Voice Codec Implementation Using MATLAB

Presented To: Dr. Samy Soliman

Done By:

Aml Tarek	202200126
Mohammad Mahmoud	202200438
Youssef Allam	202200286

Contents

Objective

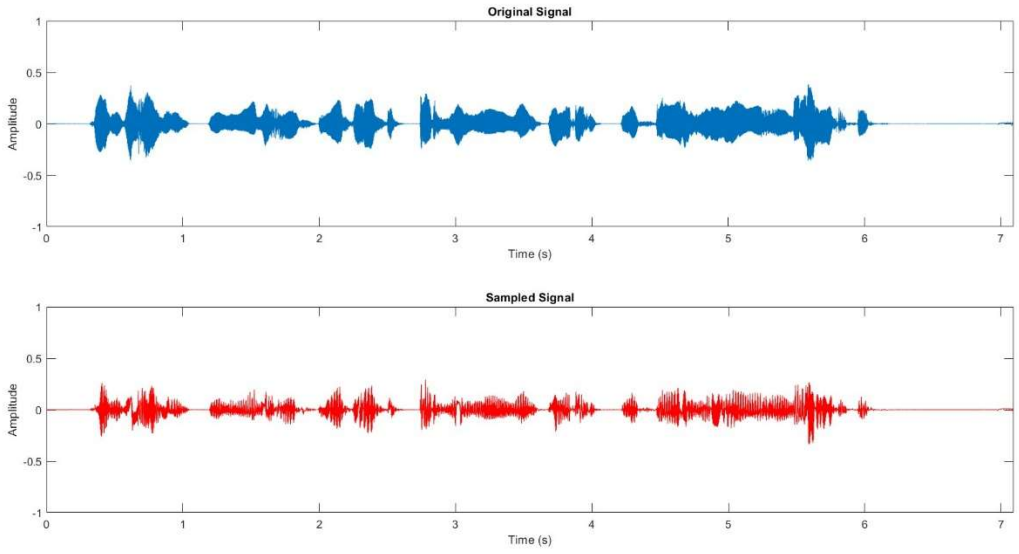
The goal of this project is to create a voice codec software using MATLAB. The software will consist of 4 steps.

1. Sampler
2. Quantizer
3. Encoder
4. Decoder

Between the encoder and the decoder will be the transmission channel where there could be noise added to the signal.

Sampler

The goal of the sampler is to take an oversampled analog signal such as an audio file as well as the sampling frequency and perform natural sampling on the signal. Natural sampling is where a pulse function traces out the envelope of a signal for some time per sample.

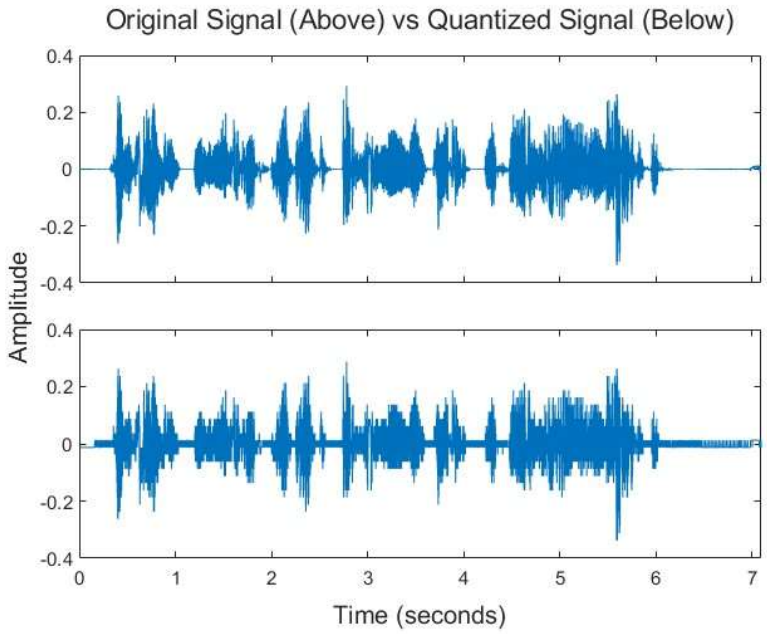


Quantizer

The quantizer takes input parameters for the number of quantization levels(L), maximum peak value (mp), setting (indicates midrise or midtread), amplitude vector, time vector. The

quantizer outputs the quantized signal, the corresponding bitstream and the average quantization error.

The quantizer then loops over each sample, finds the nearest level, and stores the quantized value. The quantized values are then converted to their binary representation and returned sent as the bitstream to the encoder. The function then plots the quantized and original signals, as well as calculates the average quantization error.



Encoder

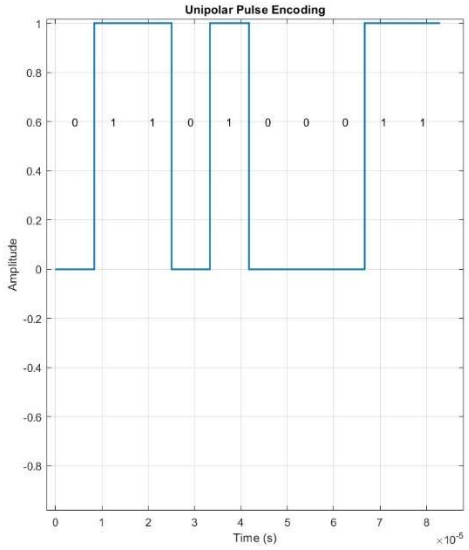
The 0 and 1 bits generated by the quantizer will need to be translated into line codes. We will implement 2-line codes Polar NRZ, and Unipolar NRZ.

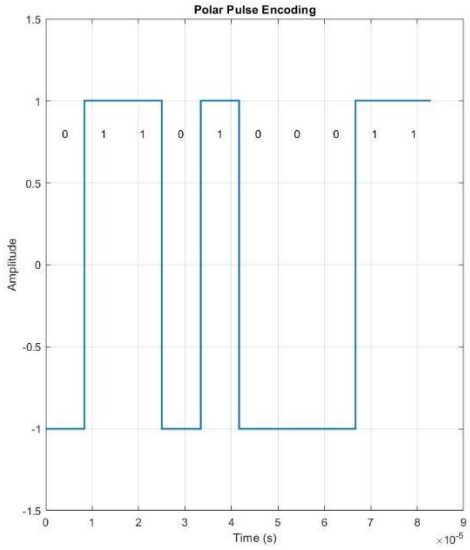
In unipolar NRZ bit '1' is translated into an amplitude of 1 and bit '0' is translated to an amplitude of 0.

In polar NRZ bit '1' is translated into an amplitude of 1 and bit '0' is translated to an amplitude of -1.

For each bit we will generate its respective pulse.

Below is a testcase for the bitstream '0 1 1 0 1 0 0 0 1 1'

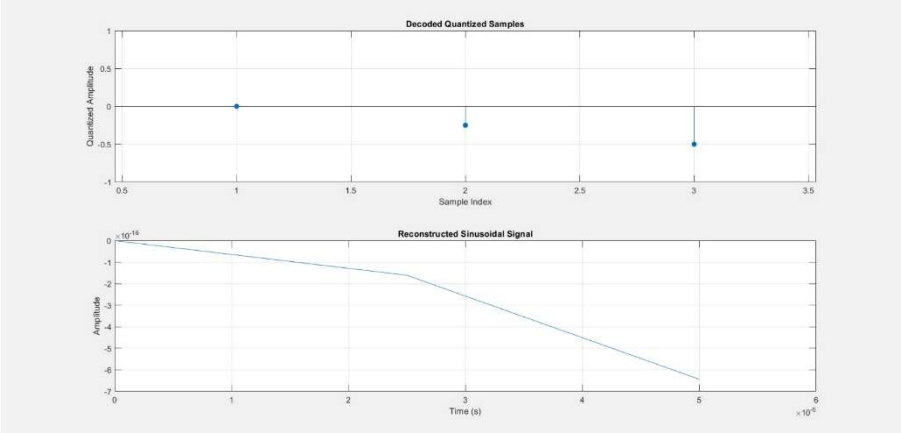




Decoder

The first part of the decoder will have to receive the PCM signal and from it extract the bits. The bits will be extracted by determining the average of the signal over each bit and determining whether it's a 0,1, or -1 and based on the line code used we will translate the PCM signal back into a stream of 1s and 0s.

After getting the stream of 1s and 0s we use the bit duration to determine the number of bits per sample then we divide the bitstream into subarrays each of which is equal to the number of bits per sample. Finally, we convert the number for each sample from binary to decimal based on whether the quantizer is a midrise or mid tread quantizer.



Test 1

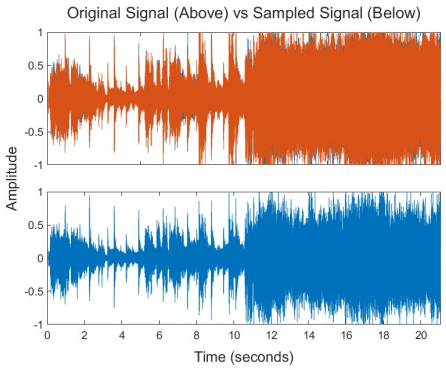
Sampler

We will be testing our entire script over an audio sample using the following parameters

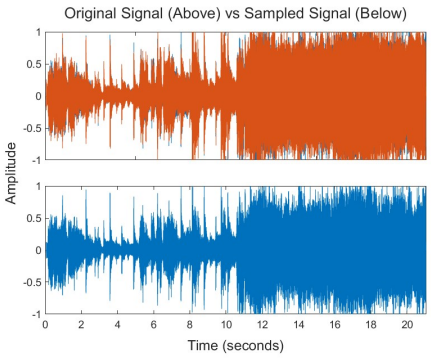
- Fs=5KHz, 20KHz, 40KHz
- L=4,8.64

The peak of the signal will be determined from the audio file itself.

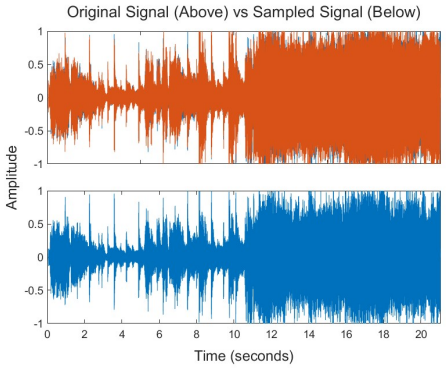
We first start by sampling our signal at 3 different Frequencies different frequencies with the results of the sampler shown below



Sampled at 5KHz



Sampled at 20KHz



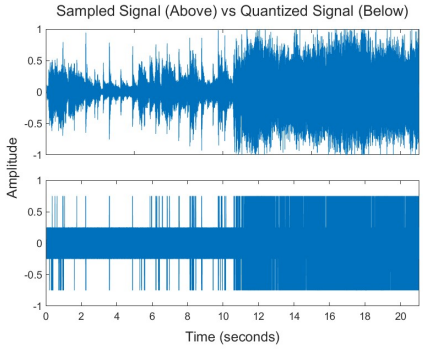
Sampled at 40KHz

We notice that as the sampling frequency increases the sampled function looks more like the original function

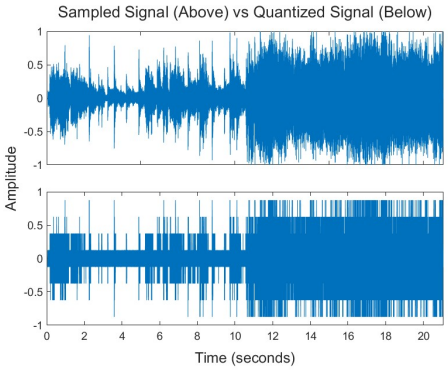
Quantizer

The figure below shows the output of the quantized signal compared to the output of the sampled signal.

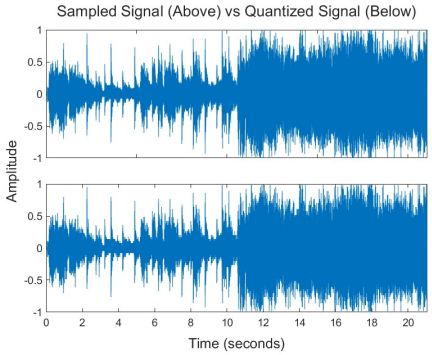
Sampled at 5000Hz



Fs=5000Hz
L=4



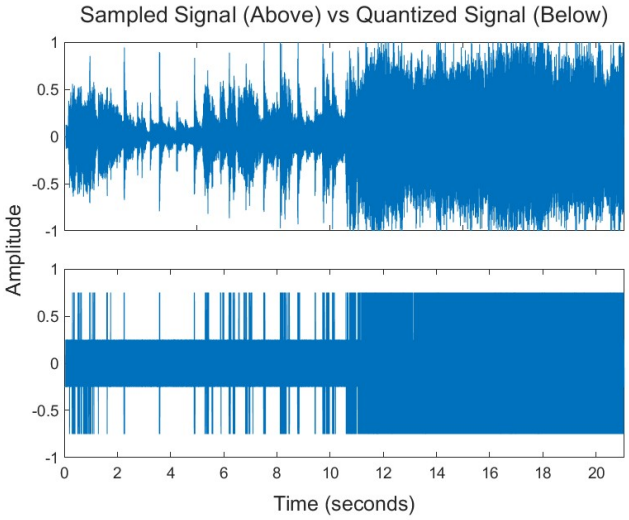
Fs=5000Hz
L=8



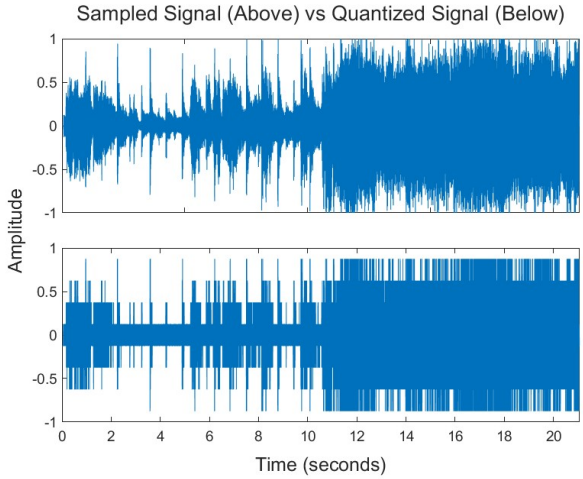
Fs=5000Hz
L=64

As we expect when the number of levels increases the quantization error decreases and the quantized signal becomes a more accurate representation of the sampled signal.

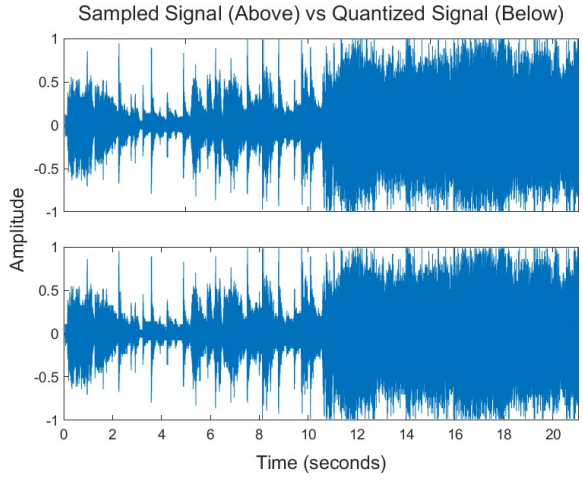
Sampled at 20000Hz



Fs=20000Hz
L=4



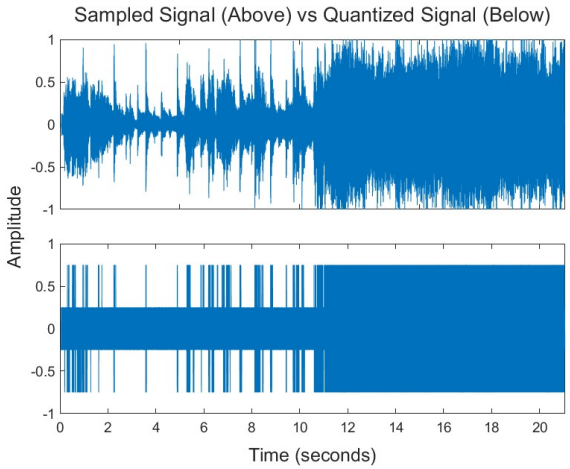
Fs=20000Hz
L=8



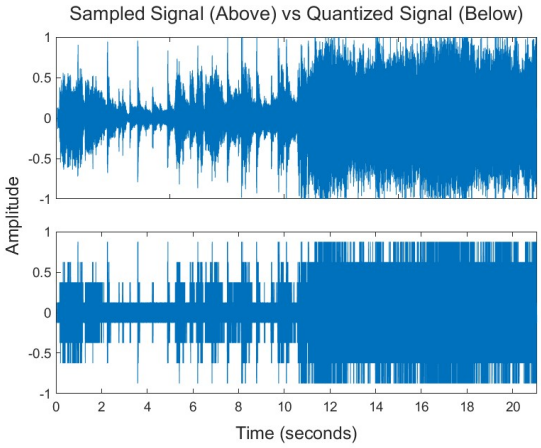
Fs=20000Hz
L=64

We notice that the quantized signal is closer to the sampled signal as the number of levels increases just like last time however this time the sampled signal itself is closer to the original signal as it was sampled at a higher frequency.

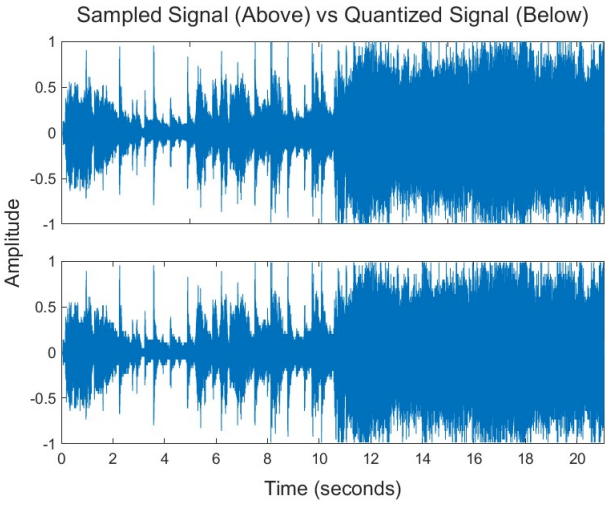
Sampled at 40000Hz



Fs=40000Hz
L=4



Fs=40000Hz
L=8



Fs=40000Hz
L=64

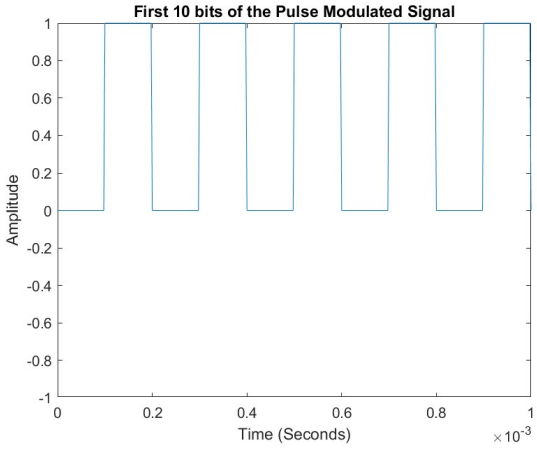
In these finals set of figures the sampled signal is as close as possible to the original signal and the quantized signals get closer to the sampled signal as the number of levels increases

Encoder

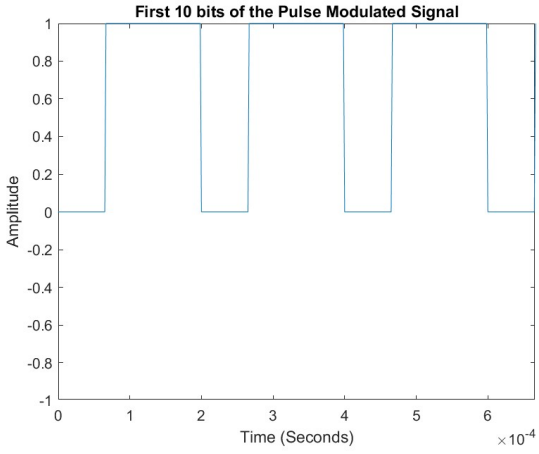
We next pass the signal through 2 encoders, a polar NRZ encoder and a unipolar NRZ encoder. In the below figures we will show the first 10 bits for each PCM signal. In unipolar NRZ coding the bit 1 is translated to a pulse of amplitude 1 for the entire duration Tb while the bit 0 is translated to a pulse of amplitude 0 for the entire duration Tb. Polar NRZ is similar except for the fact that 0 translates into a pulse of amplitude -1.

Sampled at 5000Hz

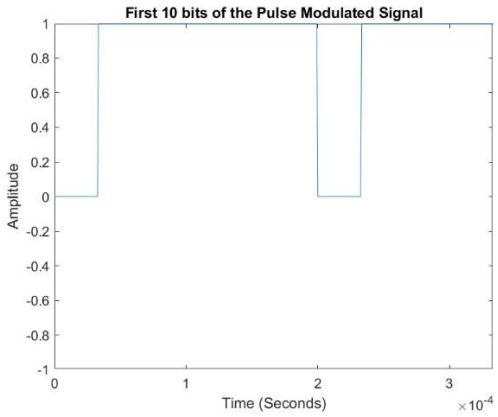
Unipolar NRZ



Fs=5000Hz
L=4



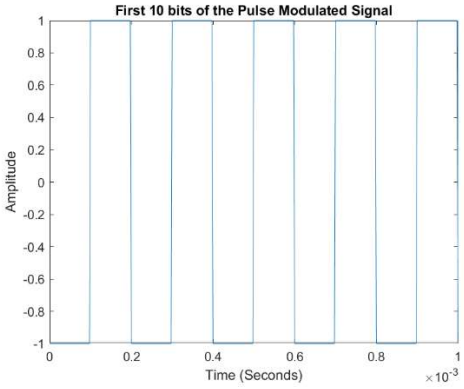
Fs=5000Hz
L=8



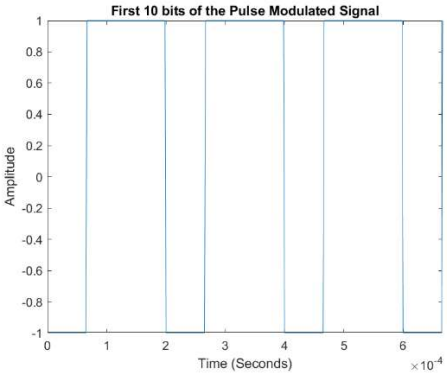
Fs=5000Hz
L=64

We notice that as the number of levels increases the binary number increases, which makes sense, since the number of levels between any 2 points increases as the number of levels increases which results in the equivalent level having a higher binary representation.

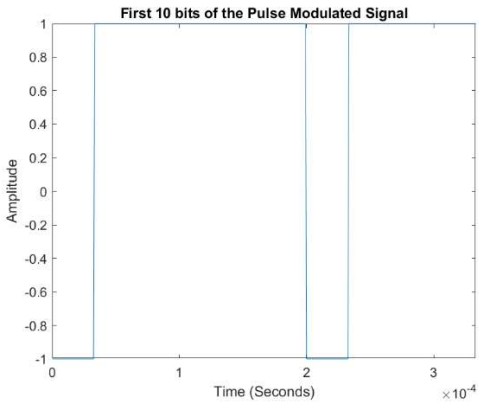
Polar NRZ



Fs=5000Hz
L=4



Fs=5000Hz
L=8

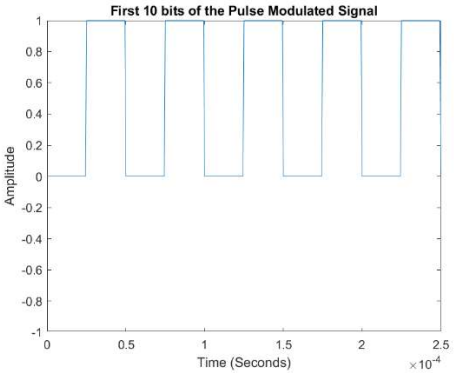


Fs=5000Hz
L=64

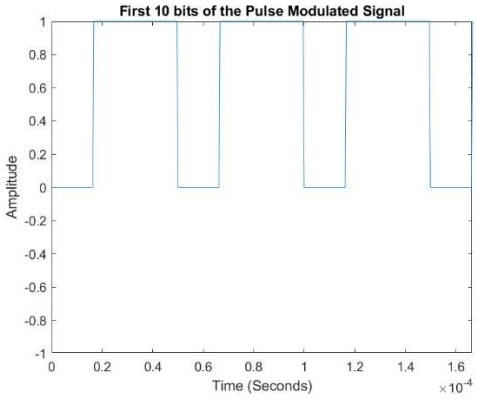
We notice similar results for the polar NRZ with the only exception of points that had a value of 0 now have a value of -1

Sampled at 20000Hz

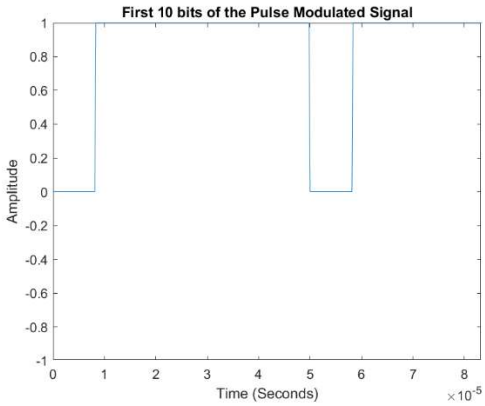
Unipolar NRZ



Fs=20000Hz
L=4



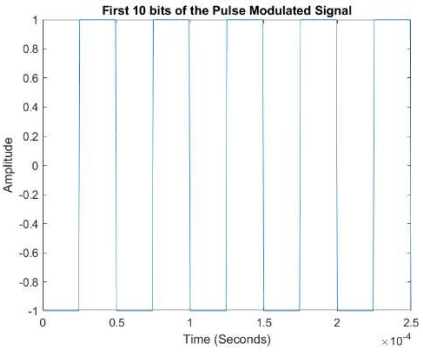
Fs=20000Hz
L=8



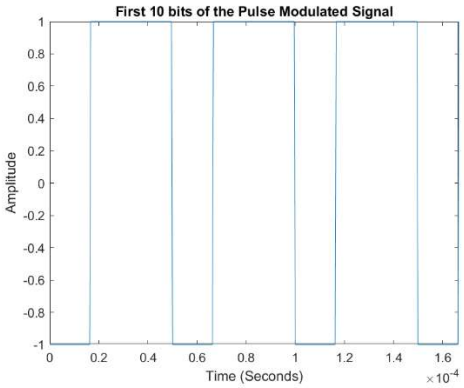
Fs=20000Hz
L=64

The above graphs show the same results for a different sampling frequency. We notice however that the binary representation is equivalent to the one for 5000Hz sampling frequency. This is not an indication that the sampling frequency has no effect on the PCM signal and is just a coincidence.

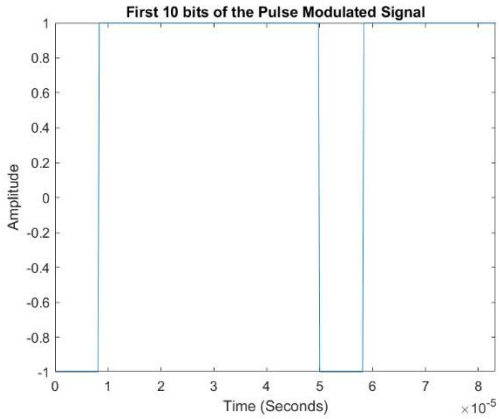
Polar NRZ



Fs=20000Hz
L=4



Fs=20000Hz
L=8

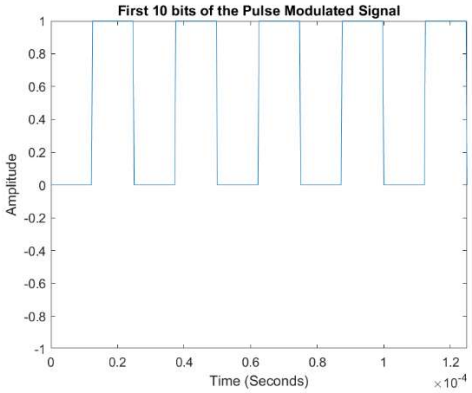


Fs=20000Hz
L=64

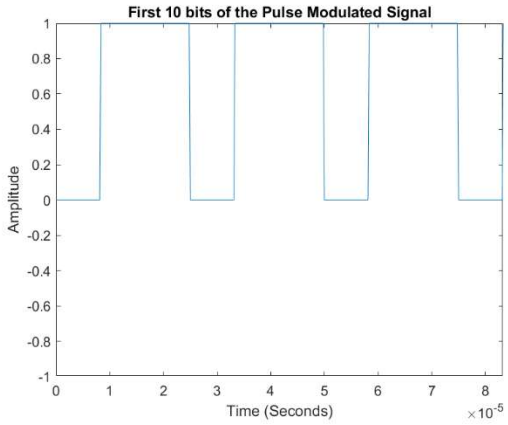
Similar to the usual contrast between the unipolar and polar we notice that these results alternate between 1 and -1 instead of 1 and 0.

Sampled at 40000Hz

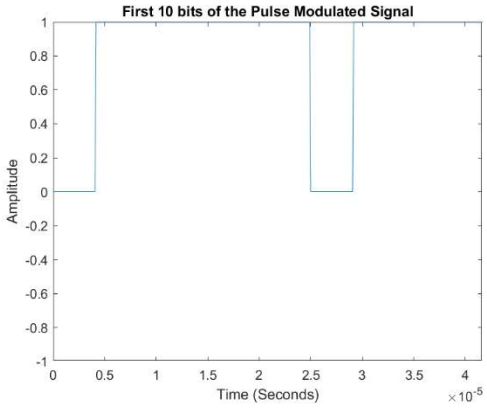
Unipolar NRZ



Fs=40000Hz
L=4

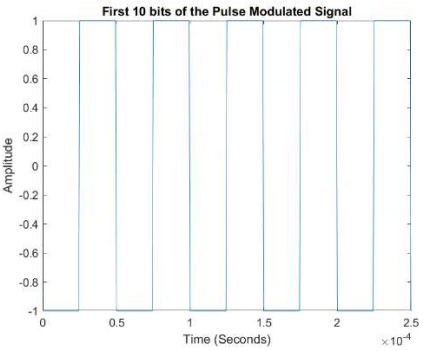


Fs=40000Hz
L=8

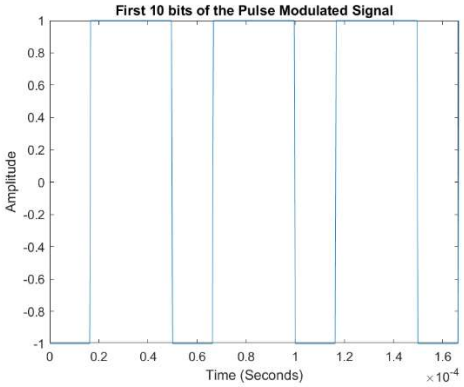


Fs=40000Hz
L=64

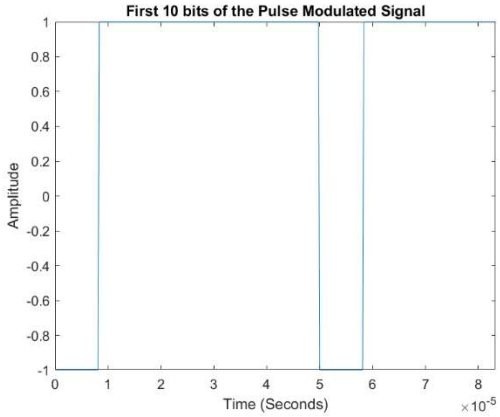
Polar NRZ



Fs=40000Hz
L=4



Fs=40000Hz
L=8



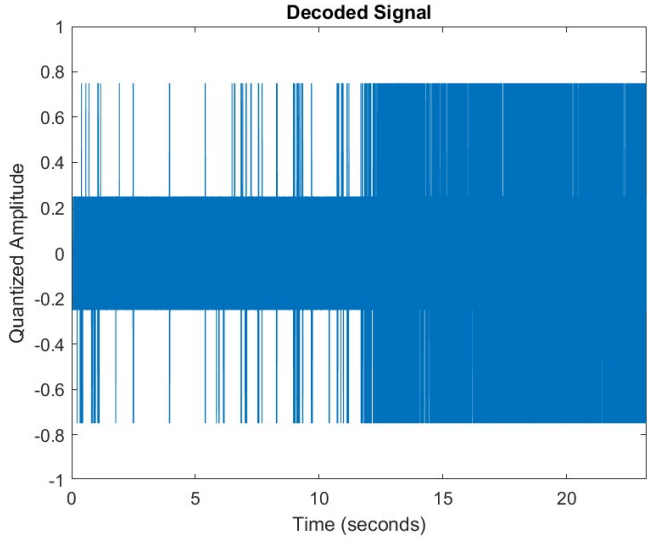
Fs=40000Hz
L=64

Decoder

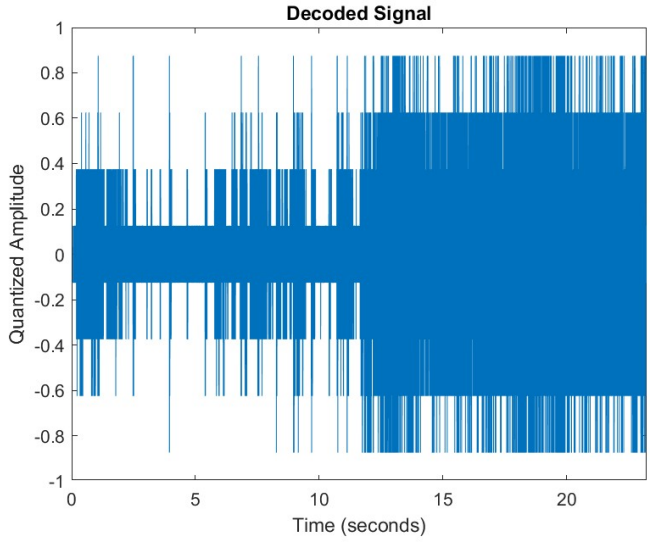
The signal will be passed through a decoder that will translate the pulses into a stream of 0s and 1s then translate these 0s and 1s to their decimal representations and return the original quantized samples.

Sampled at 5000Hz

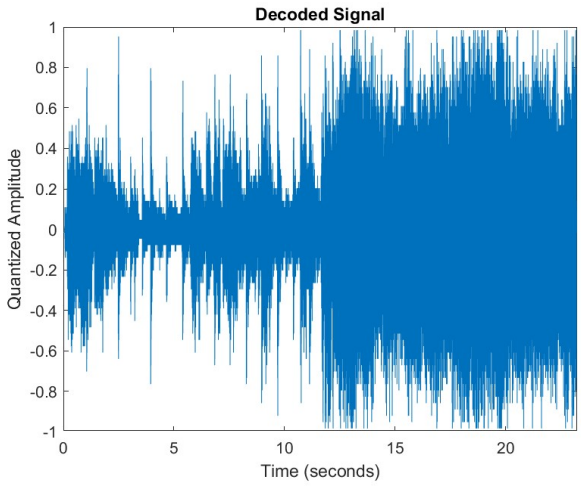
Unipolar NRZ



Fs=5000Hz
L=4

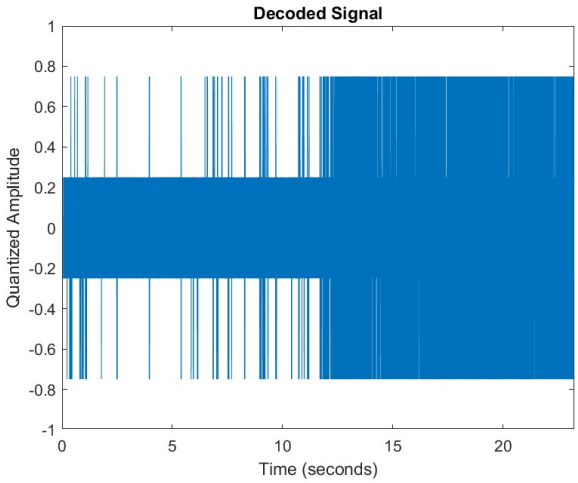


Fs=5000Hz
L=8

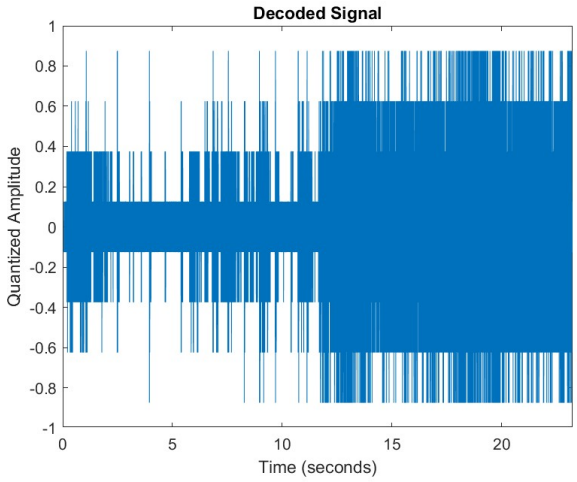


Fs=5000Hz
L=64

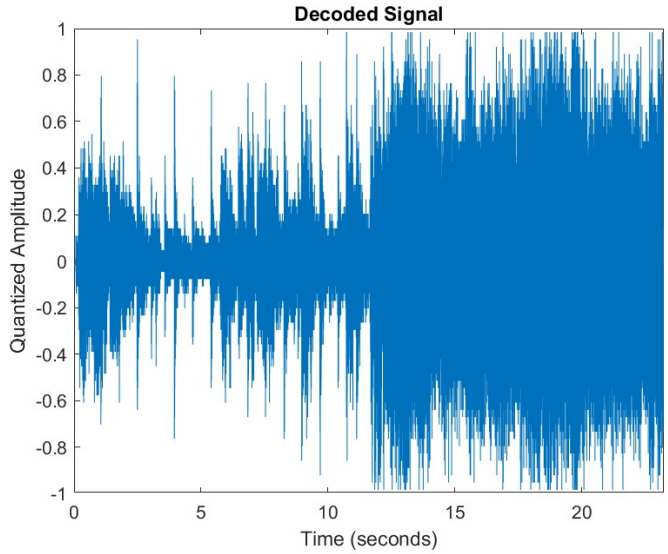
Polar NRZ



Fs=5000Hz
L=4



Fs=5000Hz
L=8

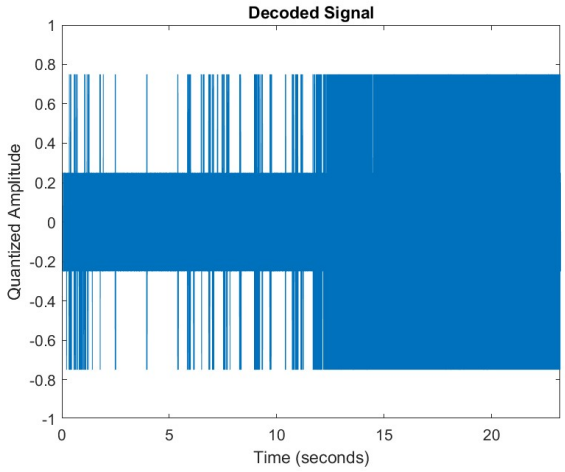


Fs=5000Hz
L=64

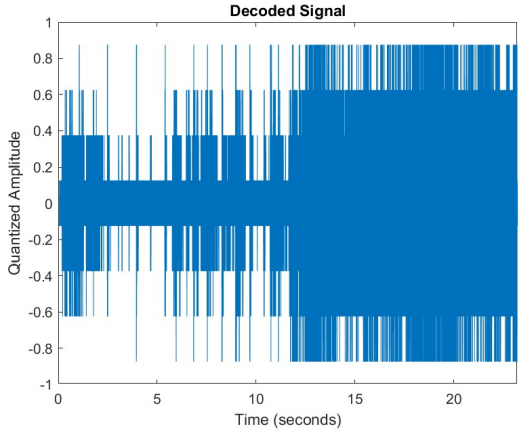
In both cases of polar and unipolar coding the output of the decoder got closer to the original signal as the number of levels increased i.e., the quantization error decreased.

Sampled at 20000Hz

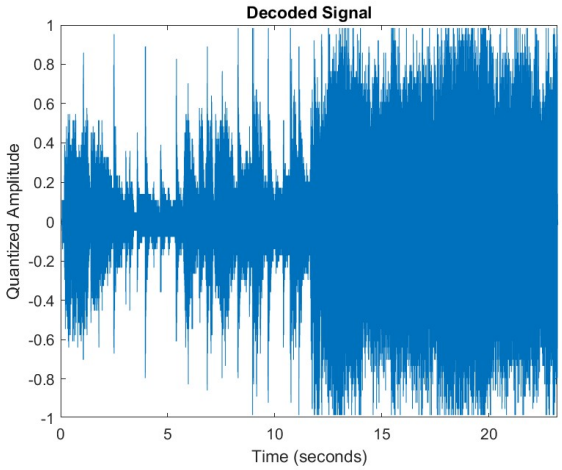
Unipolar NRZ



Fs=20000Hz
L=4

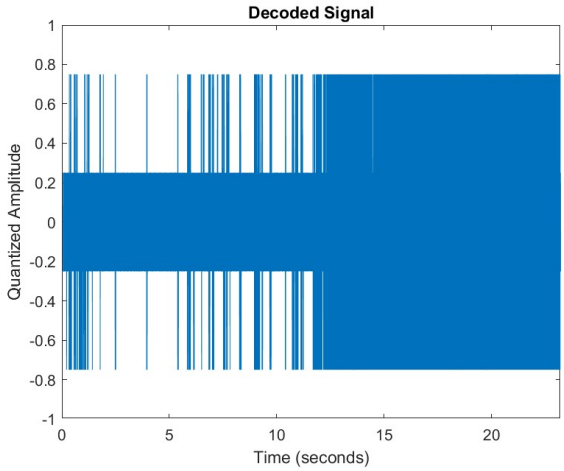


Fs=20000Hz
L=8

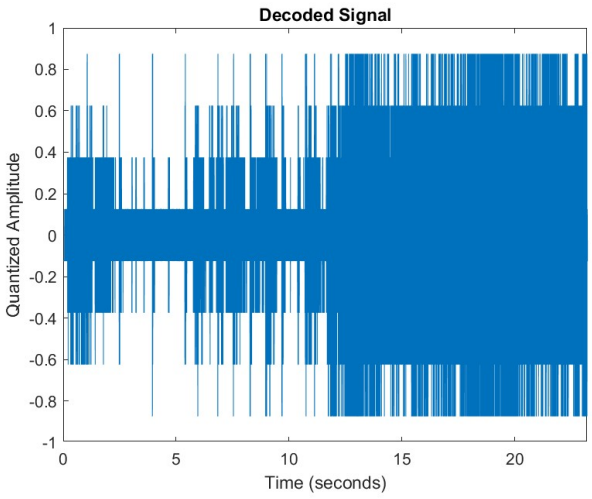


Fs=20000Hz
L=64

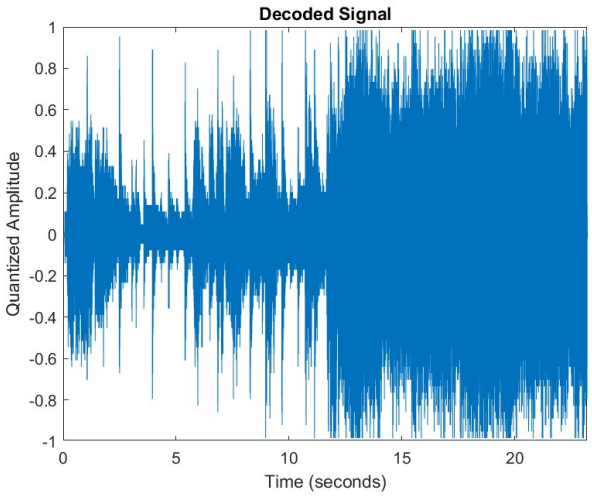
Polar NRZ



Fs=20000Hz
L=4



Fs=20000Hz
L=8

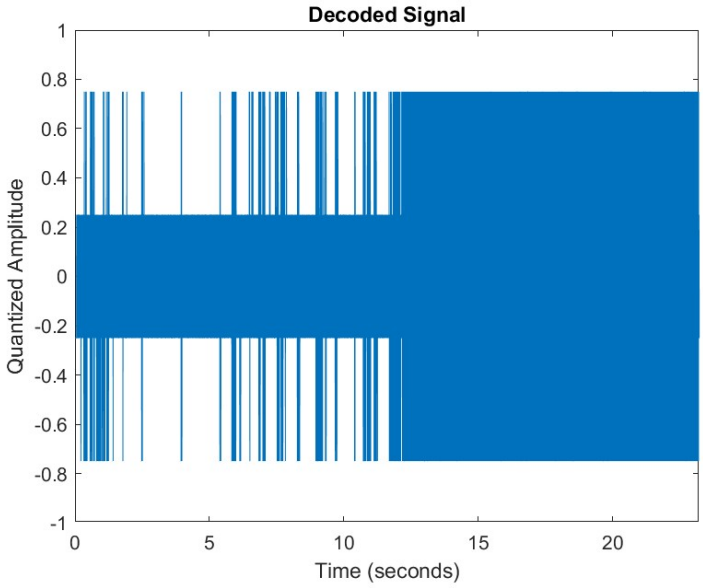


Fs=20000Hz
L=64

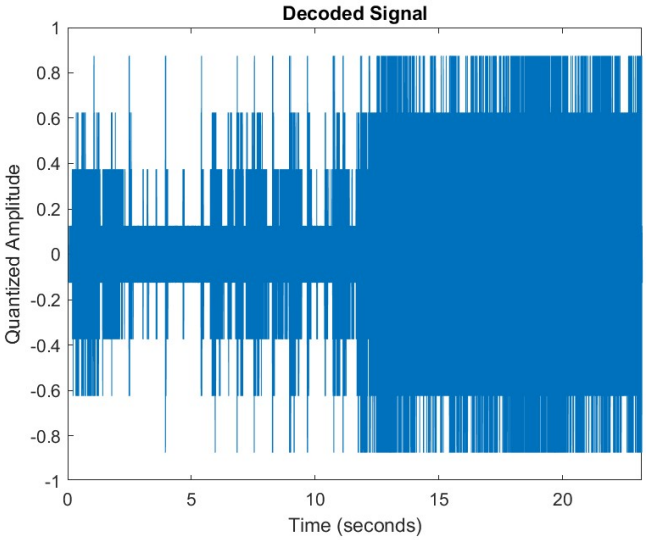
Like the sampling at 5000Hz as the number of levels increased, we got better results however due to the higher sampling frequency we got results closer to the original results.

Sampled at 40000Hz

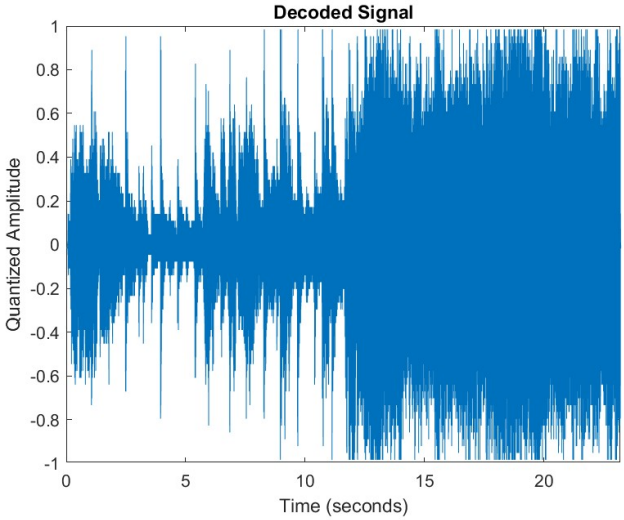
Unipolar NRZ



Fs=40000Hz
L=4

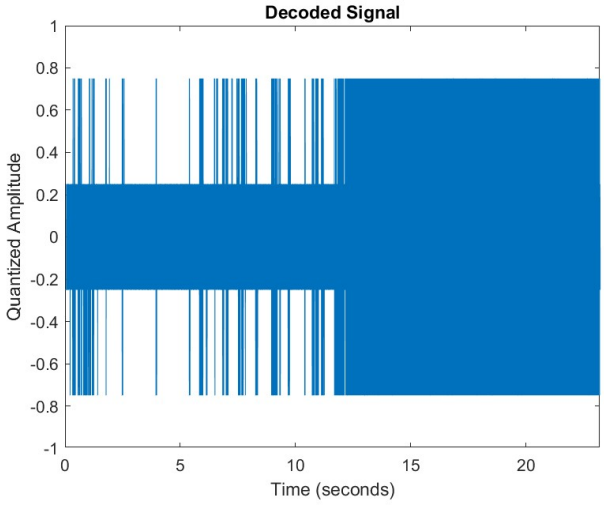


Fs=40000Hz
L=8

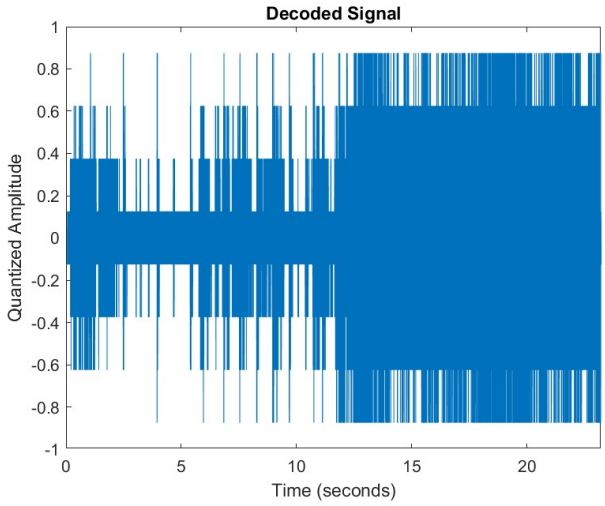


Fs=40000Hz
L=64

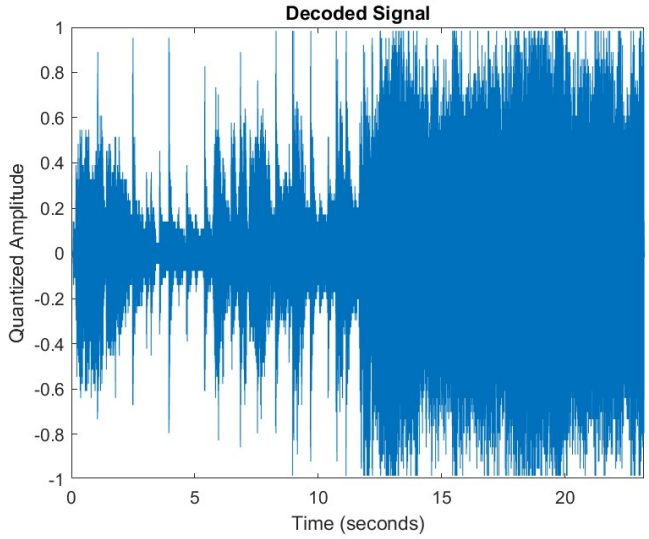
Polar NRZ



Fs=40000Hz
L=4



Fs=40000Hz
L=8



Fs=40000Hz
L=64

Discussion of Test 1

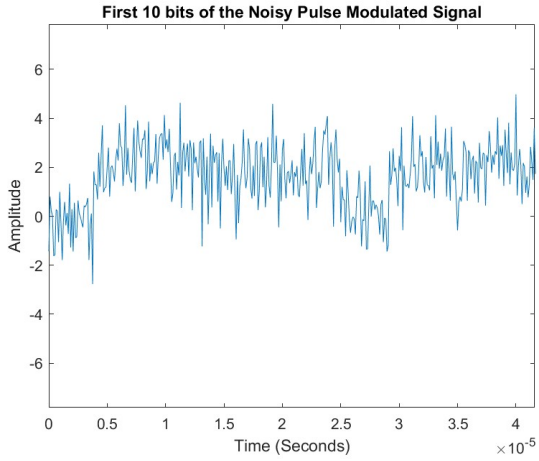
Above we showed a variety of graphs for different values of different parameters at different stages of the PCM process. However, these graphs though provide a bit of insight into the effect of the number of levels they don't really show the effect of the sampling frequency that well. The effect of the sampling frequency can be shown by listening to the audio files where we notice distortion in the noise as the number of levels decreases or as the sampling frequency decreases. The distortion shows up in the form of missing pieces of the sound or tiny ripples in the noise quality.

Test 2

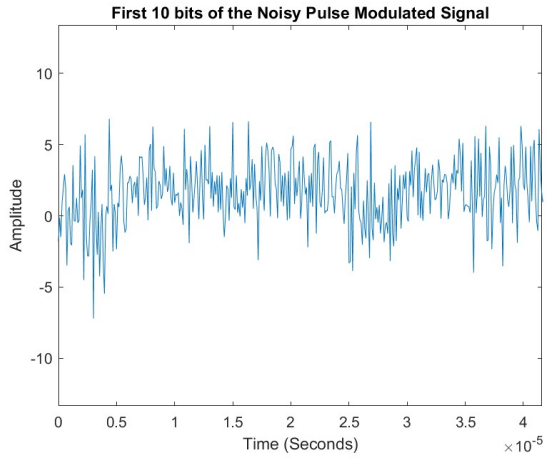
Next, we will repeat the same experiment, but we will add AGWN to the signal. We will vary the variance of the noise added as well as the code line used to view different possible variations.

Noisy Signals

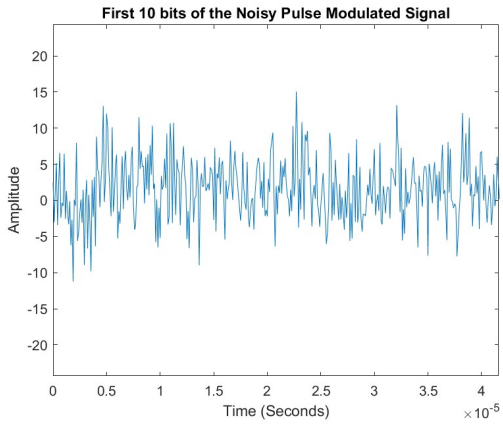
Unipolar NRZ



Variance =1

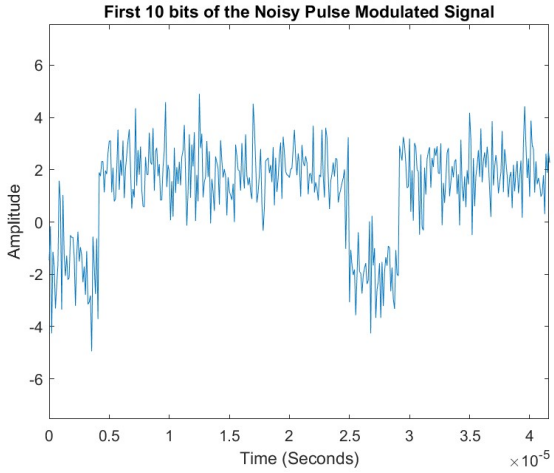


Variance =4

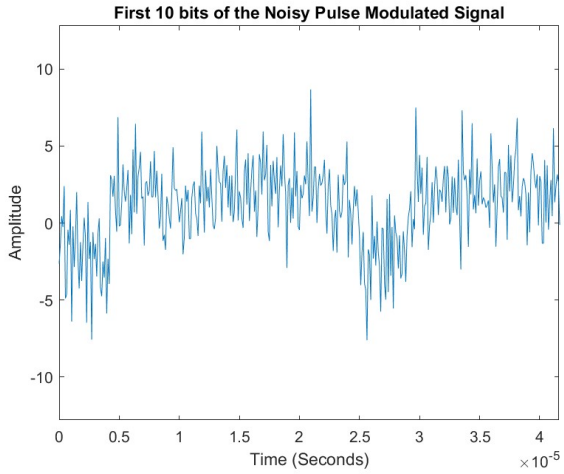


Variance =16

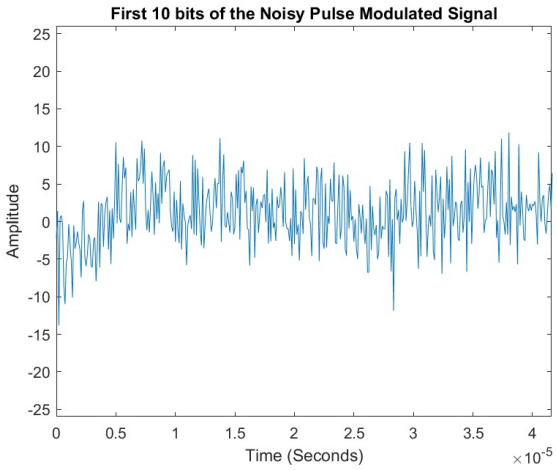
Polar NRZ



Variance =1



Variance=4



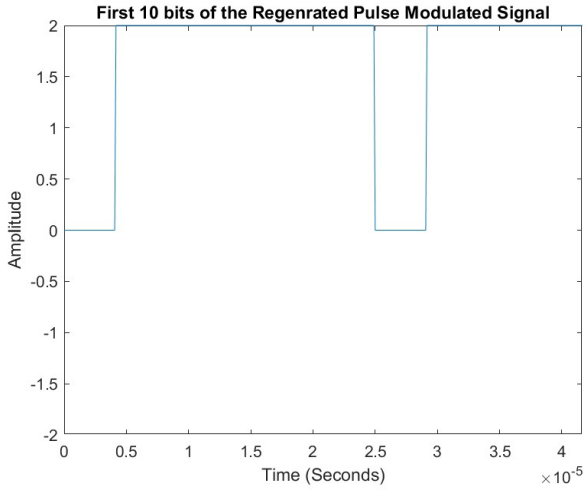
Variance=16

We notice that as the variance increases whether for polar or unipolar the distortion in the pulses increases.

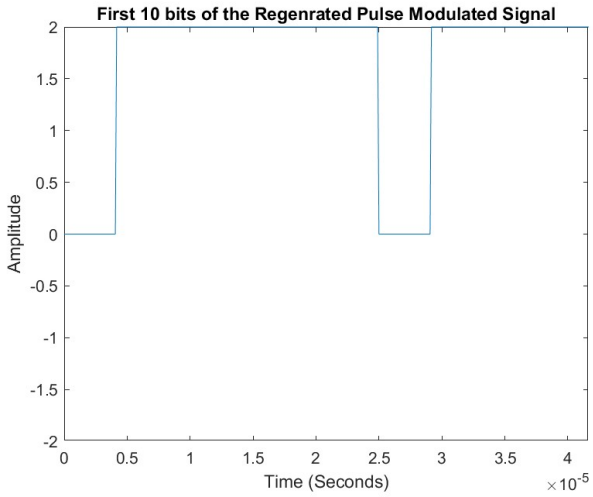
Regeneration

The pulses are sent to the regenerative repeater before the decoder to reproduce the pulses

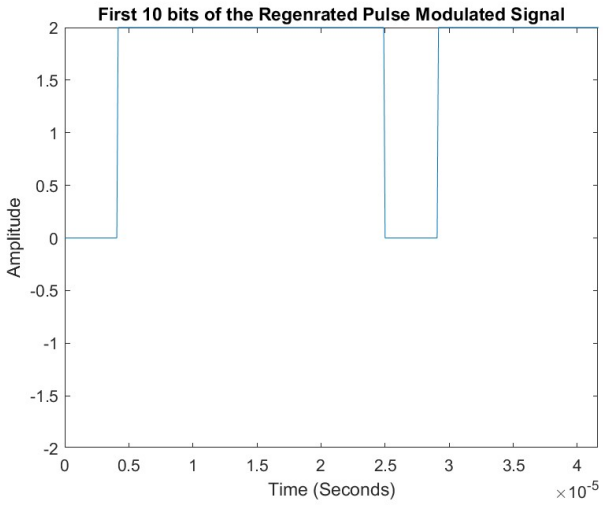
Unipolar NRZ



Variance =1

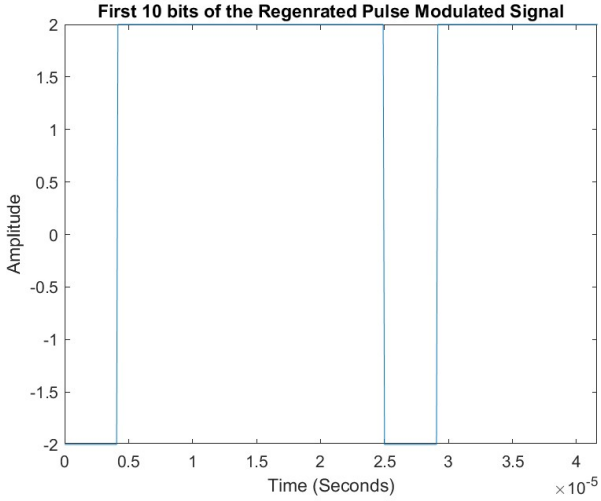


Variance =4

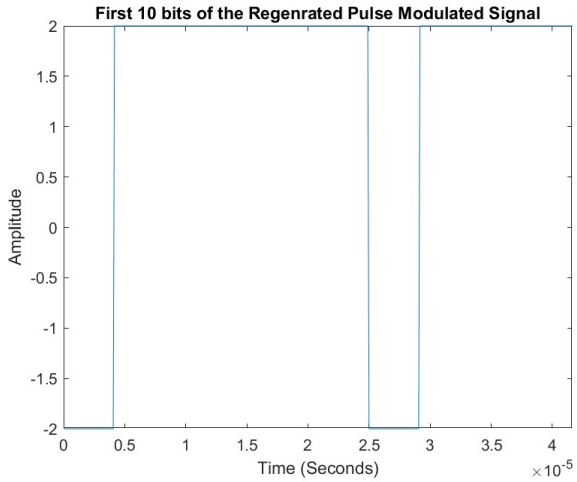


Variance =16

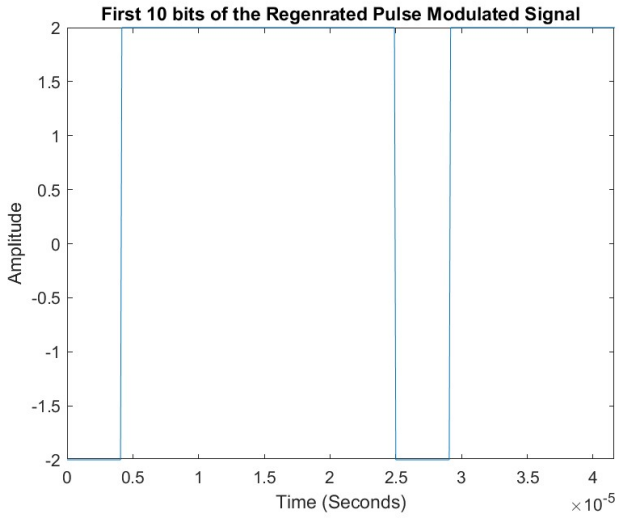
Polar NRZ



Variance=1



Variance=4



Variance=16

Discussion of test 2

The above graphs show that as the variance of the noise increases the more distorted the signal becomes. Despite that the original values are restored the regenerated decoded signal still has some extra distortion to it as the variance increases. Though the regenerator should ideally restore the bits to their correct form depending on the random variable distribution, there is still a chance that the bits are incorrectly interpreted by their repeater. This implies that even though the regenerative repeater gets rid of a high percentage of the distortion there is still a slight probability that some bits are misinterpreted, and that probability is proportional to the variance of the noise.