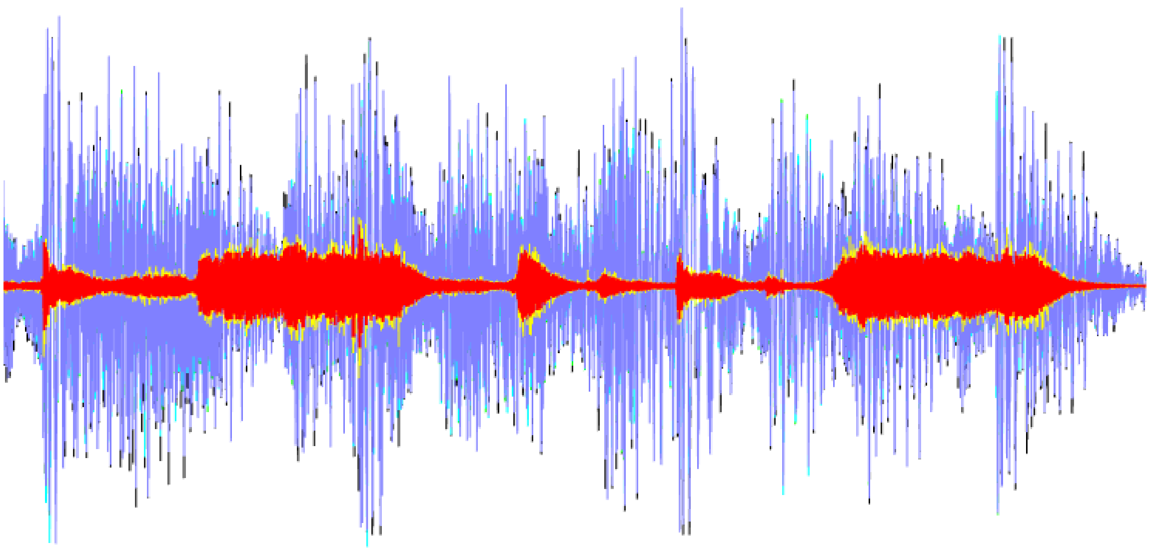

LPC LOSSY COMPRESSION OF SPEECH SIGNALS

ECE 174 - Project 1
Internal Technical Memorandum

Professor :
Ken Kreutz-Delgado

Author :
Pierre MOREAU
U07342256

October 27 2016

Contents

List of Figures

Purpose

Speech and audio signals, represented by sampled data $Y^N = \{y(n), n = 1, 2, ..., N\}$ can be compressed by being quantized to a low bit rate during data transmission in order to obtain faster data transfer rates. Quantization induces distortion in the signal, so this form of compression is said to be "lossy". **The motivation of this project is to reduce the transmitted data needed to construct the original signal using Linear Predictive Coding (LPC).**

Linear predictive coding consists of fitting an infinite impulse response (IR) digital filter to the original audio signal by decomposing the signal into a sum of a few harmonics. Harmonics can then be recreated using only an initial condition and 2 coefficients, reducing the data needed to represent the signal. In order to fit this linear prediction analysis, the signal is decomposed in short enough data blocks for the linear assumption to hold.

Since the signal blocks will probably be composed of more harmonics than I want to model, I will compute the residuals, which should be small enough to heavily quantize them without loosing quality, in order to transmit them and allow a decent (but lossy) reconstruction of the signal.

I will be using Matlab to script this filter, as it allows to process, plot, and listen to the signal easily.

Procedure

Step 1 : choosing a sample

In order to test my filter, I need to choose a sound to model. This sample has to be complex enough to be a representative for the effectiveness of this filter in real life.

Step 2 : split and quantize the signal

When streaming a speech signal, the computation cannot wait for the full audio to be finished, to process it before transmission. This is why I split the signal into small sequential blocks, that will be processed separately. I will perform the quantization on each of those small blocks.

Since the linear prediction is applied in a per block manner, those blocks have to be short enough to assume a stationary linear behavior of the signal. Choosing too small blocks would increase the data to be transferred, as I need to send the harmonics coefficients for each block.

The block size of 160 is chosen because it is small enough that in a 44.1Hz sampled rate audio, there is more than 275 blocks per second, the computation is fast enough to process those blocks rapidly, while the human ear won't be able to notice anything.

I reassemble the signals in the end, in order to listen and plot them.

Quantizing the signal reduces the data needed to represent it. The key is to find a quantization rate reasonably effective, that allows to keep enough dynamic and precision for the signal not to be clear. I test different values, and check by listening to the signals, and by measuring the distortion.

Step 3: IIR filter to deconstruct the signal - learn filter coefficients

To reduce the degradation due to quantization, I fit an infinite impulse response (IIR) (*aka, all poles; aka, autoregressive*) digital filter. This filter maps the original signal $y(n)$ into a sum of $l/2$ harmonics signal $\hat{y}(n)$:

$$y(n) = \underbrace{\sum_{k=1}^{l} a(k)y(n-k)}_{\hat{y}(n)} + e(n) \tag{1}$$

The signal $\hat{y}(n)$ is a linear prediction of $y(n)$ given the l previous time samples $y(n-1), ..., y(n-l)$, assuming that the signal is stationary, which holds onto small data blocks. The residuals $e(n)$ represent the prediction error.

Providing that we have the coefficients $a(n)$, the residuals $e(n)$, and the initial conditions $y(0), y(-1), ..., y(-l+1)$, we can easily reconstruct the audio signal $y(n)$, so this filter is not destructive.

The goal is to learn the most efficient filter coefficients $a(k)$ within each block, as it will make $\hat{y}$ fit y better and forces the sum-of-squares of the residual error $e(n)$ to be as small as possible within each separate block.

Step 4 : calculate the residuals from the linear problem

Since there is mostly no exact solution, I compute the residuals error within each block from equation (1.1).

Step 5 : compute the residuals from the FIR filter

Note that we can rewrite equation (1.1) as the Finite Impulse Response (FIR) (*aka, moving average*) digital filter model :

$$e(n) = y(n) - \hat{y}(n) = y(n) - \sum_{k=1}^{l} a(k)y(n-k) \tag{2}$$

This equation shows that given the filter block's coefficients $a(k)$ one can easily compute the block's residuals $e(n)$ via an FIR moving average digital filtering of the audio signal $y(n)$. If this AR model from step 3 is "good enough", the residuals (*linear prediction*) errors should have much smaller dynamic range than the original signal $y(n)$, and quantizing the $e(n)$ should do less damage than quantizing the original $y(n)$ as done in step 2.

I first need to check that errors computed from the equation (1.2) with a FIR filter are the same as the residuals computed in step 4.

Step 6 : AR filter to reconstruct the signal

To reconstruct the signal, I need to write a small autoregressive (AR) digital filter based on the equation (1.1). It uses the block coefficients learned in step 3, and the residuals from step 5.

I then perform a sanity check to see if this reconstructed signal is identical to the original audio sample.

Step 7 : quantization of the residuals

The residuals can be quantized with the same procedure from step 2, but the quantization parameters have to be chosen carefully.

Let $y_{hat} = \hat{y} + e_q$. This reconstructed signal y_{hat} includes the reconstructed signal from step 6 with quantized residuals e_q. I need to sound-compare it with the original signal y.

Step 8 : comparison

To analyse my results and know if this filter is useful, I need to compare the quality versus the size of the signals y_q and y_{hat}. I listen and measure the distortion of both signals versus the original for various quantization rates, to determine which form of compression, if any, is superior in terms of channel bandwidth utilization.

Step 9 : more compression

To furthermore reduce bandwidth usage, the filter coefficients a can also be quantized. I repeat steps 7 and 8 with the quantized filter coefficients a_q.

Chapter 1

Results

Step 1

In order to test my program, I downloaded an audio signal from the web. I chose a *.wav* file, sampled at $44.1Hz$ which allows to have enough samples per second to divide into small blocks. The sound file is encoded with 24 bits per sample, which is enormous, and will look like a continuous signal from the smaller bit rates that I will work on. Since the audio is quite long, I focus on a 2 second sample taken where there is music and speech, in order to speed up the testing process.

I have purposely chosen a song sample with music and voice, hard to model with this filter, because I know stationary sound will be perfectly modeled but not representative.

Step 2

Let r be the quantization rate, and α the threshold parameter. To help me determine those quantization parameters, I plot the Mean Squared Error (MSE) between the original signal y and the quantized signal y_q for different values of r and α (*see Appendix*). It seems judicious to choose r at the point where the MSE gets stationary, because the error would remain constant while the data usage would grow linearly (r bits per $y_q(n)$).

Looking at Figure 1 on page 11, we can see clearly that for $r < 4$ the MSE grows exponentially, but for $r \geq 5$ the MSE seems stationary. $r = 4$ is a great compromise, because it reduces greatly the data needed to represent each $y_q(n)$, while keeping a minimal distortion.

Choosing a α is done by minimizing the MSE for each r. Figure 3 on page 11 shows minimum α values for $r = 3, 4, 5$. By listening to y_q quantized with those couples of values, I hear that for $r = 3$ there is too much distortion, while for $r = 4$ and $r = 5$ the quantized signal remains very clear and close to the original.

The optimum quantization values for y_q are therefore $r = 4$ and $\alpha = 5.3$. The MSE of the original signal y versus the directly quantized signal y_q is $2.61 \cdot 10^{-4}$.

Step 3

To learn the most efficient filter coefficients $a(k)$ within each block, I determine the least-square estimate solution of the problem $y = A * a$, where y is the block, A a toeplitz matrix that maps the previous data samples onto the vector a of all the l coefficients for the block. This makes $\hat{y}$ fit y better and forces the sum-of-squares of the residual error $e(n)$ to be as small as possible within each separate block.

I assume that $rank(A) \neq 0$, so the least-square estimate solution of the problem is computed using the pseudo-inverse of A.

Looking at the a, if the absolutely 2 smaller coefficients per blocks are close to 0, it means that the order of the model is good enough. This is because there are 2 coefficients needed to represent each component sinusoid of the blocks signal, and if every sinusoid component of the signal in the block has been matched to some coefficients for the block, the remaining coefficients will be 0.

By choosing the model order to be $l = 10$ in the equation (1.1), it is interesting to see that even though the magnitude of the a are comprised in a $\pm 10^1$ range, with the average absolute per-block maximum to be 2.5 ; the average absolute per-block minimum is 0.14. We can assume this is close enough to 0, and that the model order $l = 10$ is good enough.

Step 4 & 5

The residuals e computed in step 4, and e_{fir} computed from the FIR filter in step 5 are identically the same. Which is correct because they have been computed from equation (1.1), and equation (1.2) which is a rewrite of equation (1.1). The MSE computed between those 2 residuals is in 10^{-33} (*such an error results from numerical representation precision in Matlab*).

Those residuals are between $\pm 10^{-7}$ and $\pm 4 \cdot 10^{-1}$, with an absolute average of $2.42 \cdot 10^{-2}$. However, the original signal values are between $\pm 10^{-6}$ and $\pm 10^{0}$, with an absolute average of $1.83 \cdot 10^{-1}$. It looks like the residuals have approximately 10 times less the dynamic range of the original signal. This is a great result, as the quantization of the residuals shouldn't do as much damage as the quantization of the original signal.

Step 6

The reconstruction of the signal with the filter I wrote in step 6, using the residuals computed in step 5, should be identical to the original signal. This is the application of the equations. To do a sanity check, I computed the MSE between the original signal and the reconstructed signal, which is close to zero (10^{-29} *results from numerical representations in Matlab*).

Step 7

I use the same process as in step 2 to determine the best values for r and α. I found different interesting couples of quantization parameters on figures Page 12, compared their MSE of the original residuals e vs the quantized residuals e_q, and listened to the resulting signal reconstructed with the quantized residuals y_{hat}.

r	α	MSE from quantization	Perceived sound quality of reconstructed signal
1	0.6	$2.67 \cdot 10^{-4}$	quite clear but dull
2	1.25	$1.15 \cdot 10^{-4}$	clear enough
3	1.83	$4.40 \cdot 10^{-5}$	clear
4	2.9	$1.49 \cdot 10^{-5}$	clear

Table 1.1: Selection of quantization parameters for the residuals

From Table 1.1, we have chosen $r = 2$ and $\alpha = 1.25$ to quantize the residuals. We can clearly see that quantizing the residuals doesn't affect the sound quality as much as quantizing it directly, as we can quantize to lower rates without deteriorating the signal much.

Step 8

By sound comparing both directly quantized signal y_q, and the signal reconstructed from quantized residuals y_{hat}, I found that the directly quantized signal y_q is less pleasant to listen. Even if the signal remains quite clear, the distortions affect greatly the quality. The LPC filtered signal y_{hat} on the other hand, is more pleasant to listen, even if the MSE is greater (*see Table 1.2*). It's interesting to see that the MSE, which is a mathematical measure of the distortion, doesn't correlate to the subjective percept quality [1].

Those results prove that this filter allows a heavier rate of quantization while keeping a clear sound quality, and the figures Page 13 shows that quantizing the residuals doesn't affect the signal as much. This is very interesting as heavier rates of quantization saves bandwidth usage. With this filter, y_{hat} uses almost half the bandwidth usage as y_q for an even better quality. But we still have to take into account the coefficients and add them to bandwidth usage. I will try to quantize them in the next step.

Signal	r	MSE of signal vs original	Perceived sound quality
y_q	4	$2.61 \cdot 10^{-4}$	good : clear enough, some high frequency distortion
y_{hat}	2	$5.2 \cdot 10^{-3}$	better : clear enough, low frequency distortion

Table 1.2: Quantization methods comparaison

[1] Prof. Bhaskar Rao at UCSD has done and published research on this issue.

Step 9

In order to, furthermore, reduce bandwidth usage, I need to quantize the coefficients a for each block, as they are also sent and received. I repeat the quantization steps from step 2, but I let α be big in order to fix the quantization value in Figure 9 on page 14. This is done to favor the big filter coefficients for each block, as they are linked to the most dominant part of the signal.

I get the couples of values shown on Figure 11 on page 14. But those parameters are only chosen in regard of the discrepancy between the original coefficient a and the quantized coefficients a_q. When I listen to the reconstructed the signal y_{hat*} using the quantized residuals e_q and the quantized coefficients a_q, with those parameters, it isn't intelligible anymore.

I computed the MSE between the original signal y and the reconstructed y_{hat*} with different values of r and α for the quantization of the coefficients a. The results in Figure 12 on page 15 shows that the filter is unstable and blows up if $\alpha > 0.2$. I listened to y_{hat*} reconstructed with all the quantization values in Figure 12 for which the filter is stable (*ie* $MSE \approx 0.06$), and there are no notable differences, the quality is always poor. The best quantization parameters for the coefficients a seems to be $r = 2$ and $\alpha = 0.1$. With those parameters, the filter is stable, and the bandwidth usage is reduced. However the dynamic range of the signal is lost, and the quality is poor.

Figure 13 on page 16 shows the original signal y, the directly quantized signal y_q, the reconstructed signal with quantized residuals y_{hat}, the reconstructed signal with quantized residuals and coefficients y_{hat*}, and the quantized residuals e_q. It shows that y_q and y_{hat} provide both a good fit to y, but that y_{hat*} consists mainly of the residuals. This is because I had to choose $\alpha = 0.1$ to quantize the coefficients. Those quantization parameters are only chosen because they keep the filter stable. Moreover, with $r = 2$, only 4 distinct values are allowed for the coefficients, which means that the order of the filter is levelled back to 4. Obviously, the coefficients cannot be quantized using this method, and further study and research is necessary in order to quantize the a.

Chapter 2

Conclusion

The goal was to reduce bandwidth usage using linear predictive coding (LPC), in order to heavily quantize the signal without losing too much detail in the distortion. The filter I wrote managed to map the signal onto harmonics, and left only small residuals to quantize. I was able to quantize those residuals to a very low bitrate without losing quality, and still having a better sounding signal than the directly quantized one.

However, some filter coefficients are also transmitted and quantizing those coefficients induced some instability in the filter. A quantization was possible, but the sound quality was poor. More study and research is necessary to quantize the coefficients, as a direct quantization method isn't efficient enough.

Table 2.1 compares all signals quality versus their bandwidth usage. With a right quantization method for the coefficients, the bandwidth usage could be divided by 10. Without a stable discretized AR digital filter, the direct quantization method still produces the best result, by dividing bandwidth usage by 6 while keeping a good sound quality.

Signal	bits per block	MSE of signal vs original	Perceived sound quality
y	3840	0	perfect
y_q	640	$2.61 \cdot 10^{-4}$	good
y_{hat}	320 + 640 (64 bits coefficients)	$5.2 \cdot 10^{-3}$	great
y_{hat*}	340	$5.97 \cdot 10^{-2}$	poor

Table 2.1: Quantization methods bitrates versus quality comparaison

Appendices

Step 2

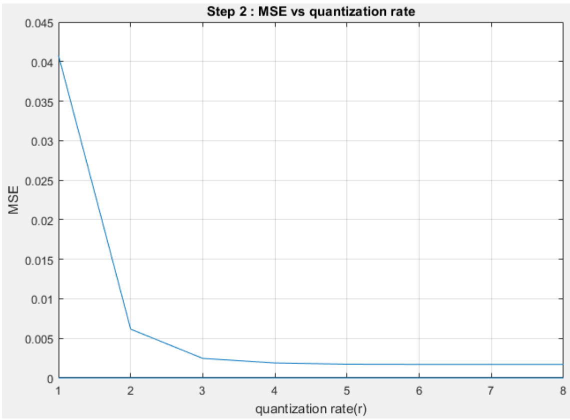

Figure 1: MSE vs quantization rate r

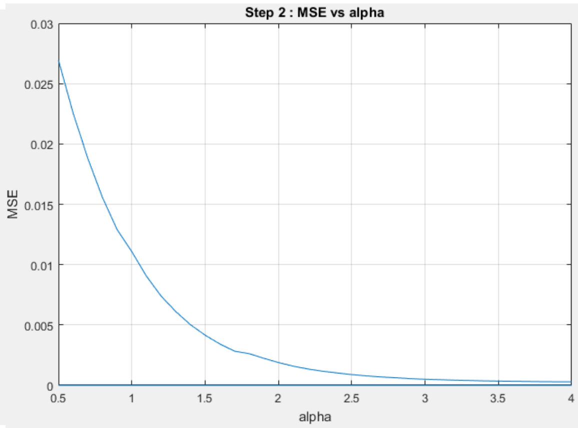

Figure 2: MSE vs α for $r = 2$

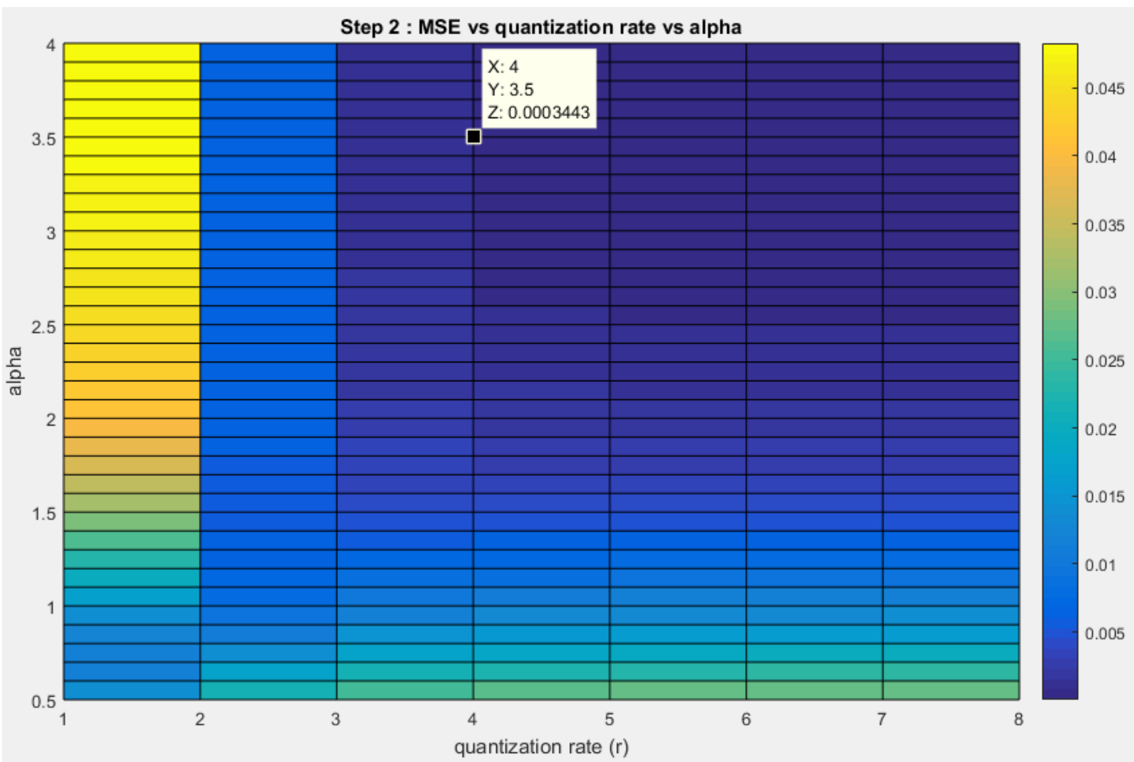

Figure 3: MSE vs quantization rate r vs α
The most interesting values for α and r is shown.

Step 7

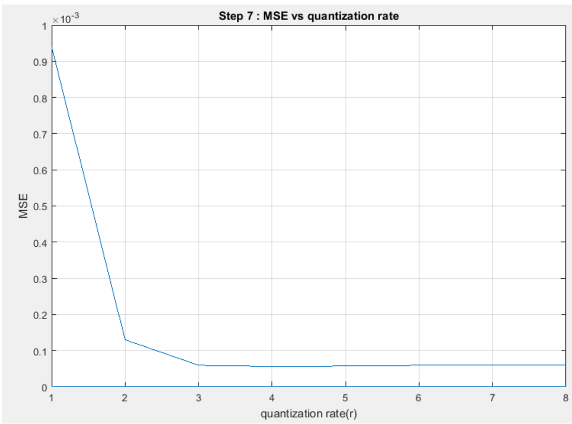

Figure 4: MSE vs r for $\alpha = 1.5$

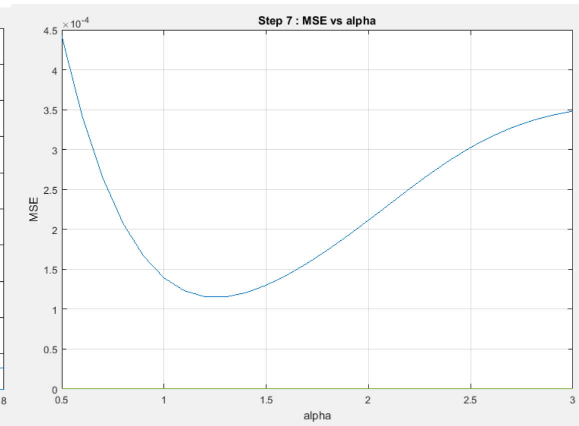

Figure 5: MSE vs α for $r = 2$

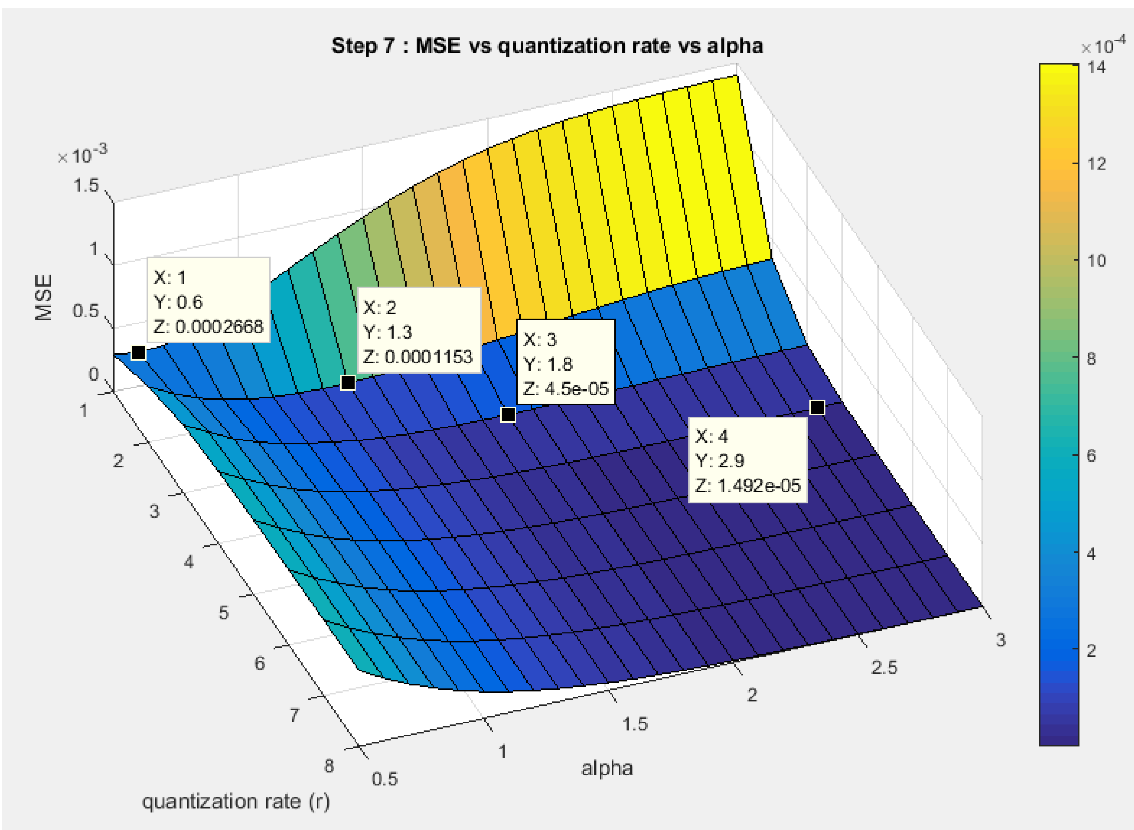

Figure 6: MSE vs r vs α
The most interesting values for α and r are shown.

Step 8

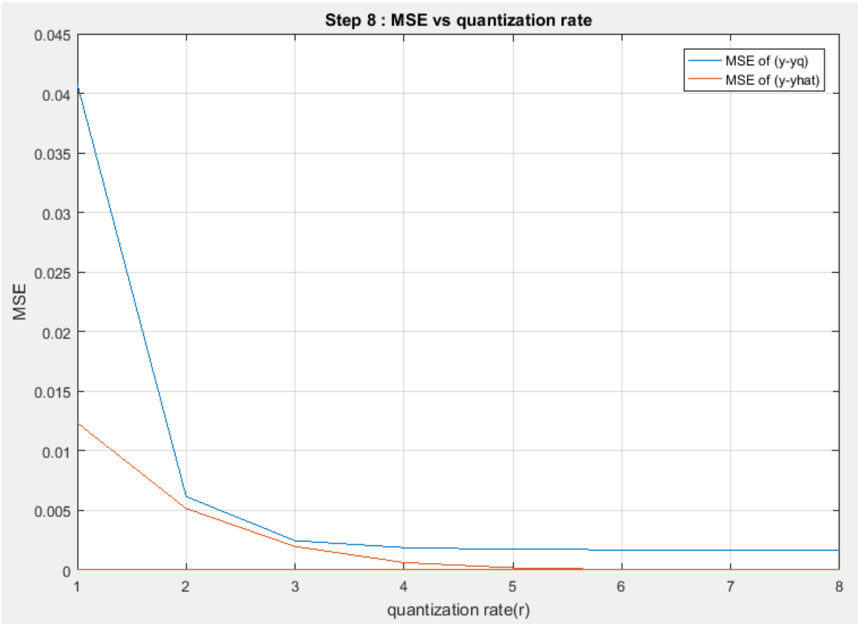

Figure 7: MSE vs r

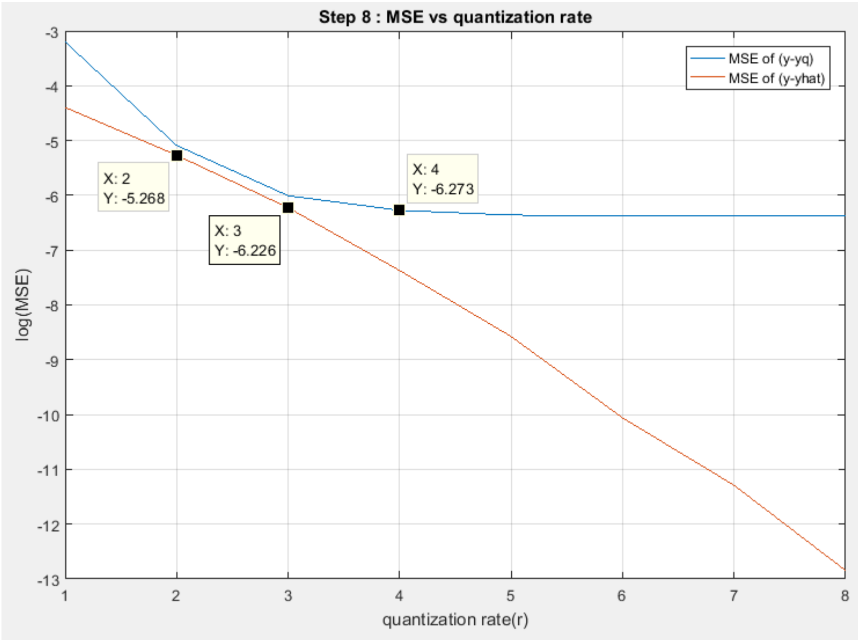

Figure 8: $log(MSE)$ vs r

Step 9

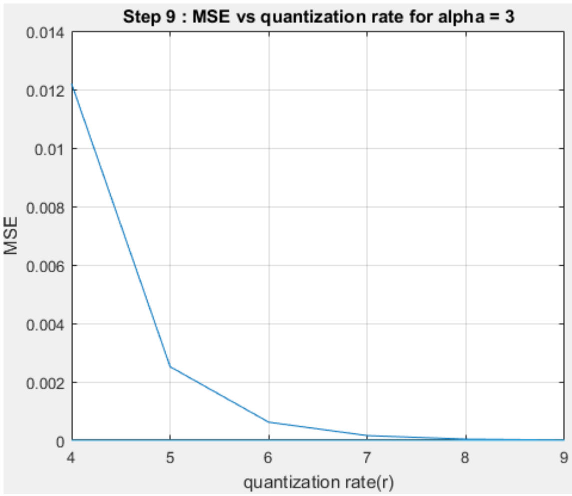

Figure 9: MSE vs r for $\alpha = 3$

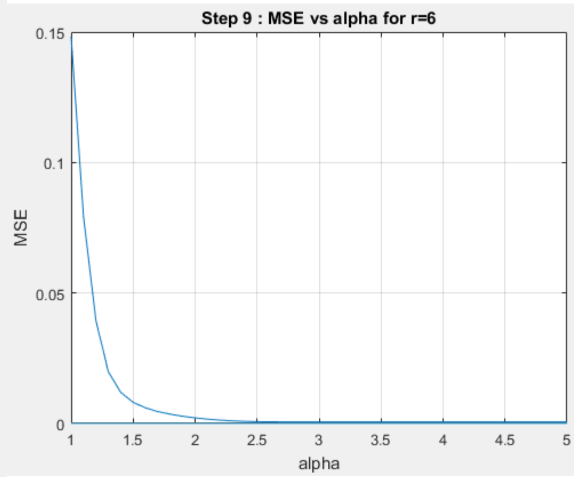

Figure 10: MSE vs α for $r = 6$

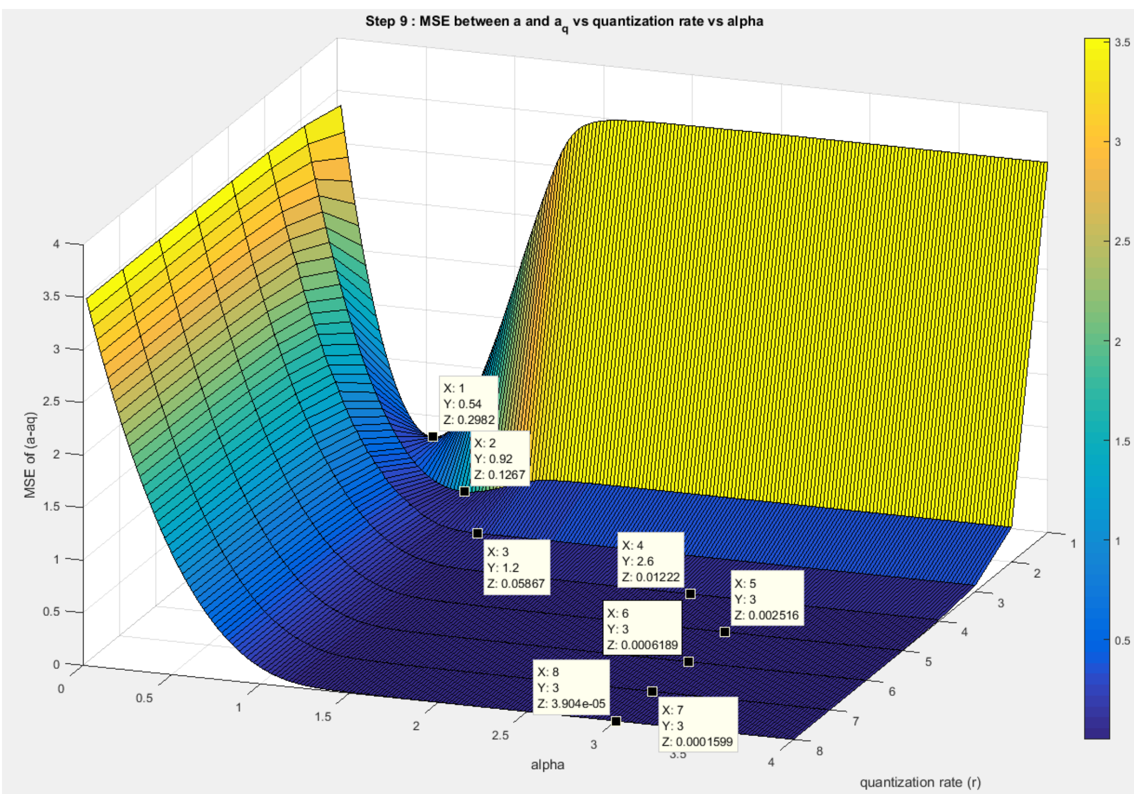

Figure 11: MSE vs r vs α
The best values for α are shown for each r.

alpha \ r	1	2	3	4	5	6	7	8
0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
0.1	0.4234	0.0597	0.0597	0.0597	0.0597	0.0597	0.0597	0.0597
0.2	Inf	3.5702e+06	0.0597	0.0597	0.0596	0.0596	0.0596	0.0596
0.3	Inf	6.3969e+53	1.5345e+41	2.6548e+06	583.2830	49.4389	1.1281	0.5988
0.4	Inf	Inf	1.4651e+96	1.6605e+100	1.1079e+71	5.9970e+58	3.0661e+59	1.5425e+56
0.5	Inf	Inf	Inf	8.9659e+246	4.2950e+257	8.9806e+216	4.4007e+223	6.2644e+217
0.6	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.7	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.8	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
0.9	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.1	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.2	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.3	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.4	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.5	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.6	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
1.7	Inf	Inf	Inf	Inf	Inf	Inf	Inf	1.0689e+238
1.8	Inf	Inf	Inf	Inf	Inf	Inf	Inf	6.6839e+94
1.9	Inf	Inf	Inf	Inf	Inf	Inf	Inf	6.6839e+94
2	Inf	Inf	Inf	Inf	Inf	Inf	Inf	6.6839e+94
2.1	Inf	Inf	Inf	Inf	Inf	Inf	Inf	6.6839e+94
2.2	Inf	Inf	Inf	Inf	Inf	Inf	Inf	6.6839e+94
2.3	Inf	Inf	Inf	Inf	Inf	Inf	Inf	6.6839e+94
2.4	Inf	Inf	Inf	Inf	Inf	Inf	Inf	6.6839e+94
2.5	Inf	Inf	Inf	Inf	Inf	Inf	Inf	6.6839e+94
2.6	Inf	Inf	Inf	Inf	Inf	Inf	Inf	6.6839e+94
2.7	Inf	Inf	Inf	Inf	Inf	Inf	Inf	6.6839e+94
2.8	Inf	Inf	Inf	Inf	Inf	Inf	Inf	6.6839e+94
2.9	Inf	Inf	Inf	Inf	Inf	Inf	Inf	6.6839e+94
3	Inf	Inf	Inf	Inf	Inf	Inf	Inf	6.6839e+94
3.1	Inf	Inf	Inf	Inf	Inf	Inf	Inf	6.6839e+94
3.2	Inf	Inf	Inf	Inf	Inf	Inf	Inf	6.6839e+94
3.3	Inf	Inf	Inf	Inf	Inf	Inf	Inf	6.6839e+94
3.4	Inf	Inf	Inf	Inf	Inf	Inf	Inf	6.6839e+94
3.5	Inf	Inf	Inf	Inf	Inf	Inf	Inf	6.6839e+94
3.6	Inf	Inf	Inf	Inf	Inf	Inf	Inf	6.6839e+94
3.7	Inf	Inf	Inf	Inf	Inf	Inf	Inf	6.6839e+94
3.8	Inf	Inf	Inf	Inf	Inf	Inf	Inf	6.6839e+94
3.9	Inf	Inf	Inf	Inf	Inf	Inf	Inf	6.6839e+94
4	Inf	Inf	Inf	Inf	Inf	Inf	Inf	6.6839e+94

Figure 12: MSE of original signal y and reconstructed signal with quantized coefficients and residuals y_{hat*}, vs r vs α
The filter is unstable and blows up if $\alpha > 0.2$. The best quantization parameters for the coefficients a are $r = 2$ and $\alpha = 0.1$.

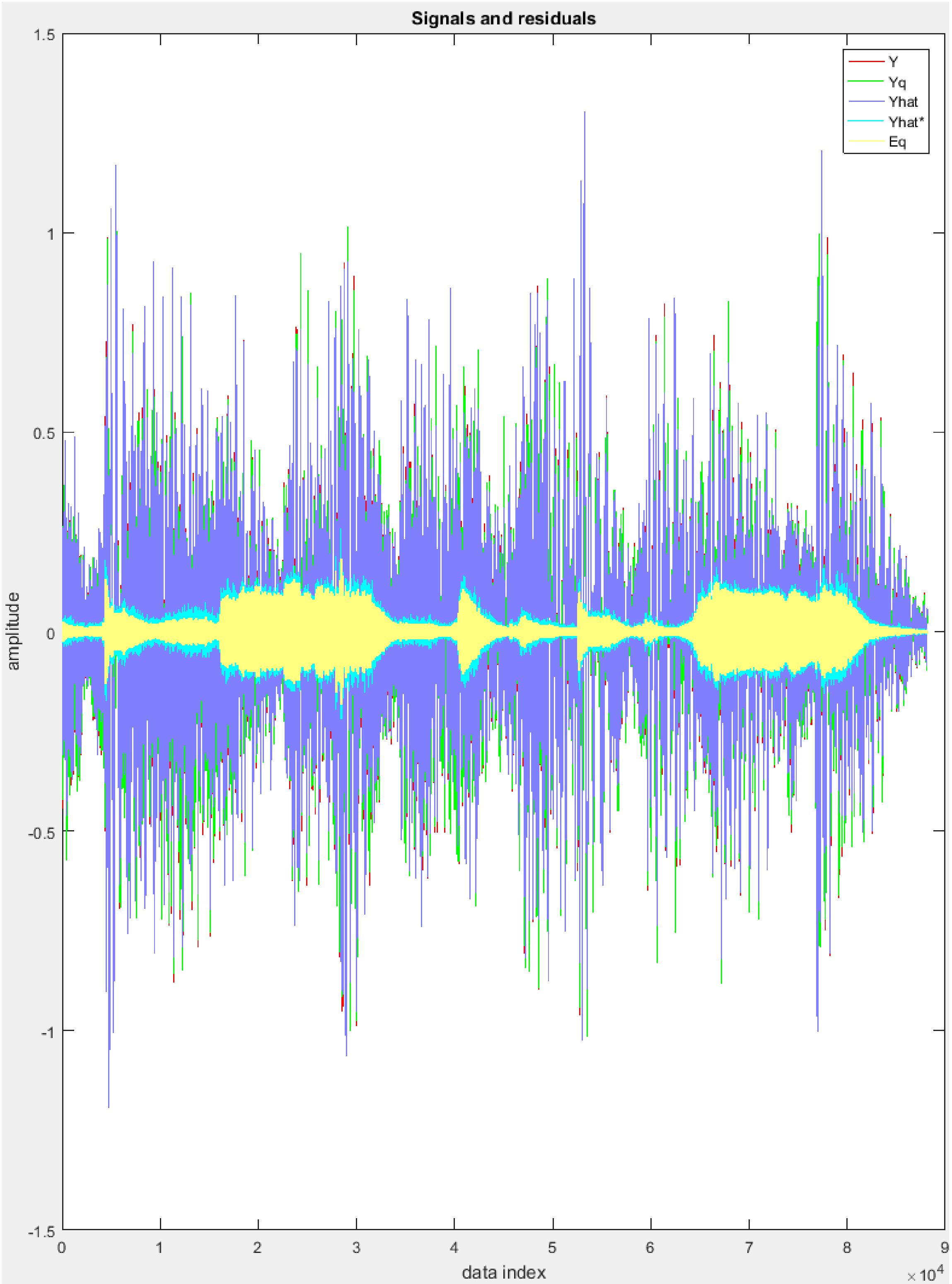

Figure 13: Amplitude of different signals and residuals