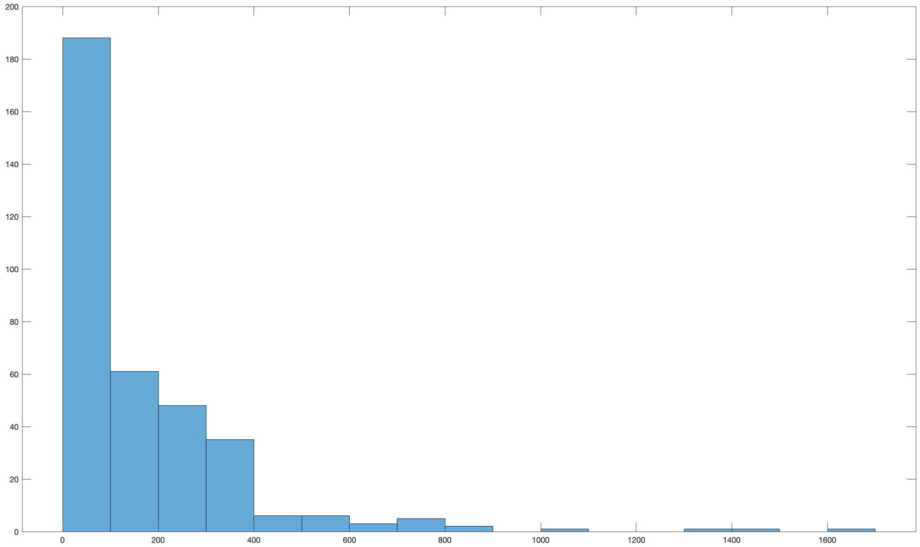

FIGURE 2 – Histogram of the gain parameter

By observing the histogram of the gain parameter we can design the USQ :

- m = 850
- xmax = 850

We modify the parameter n_{bits} until there is no quantization error while listening to the speech. We had to go until $n_{bits} = 8$ to have a clear speech (without quantization error). This means that we have used 8 bits to code the gain for one frame.

Coding one value on 8 bits seems excessive, so we will find another to code the gain with less bit but with the same quality. We will encode the logarithm of the gain to determine if this technique gives better results or not.

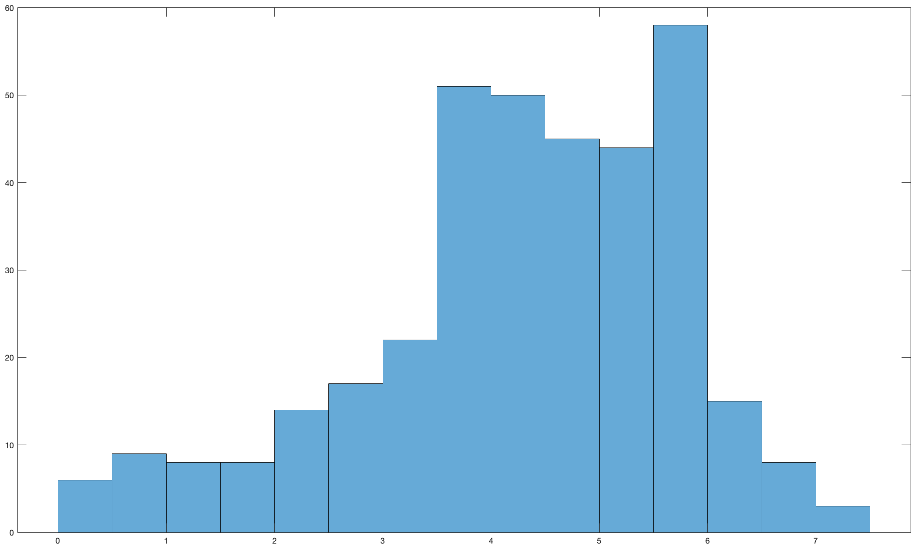

Figure 3 – Histogram of the log-gain parameter

By observing the histogram of the log-gain parameter we can design the USQ :

- m = 3.75
- xmax = 3.75

By repeating the same experiment we find that with $n_{bits} = 4$ we cannot hear the quantization error.

The ear has a better perception in the log-domain than in the linear-domain so coding the gain in the linear-domain goes back to code too much information for a good perception. The same quality of perception can be achieved with fewer resources in the log-domain, therefore the encoding the gain in the log-domain is better.

2.2 Quantizing the Pitch and Voiced/Unvoiced Decision

The Voiced/Unvoiced decision is a binary decision so we will use 1 bit per frame to encode the decision.

Regarding the pitch, both techniques have been tested and did not find much difference between the linear-domain and the log-domain. But to stay coherent with the gain quantization and the work-space of the hearing we will encode the pitch in the log domain.

By running the same experiment that the log-gain quantizer, we find that with $n_{bits} = 5$, we cannot hear the quantization error on the pitch.

2.3 Quantizing the LP parameters

The quantization of the LP parameters is different from what we have done previously. We do not use a uniform scalar quantizer this time but instead use a vector quantizer (VQ). We use an order of 10 to compute the LP parameters. This means that we have to transmit a vector of 10 scalars for each window.

The encoder has already been trained for us, so to encode the vector we have to find in which cluster it belongs to. We will use the L2 distance and choose the cluster that has the minimum distance with the LP vector parameters.

In order to reduce the error between the continuous LP parameters and the quantized LP parameters, we will, in addition, encode the residuals by using the same method (the residuals encoder has already been trained too).

2.4 Optimizing the Bit Allocation

Using the observations from previous questions we design the encoder such that the reconstructed speech has the best quality as possible and we stay under the limit of 2 kbits/sec.

The table below shows how we assign the bits to have the best intelligible vocoder.

Number of bits	bits/window	bits/sample	kbits/sec
Gain (E)	4	0.03125	0.250
Voiced/Unvoiced (V)	1	0.0083	0.0662
Pitch (P)	5	0.0390625	0.3125
LP Parameters (A)	20	0.1656	1.3248
Total	30	0.2344	1.875

FIGURE 4 – Table of Bits Allocation for **speech8**

We can determine the performance of our vocoder by computing the SNR.

$SNR_{dB} = 10log_{10}(\frac{\sigma_x^2}{\sigma_q^2})$ where σ_x^2 is the power of the input signal and σ_q^2 is the power of the quantization error.

We define the quantization noise by $X(n) - \hat{X}(n)$ where $X(n)$ is the input signal and $\hat{X}(n)$ the quantized signal.

We find $SNR_{dB} = -1.1204dB$ which means that the quantization noise is really present in the quantized signal. Despite this quantization noise, the signal is intelligible.

However, it does not make sense to evaluate the SNR here because we want the reconstructed signal as close as possible to the original signal. By synthesizing a new signal we can other reduce the that can make the signal unintelligible (error on the phase, delay, ...). Therefore just computing the SNR is not relevant here.

3 Speech Waveform Quantization

In this part, we design a Uniform Scalar Quantizer (USQ) that is adapted to the speech signal. For that we define $xmax = k\sigma_x$ where σ_x^2 is the variance of the speech signal. This k has to be calculated for every bit rate we choose.

By computing the SNR of the USQ for different value of k, we find that k = 3.03 is the value optimize the SNR for R = 3. We have computed the SNR as before : $SNR_{dB} = 10log_{10}(\frac{\sigma_x^2}{\sigma_q^2})$

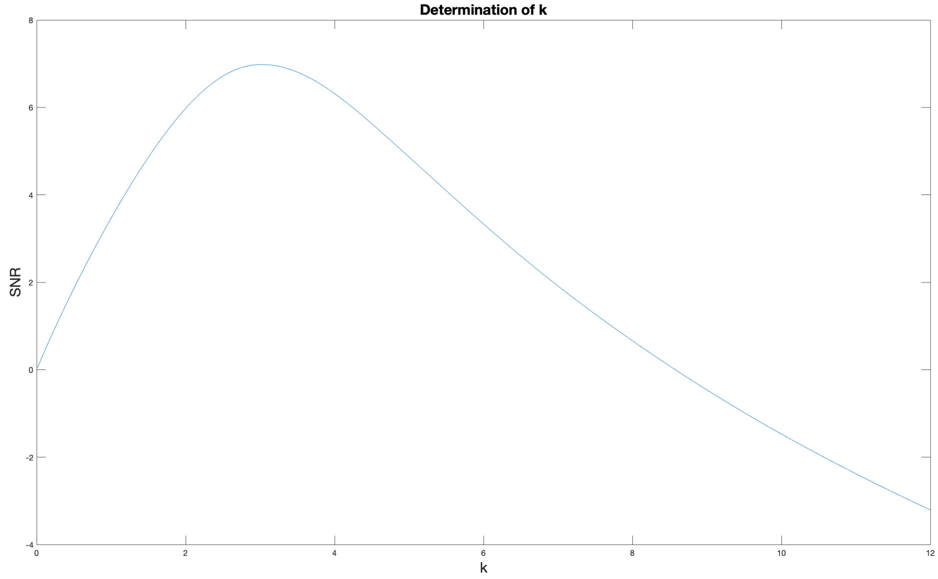

Figure 5 – Determination of k for R = 3

By finding all the k that optimize the SNR for a given rate R we can plot the SNR-Rate curve. We can compare the plot with the theoretical SNR-Rate curve. According to Shannon, it is impossible to be above the theoretical curve. Therefore the experimental SNR-Rate curve will be below the theoretical curve but we want to be as close as possible to this Shannon's limit.

The Shannon's bound is $10log_{10}(2^{2R}) = 6.02 * R$ dB (Assuming we are in high rate and the overload is negligible)

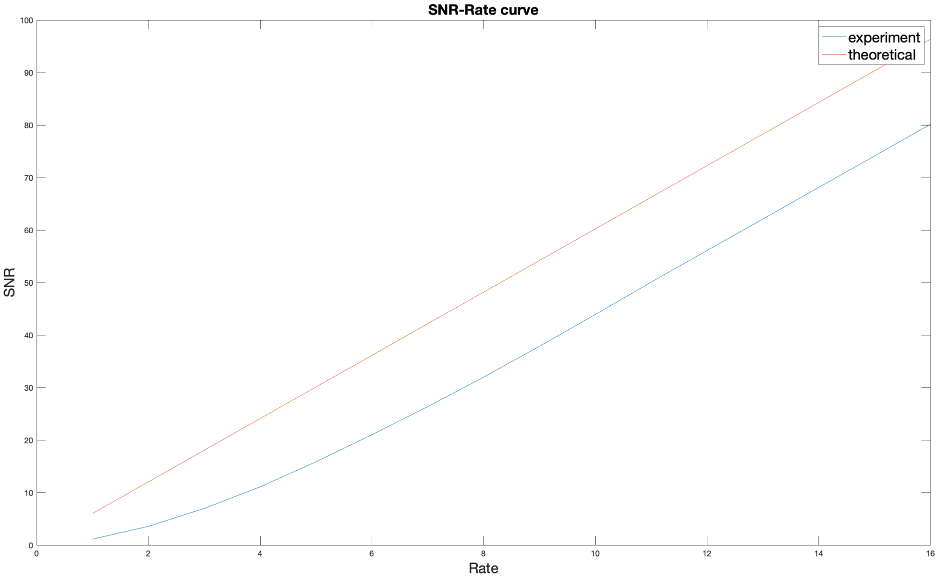

Figure 6 – SNR-rate curve

As predicted, the experimental SNR-Rate curve is below the theoretical one despite the fact that we have optimized the SNR. This difference may be due to the fact that we are using a uniform quantizer despite the fact that the signal does not have a uniform probability distribution function.

We can determine the rate we cannot tell the difference between the original and the quantized signal by listening to both signals. We find that for a rate R = 8bits we cannot hear the difference between both signals. This corresponds to SNR = 30dB which is a typical SNR number for a clean speech.

After analyzing the reconstructed signal, we will focus on the quantization noise. We will listen to the quantization noise for two different rates R = 1 and R = 12. By listening to these signals we can derive the following observations :

- At R = 1, the quantization noise is highly correlated with the original signal, it contains a lot of information (we can still hear the speech from the input).
- At R = 12, we can clearly hear a white noise, this means that there is no correlation with the original signal.

We can derive that the quantization is effective at a high rate, we remove all the correlation in the signal and create a white noise. If we have a constraint on the rate, we have to choose a quantizer that minimizes the distortion on the constraint (for example using a Loyd-Max quantizer with the entropy constraint).

At this point we have only used a midrise quantizer, which means we have a reconstruction level in the origin, thus we have a non-zero level for the silents and low background noises. The reconstruction levels in the origin increase the distortion and, one way to remove that distortion is to use a midtread quantizer.

We can hear that at a low rate the midtread has a better performance than a midrise quantizer. The silents and the low background noises are coded are not reconstructed and the signal is more intelligible.

We can conclude that for a low rate, it is better to use a midtread quantizer to have a low distortion because there is no reconstruction of the silents and low background noises (Check : there is no perceptible change at a high rate).

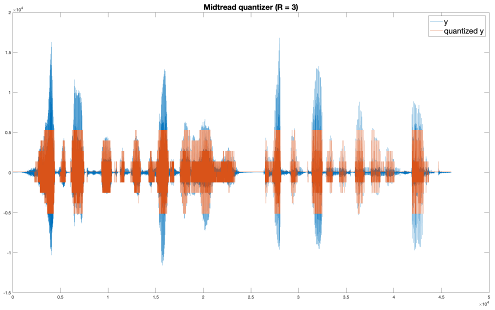

FIGURE 7 – Midtread quantizer (R=3)

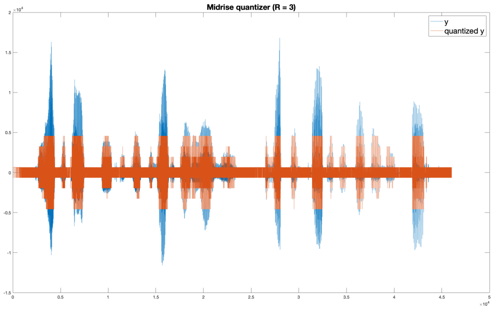

FIGURE 8 – Midrise quantizer (R=3)

4 Adaptive Open-Loop DPCM

In this section, we will study open-loop DPCM. The protocols of the open-loop DPCM is shown in the figure below :

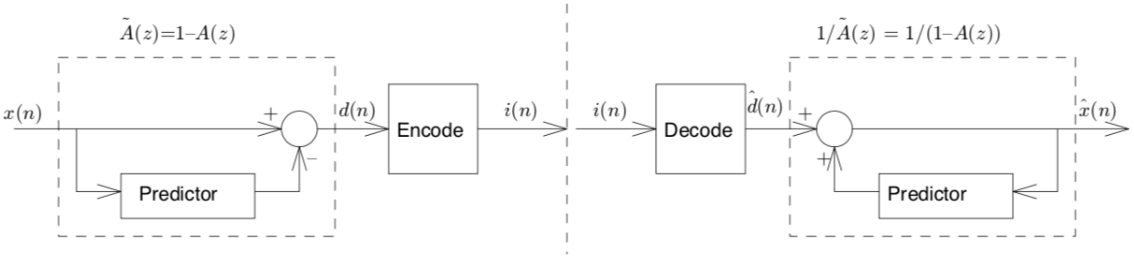

Figure 9 – Open loop DPCM

In order to optimize the rate, we will adapt both the LP coefficients and the gain in a forward fashion in our Open-Loop DPCM.

For all this part we will use the following parameters :

- Analysis frame length = 256 samples
- Update length = 256 samples
- Window function = rectangular (because there is no overlapping)
- Number of bits to quantize the gain = 4
- Number of bits to quantize the residual = 3

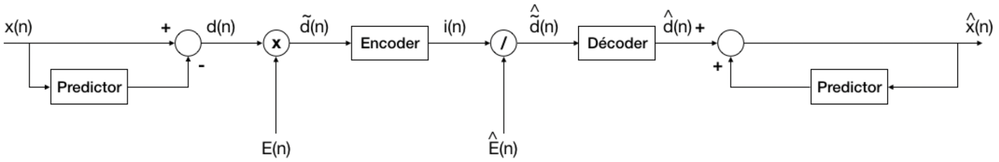

Figure 10 – Adaptive open loop DPCM

We have chosen these parameters to converse the property of stationary in a frame (alen = 256 samples). (To keep things simple we did not have implemented an update length so ulen = alen).

We want the variance of the residuals to be constant (a white noise), we multiply the residual signal by a time-varying gain which is Inversely proportional to the deviation of this signal. We know from **2.1** that it is better to encode the deviation of the residuals in the log-domain. By using our previous observations we will code the gain with 4 bits/frame.
The LP coefficients are adapted with the function *lpc(.)*.

We know that with an open loop DPCM, we have the property of noise shaping so we are expecting that the PSD of a frame of the residuals signal has the same shape that the PSD of the same frame from the input signal.

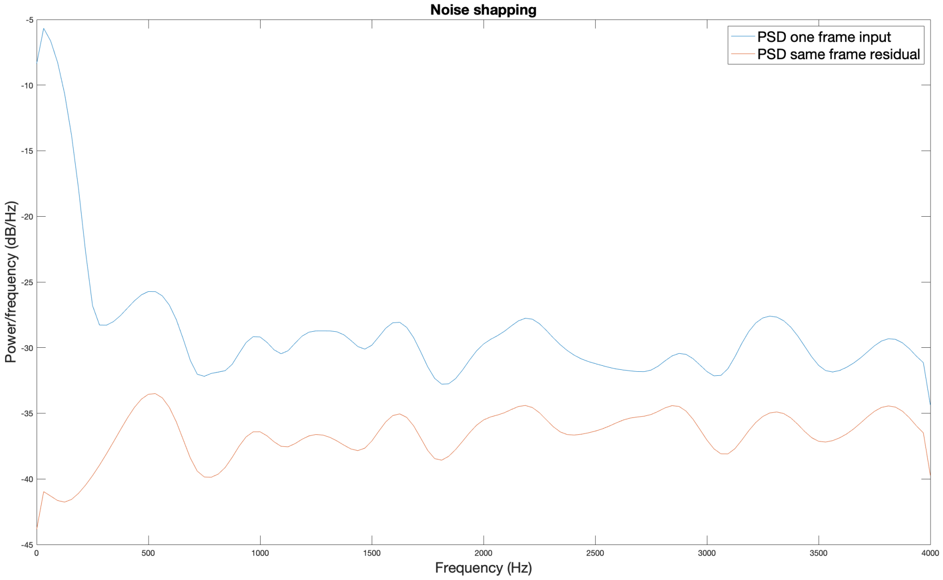

FIGURE 11 – Noise shaping

We can clearly see that the PSD of the residuals is following the shape of the PSD of the input signal. We have this property because $\frac{1}{1-A(z)}$ is the synthesis filter that models the vocal tract.

Number of bits	bits/window	bits/sample	kbits/sec
Quantizer (R)	768	3	24
Gain (E)	4	0,015625	0.125
LP Parameters (A)	20	0.078125	0.625
Total	792	3.0781	24.750

FIGURE 12 – Table of Bits Allocation for open-loop DPCM (on **speech8**)

With an adaptive Open-Loop DPCM, we have a rate of 24.750 kbit/sec which corresponds to a classic rate for the G.726 ADPCM algorithm (the classic rates are 16-, 24-, 32-, or 40-Kbps).

If we assume that the original signal was coded on 64kbit/sec, with our implementation we can achieve a 2.6 :1 compression ratio which can be not negligible for a communication system.

Here we have chosen 3 bits/sample to quantize the residuals but it is possible to use 2 bits/sample and still have an intelligible signal (with a compression ratio 4 :1).

From this implementation, we can measure the SNR and compare it with the SNR of the PCM.

— SNR_{ADPCM} = 2.0168 dB

— SNR_{PCM} = 3.0148 dB

We can see that with the open loop DPCM we have approximately the same SNR than the PCM, one way to increase the SNR is to use a closed loop DPCM but we will lose the noise shaping property. (We observe that we using the real values of A instead of the quantized values increase the SNR, but there is no difference by listening to both speeches)

To increase the SNR, it may be more efficient to have an adaptive step size quantizer Δ than an adaptive gain.

Conclusion

In summary, USQ and VQ with optimal bit allocation were developed. Applying the USQ on a different domain (in this project log-domain) showed improvement in bit size with midtread quantizer marginally leading midrise quantizer in performance at lower bit sizes. Further, VQ offered considerably high SNR. Utilising these quantizers, ADPCM was realized with suitable design parameters. In this case, the removal of associated statistical redundancies was achieved at optimal design parameters with comparable SNR improved over DPCM. In addition, bit rates were computed corresponding to intelligible speech output in each model.