University of Science and Technology
Zewail City

CIE 337 Communication Theory and Systems

Project 2 Report

Ibrahim Hanafy 202200518
Khaled Mahmoud 202200282
Abdullah Ahmed 202200206
Ahmed Shaban 202200217

Instructor

Dr. Samy Soliman

Contents

1 Aim And Scope

Our Aim Is to Implement the following system using MATLAB:

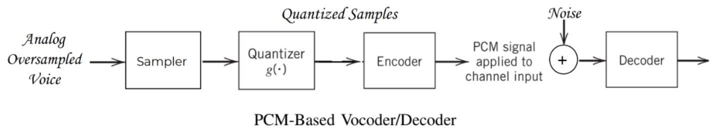

Figure 1: BLock Diagram Of The System

The Concept of transmitting Digital signals instead of Analog Ones and preciesly doing this by PCM Systems Have Major Pros Which Include:

- Some Immunity To Noise:In Comparing Digital Signal With Finite Amplitude Ranges to Analog SIgnals With Infinte Ones, Repeaters Can be very good in Comparing the signal to its Original Form and then reconstruct it again accordingly which usually isn't possible in Analog Systems
- Detection and Correction: PCM allows the use of sophisticated error-detection and error-correction algorithms (e.g., parity bits, checksums), which can significantly improve the reliability of the communication system, especially in noisy environments.
- Ease Of Processing

The PCM System is Mainly Consisting of the following:

1. Sampler: which can be said is the process of Discretization of the signal to take discrete sampling according to nyquist frequency theorem:
$$2f_{max} \leq f_s \tag{1}$$
2. Quantizer: It Is the process of digitization of the signal Done Through Choosing Number Of levels To limit the amplitudes to a finite number.
3. Encoder: each quantized sample is converted to a stream of line Code with different signaling techniques.
4. Regenerative Repeater it consists of the followig:

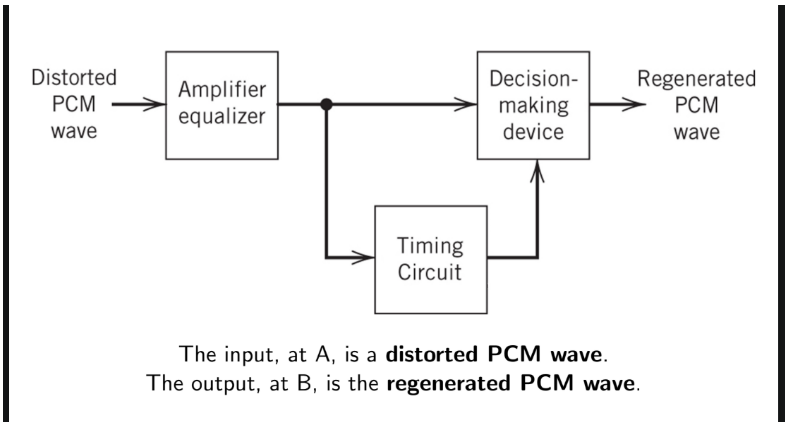

Figure 2: Repeater Block Diagram

Its Main Funtion is to compare the incoming signal to its knowledge of the amplitude sets and regenerate it again this what makes PCM Systems Special Because we can say that this elimnates Noise.

5. Decoder: The Decoder Does Dequantization,Desampling and maps the recieved line code to its quantized sample and also may offer to apply a reconstruction Filter.

2 Test Case One

2.1 Importing the Audio

```
% --- Convert to WAV if needed ---
input_file = 'Amr Diab - Lama Abeltak_ عمرو دياب - لما قابلتك(M4A_128K).m4a';
[~, name, ~] = fileparts(input_file);
converted_file = [name, '.wav'];

if ~isfile(converted_file)
    [y, fs_conv] = audioread(input_file);
    audiowrite(converted_file, y, fs_conv);
    fprintf('Converted "%s" to "%s"\n', input_file, converted_file);
end

% --- Load and preprocess audio ---
[Signal, fs_original] = audioread(converted_file);
Signal = mean(Signal, 2);  % Convert stereo to mono
start_sample = fs_original * 1+ 1;
end_sample = min(length(Signal), fs_original * 20);
Signal = Signal(start_sample:end_sample);
```

Figure 3: Uploading the file converting it if needed, and Trim The First 20 second for faster responses

2.2 Tests and Outputs

2.2.1 f_s =5KHz ,L=4,Polar Signaling :

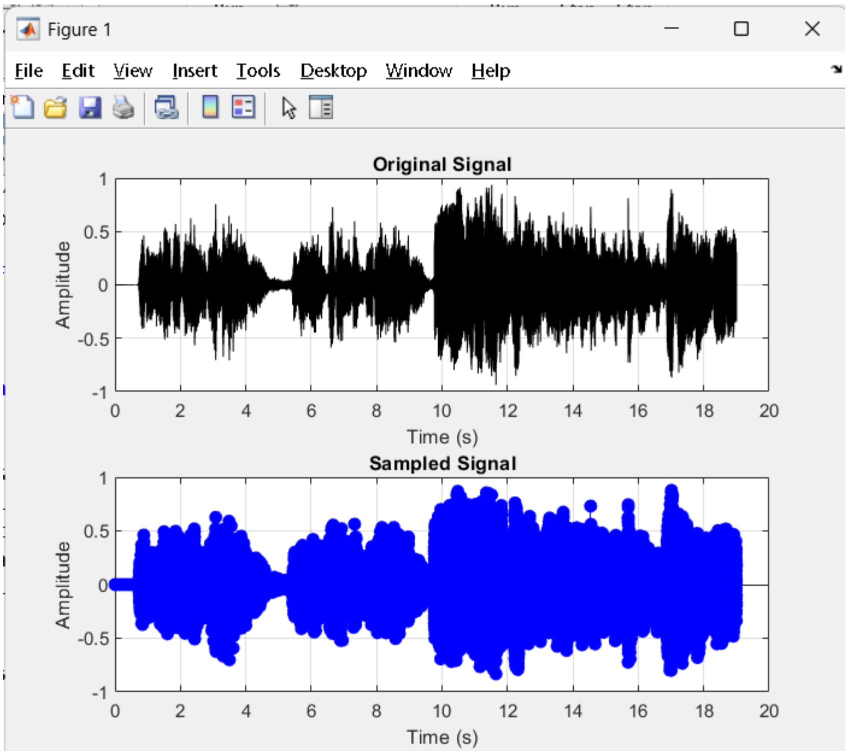

Figure 4: We can See the Figure Showing the samples took versus the Original Signal Which will not change for all the following because changing L and the Signaling type does not affect the sampling as they are independent.

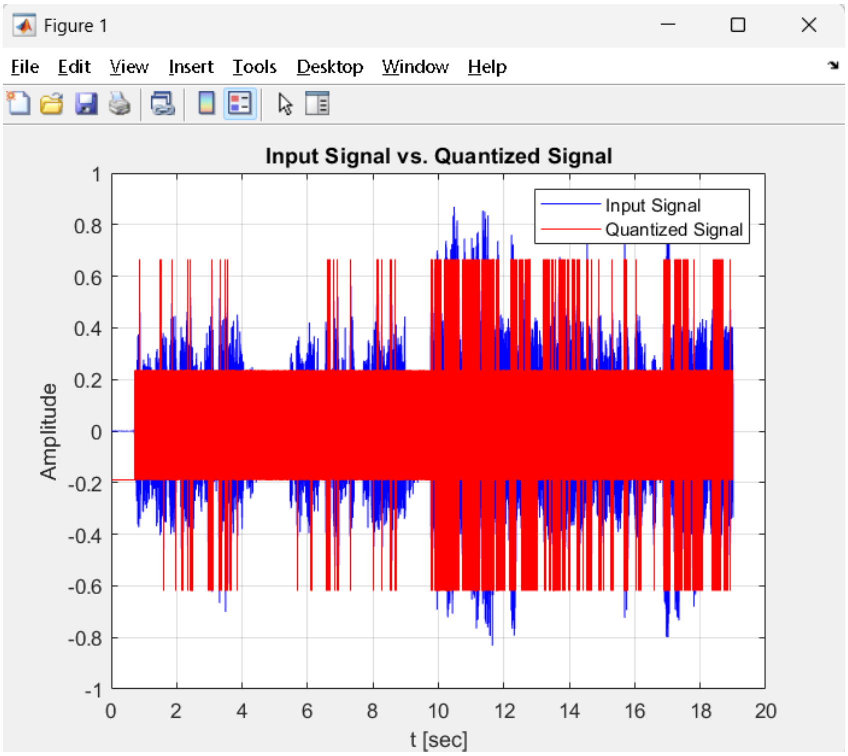

Figure 5: This is the output of the Quantizer, we clearly see that due to low Number of levels the approximations are higher.

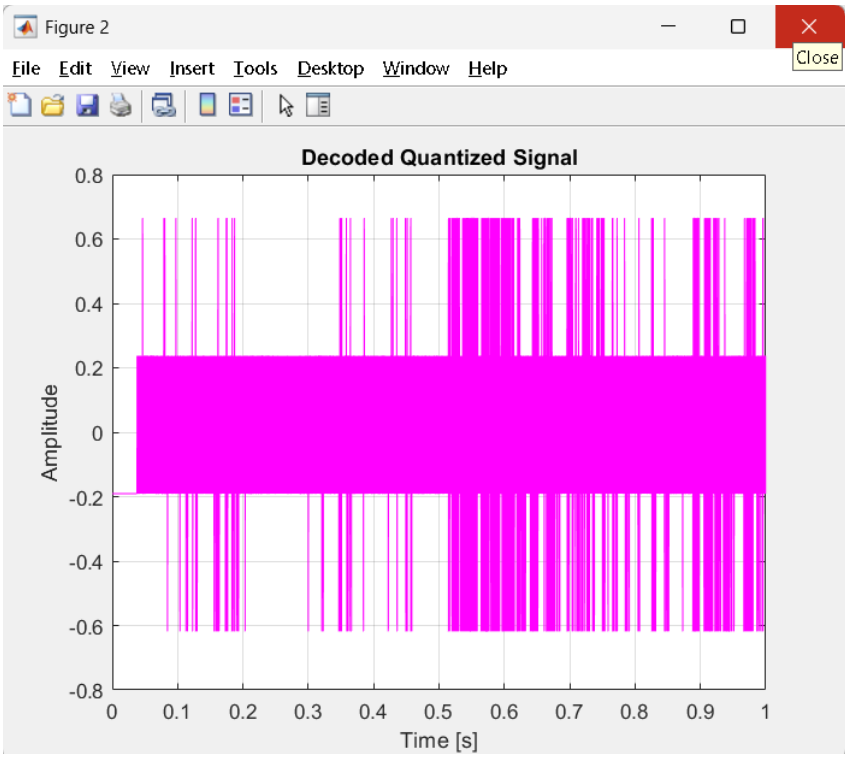

Figure 6: This is the Output of the Quantizer ad it will of course be missing main Components Due to Low Number OF Levels

2.2.2 f_s=5KHz,L=8, Polar Signaling

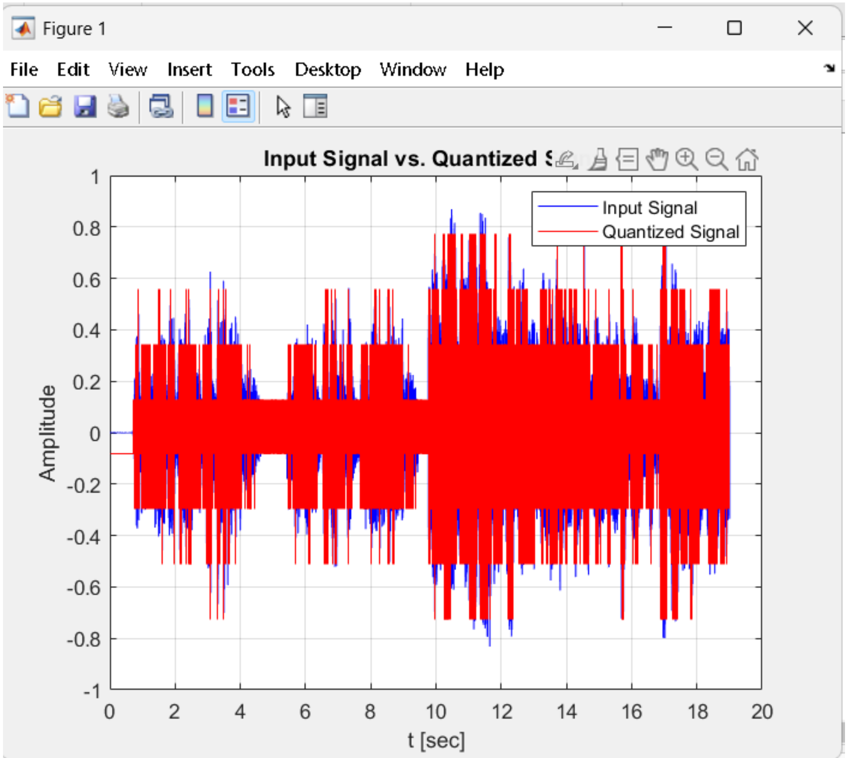

Figure 7: We See That By Just Adding 4 More Levels, More components of the original Audio were taken into consideration

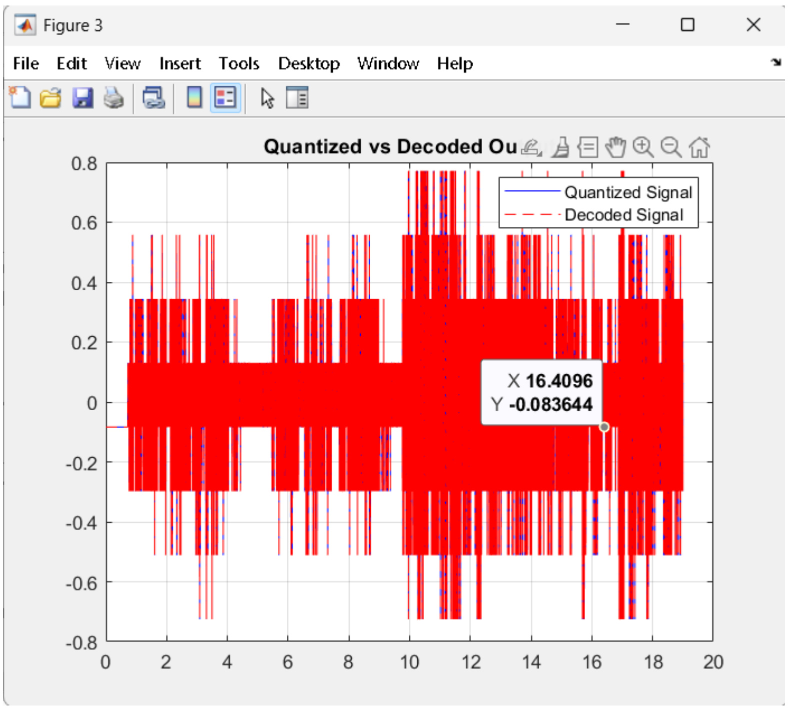

Figure 8: Adding levels Will also Affect on the decoding as more of the original signal will be retrieved.

2.2.3 f_s=5KHz,L=64, Polar Signaling

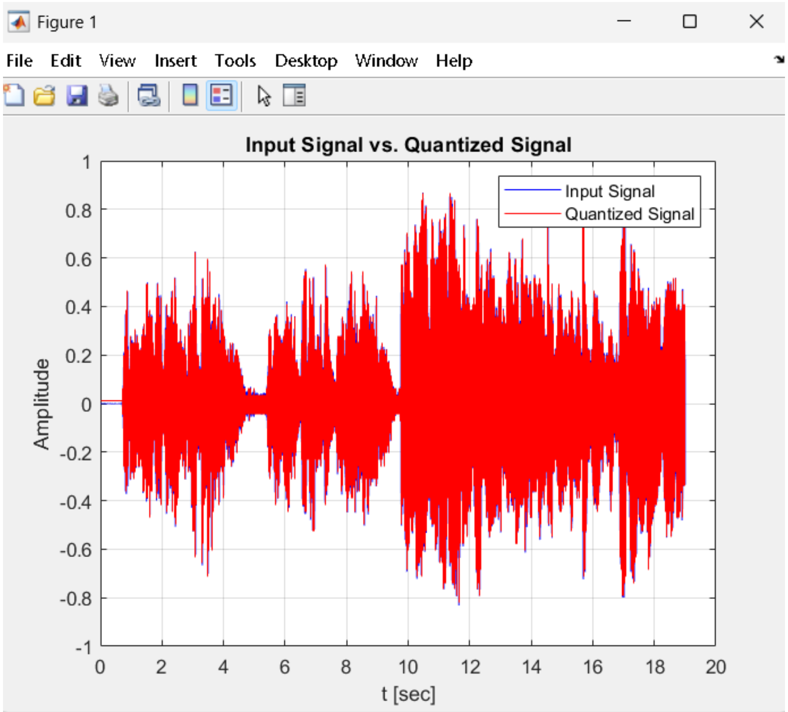

Figure 9: By Adding this many Levels we Can say that most of the amplitudes were not approximated which will be very vulnerable while retreiving

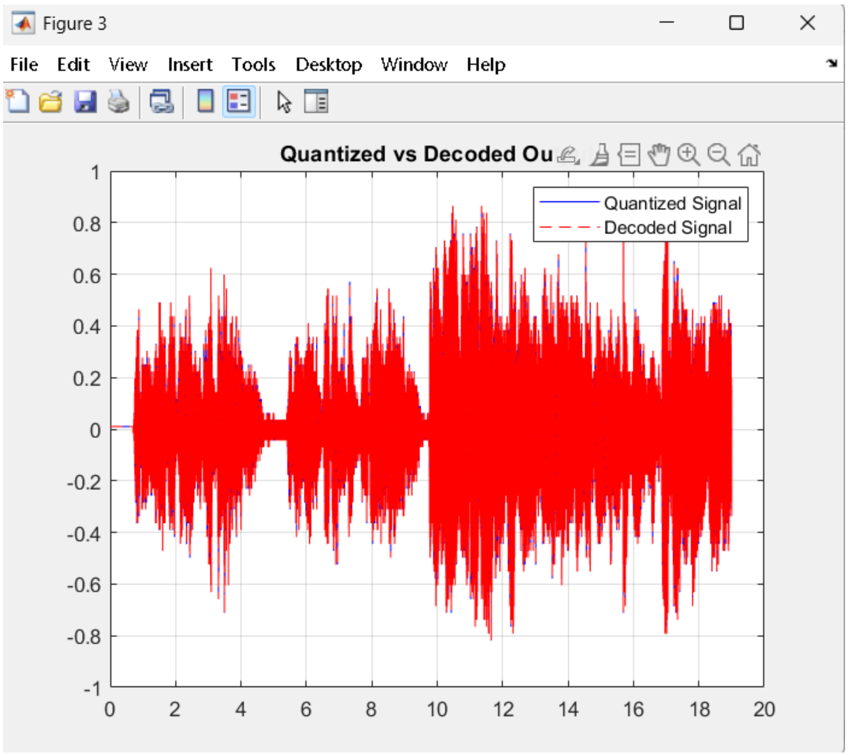

Figure 10: Adding levels Will also Affect on the decoding as more of the original signal will be retrieved.

2.2.4 f_s =5KHz ,L=4,UniPolar Signaling :

Sampling is the SAME..

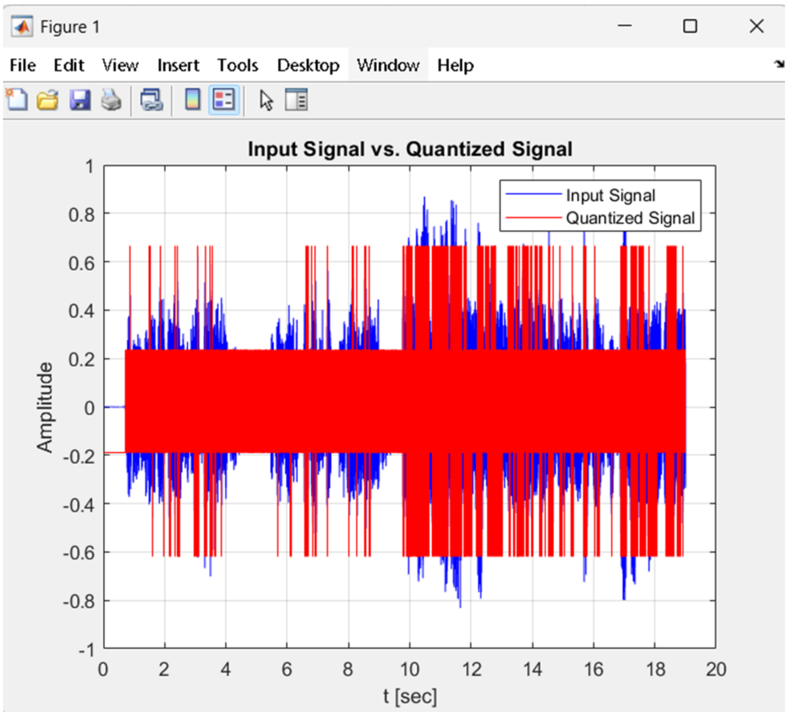

Figure 11: Very Similar Behaviour For Polar Except For The Line Code.

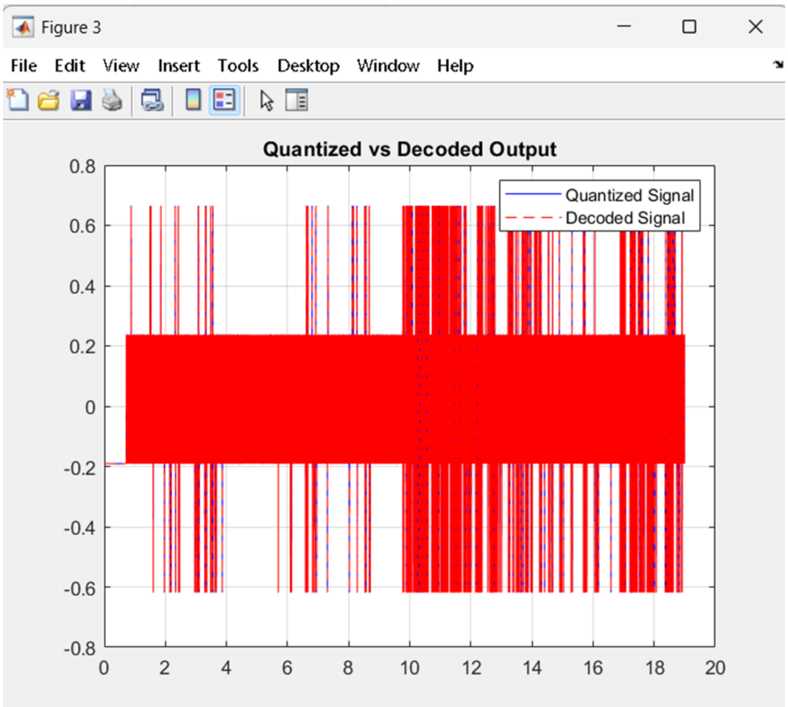

Figure 12: This is the Output of the Quantizer ad it will of course be missing main Components Due to Low Number OF Levels

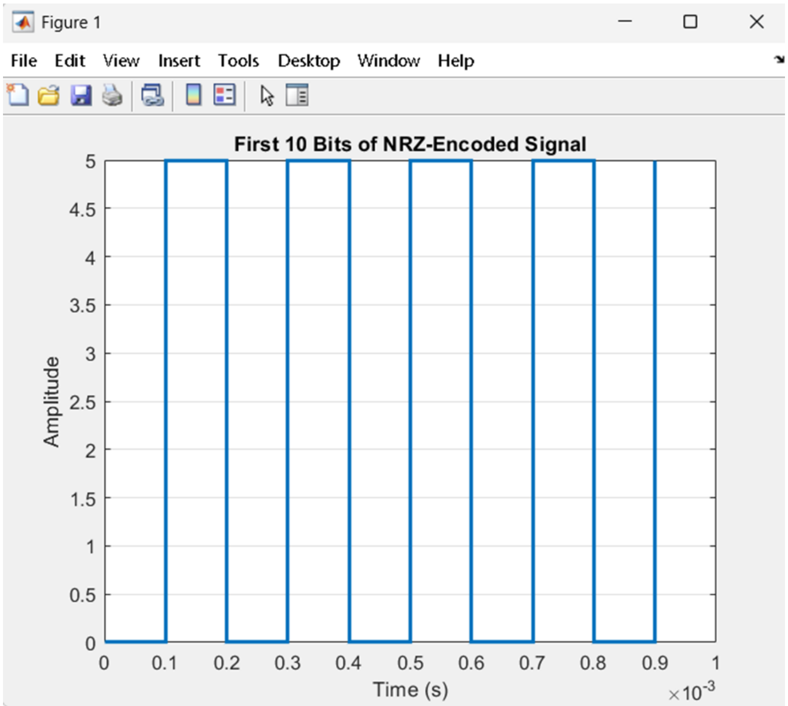

Figure 13: This is the first 10 bits With Amplitudes of 0 and 5 Volts.

2.2.5 f_s=5KHz,L=8,UniPolar Signaling

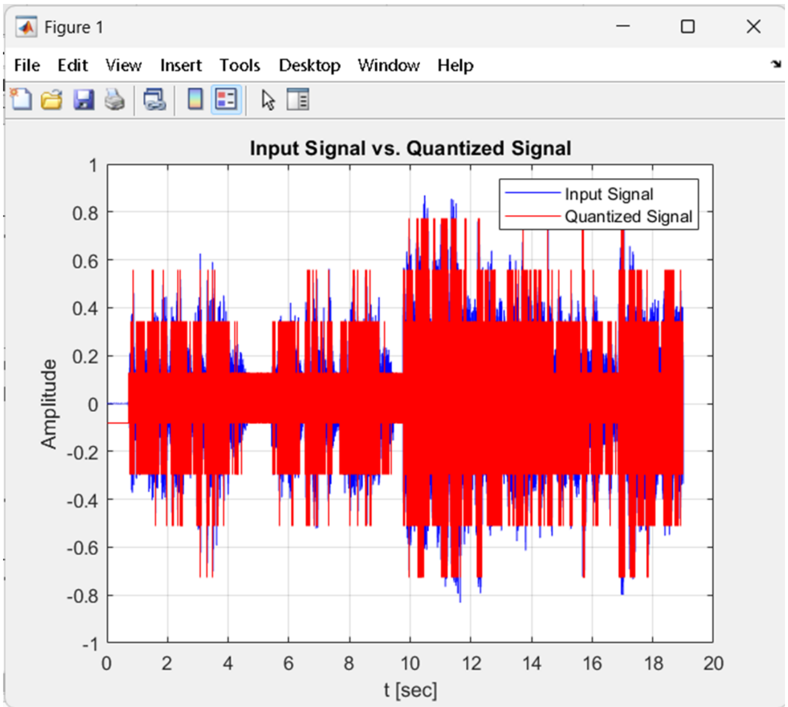

Figure 14: Adding Levels Will Accommodate More Sample's Exact value

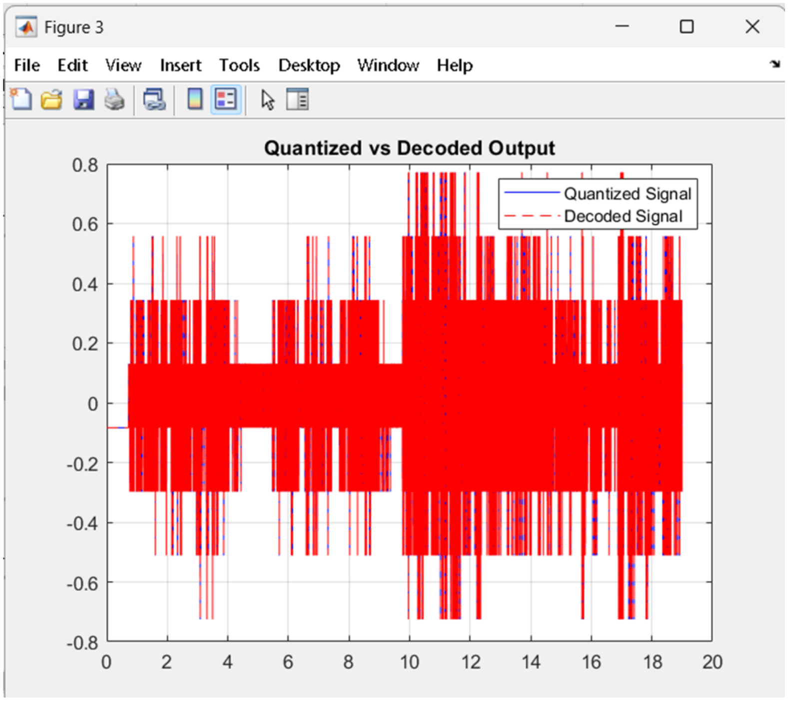

Figure 15: Better Decoded

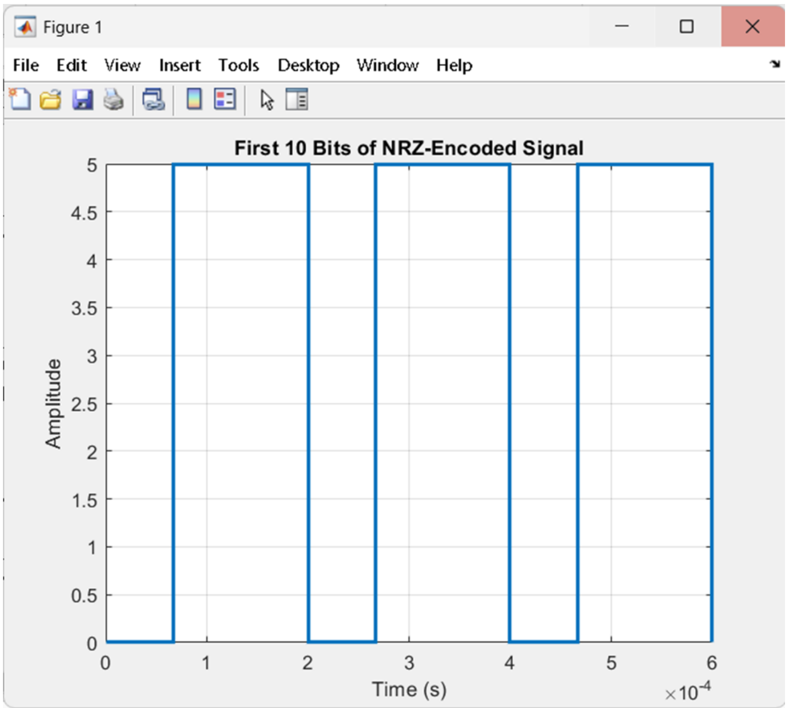

Figure 16: We Clearly See differences between this one and the one for L=4 Due to the increase of levels

2.2.6 f_s=5KHz,L=64,UniPolar Signaling

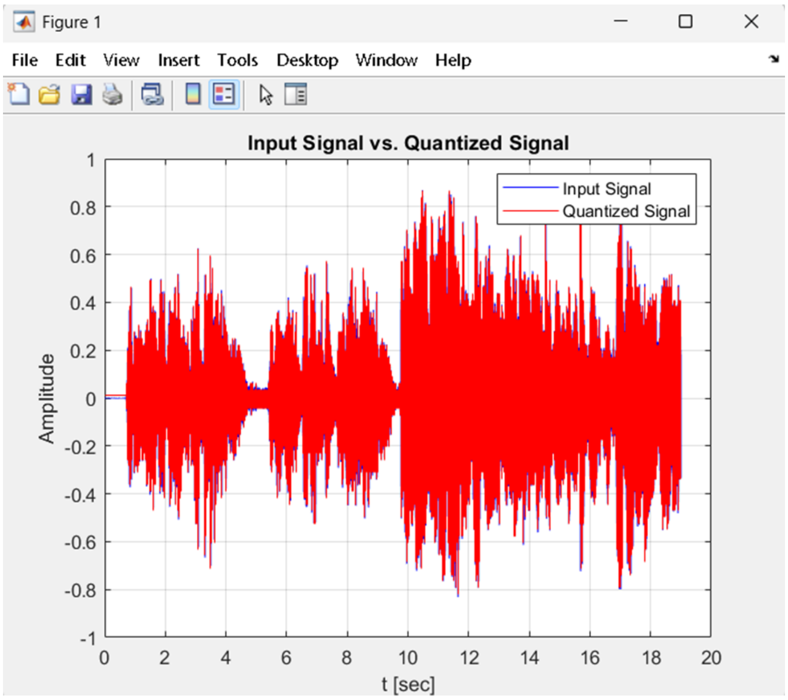

Figure 17: More Levels= Better Decoding

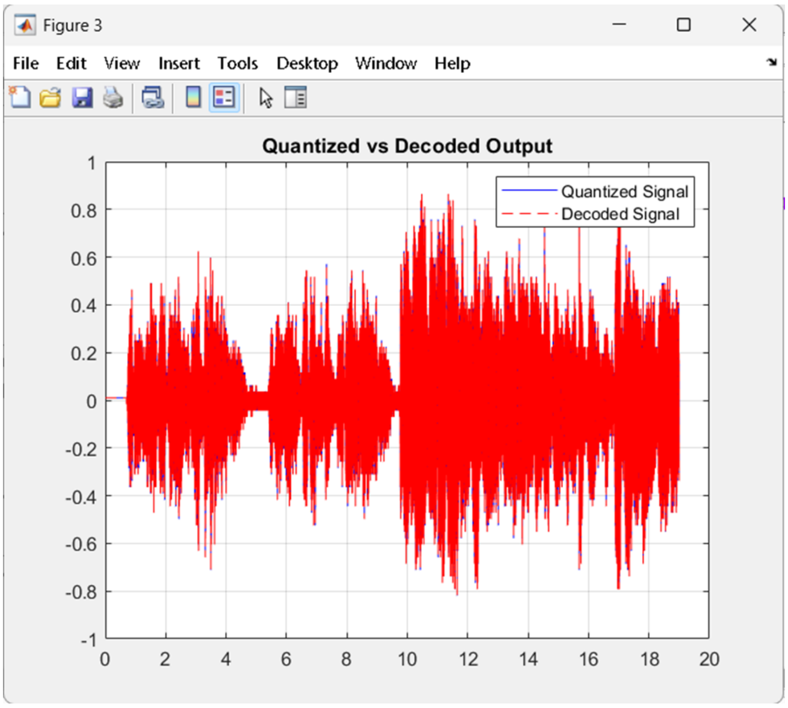

Figure 18: More Levels= Better Decoding.

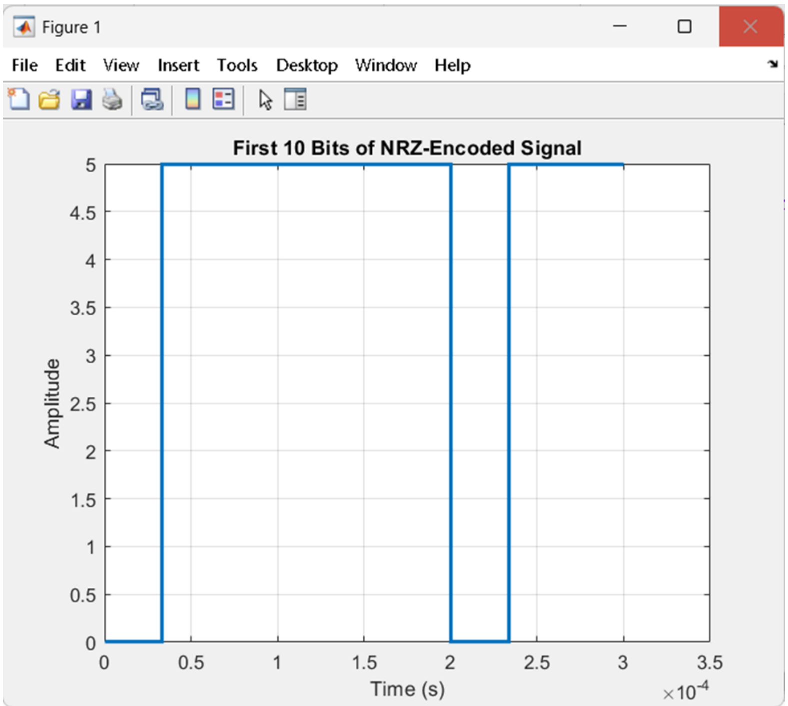

Figure 19: Difference Due to more levels = different values for mapping.

2.2.7 f_s =20KHz ,L=4,Polar Signaling :

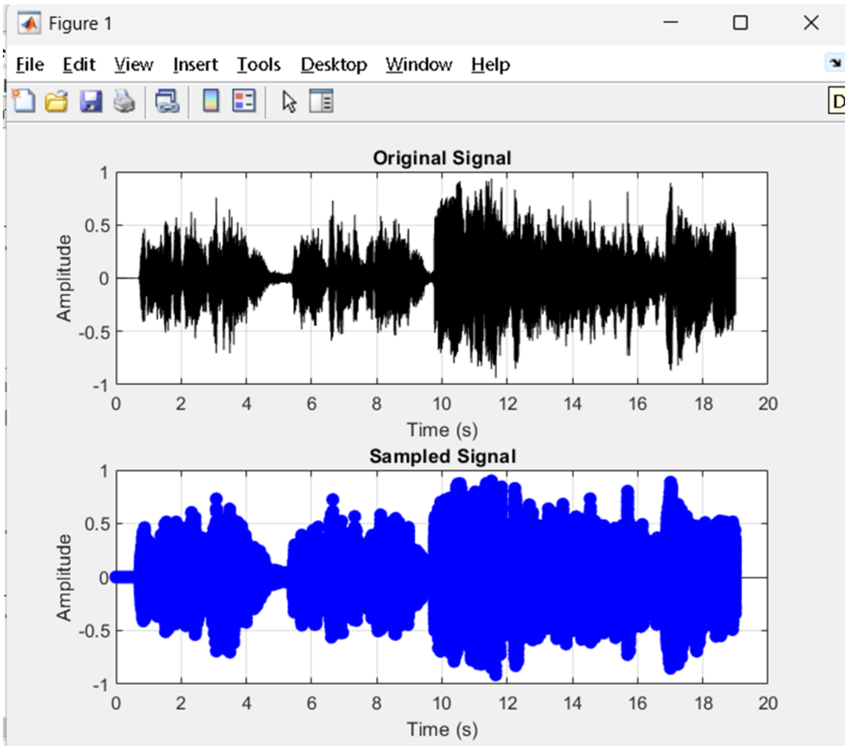

Figure 20: Increasing Sampling Frquency,Leads TO Better Audio Quality,Larger Bitstream.

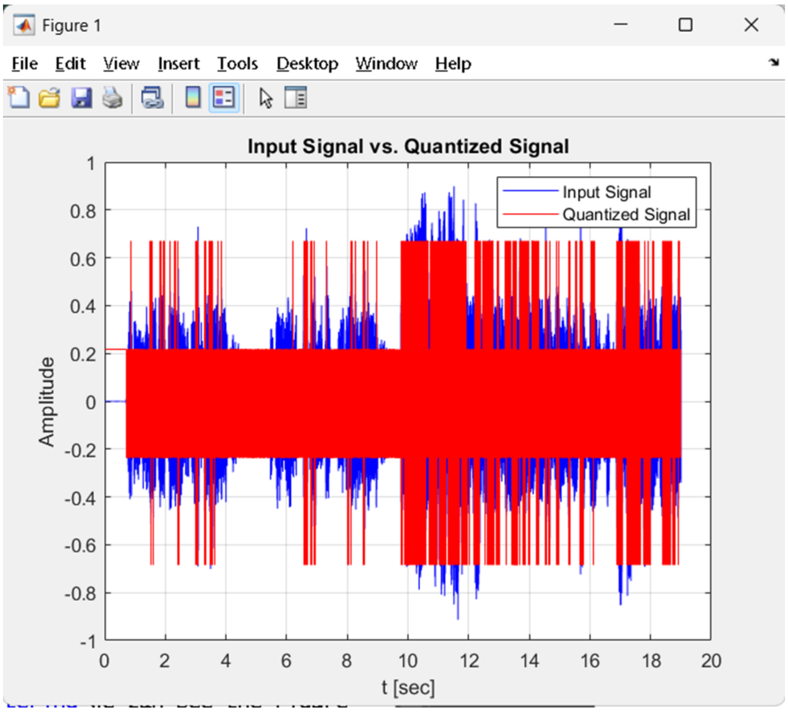

Figure 21: This is the output of the Quantizer, we clearly see that due to low Number of levels the approximations are high.

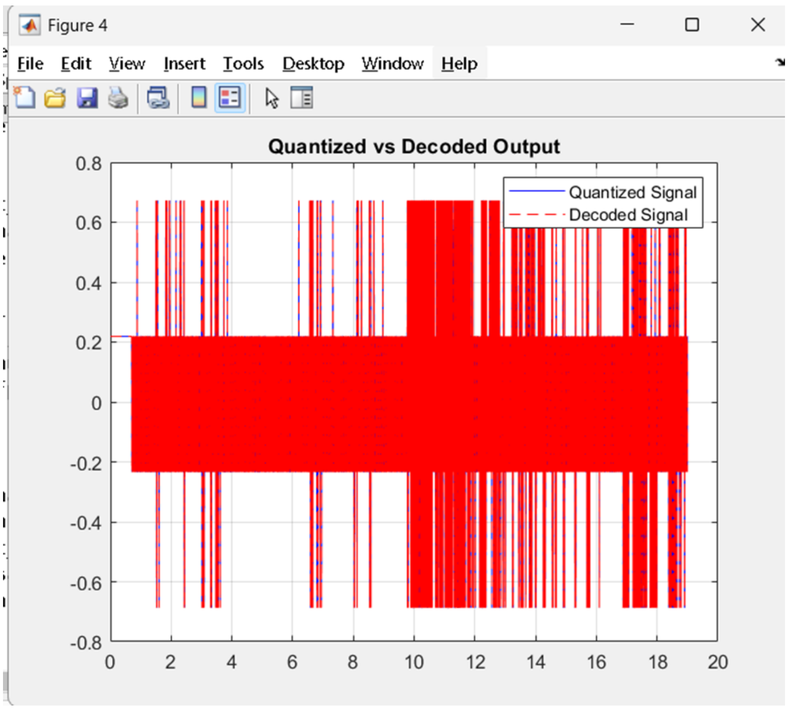

Figure 22: This is the Output of the Quantizer ad it will of course be missing main Components Due to Low Number OF Levels

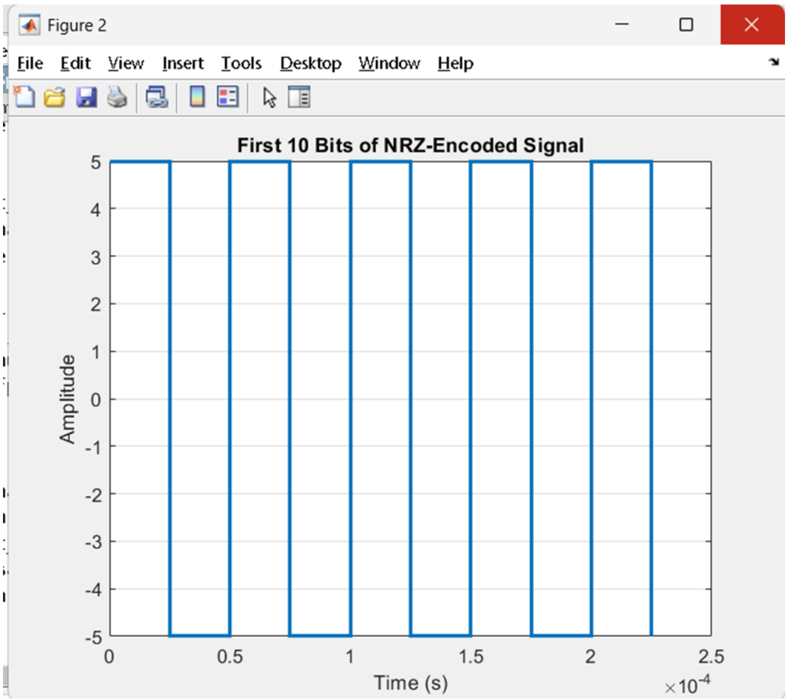

Figure 23: Encoded Signal

2.2.8 f_s=20KHz,L=8, Polar Signaling

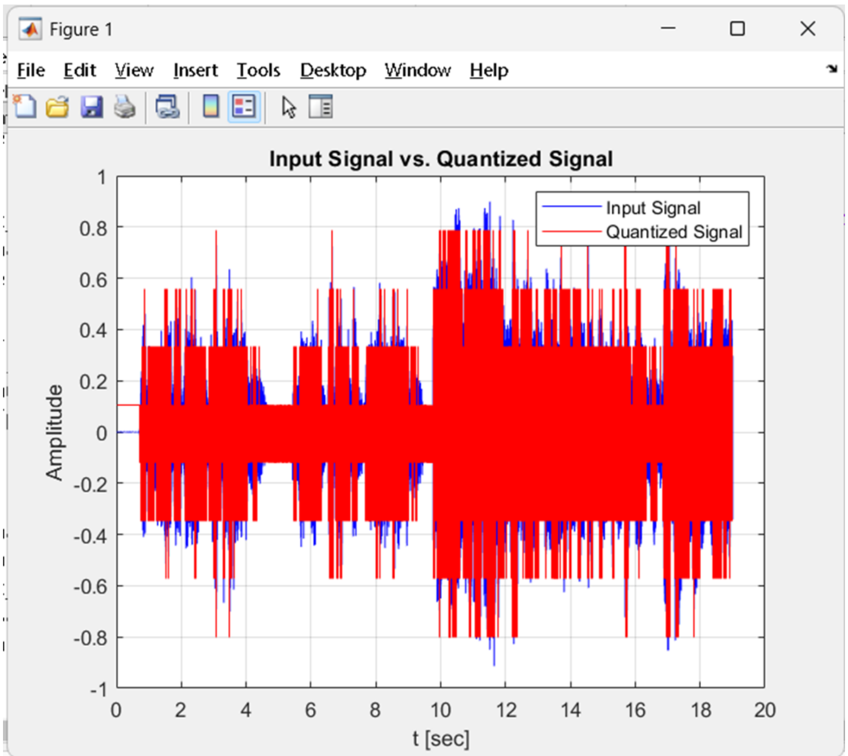

Figure 24: We See That By Just Adding 4 More Levels, More components of the original Audio were taken into consideration, combining this with higher f_S will lead to better reconstruction

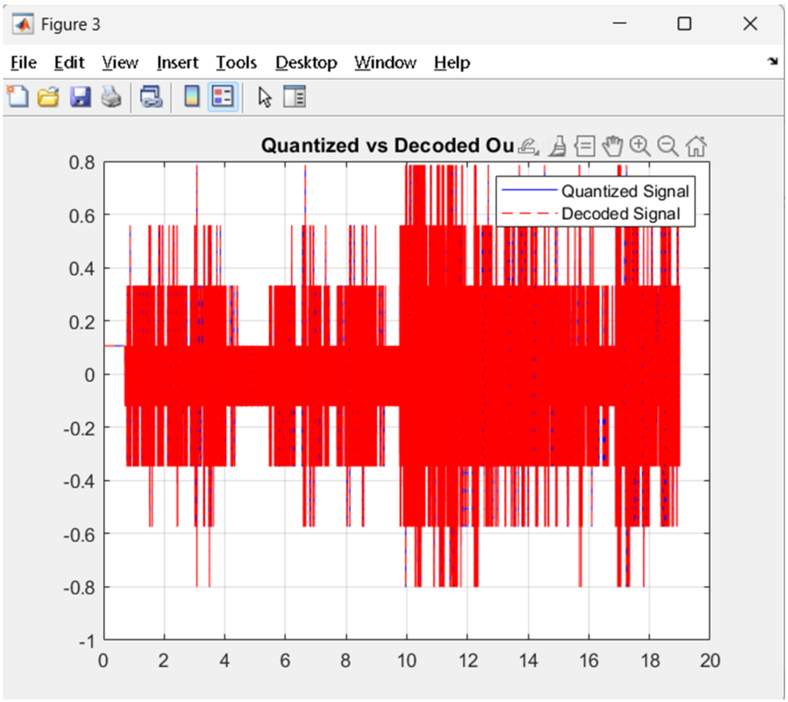

Figure 25: Adding levels Will also Affect on the decoding as more of the original signal will be retrieved.

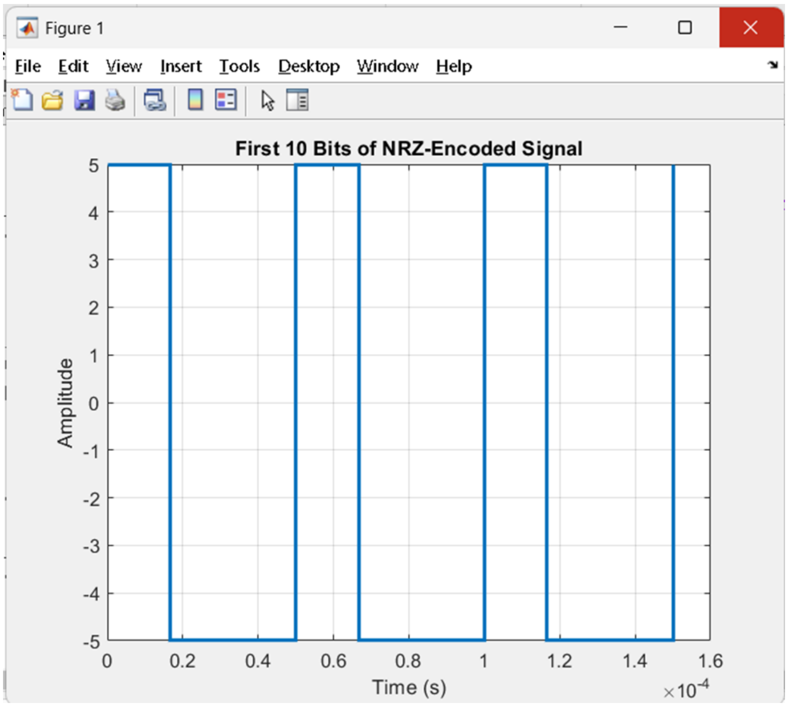

Figure 26: Line Code

2.2.9 f_s=20KHz,L=64, Polar Signaling

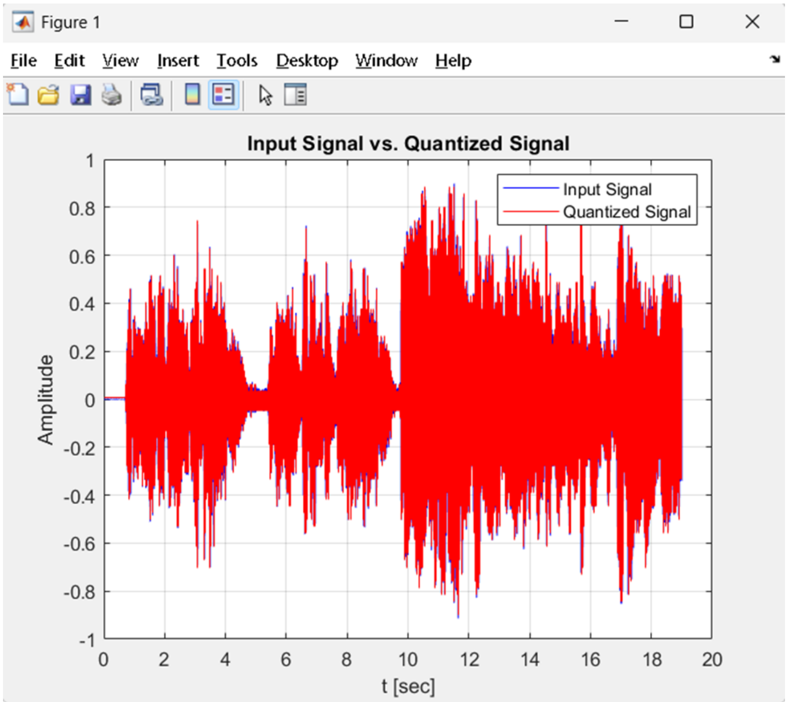

Figure 27: Combining More Levels With This f_S will be the most Clear Output

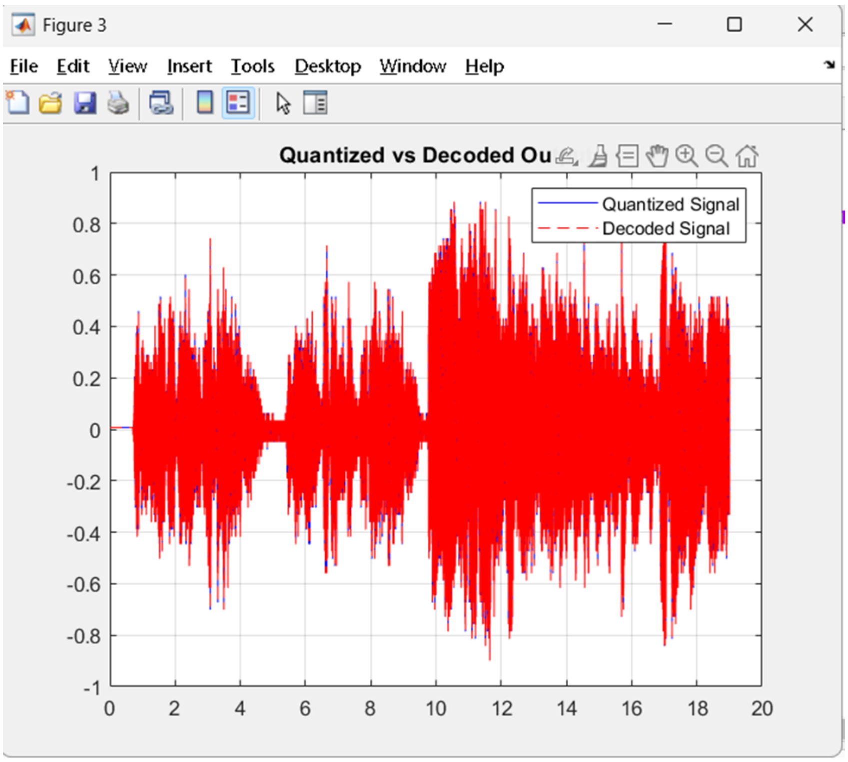

Figure 28: Adding levels Will also Affect on the decoding as more of the original signal will be retrieved.

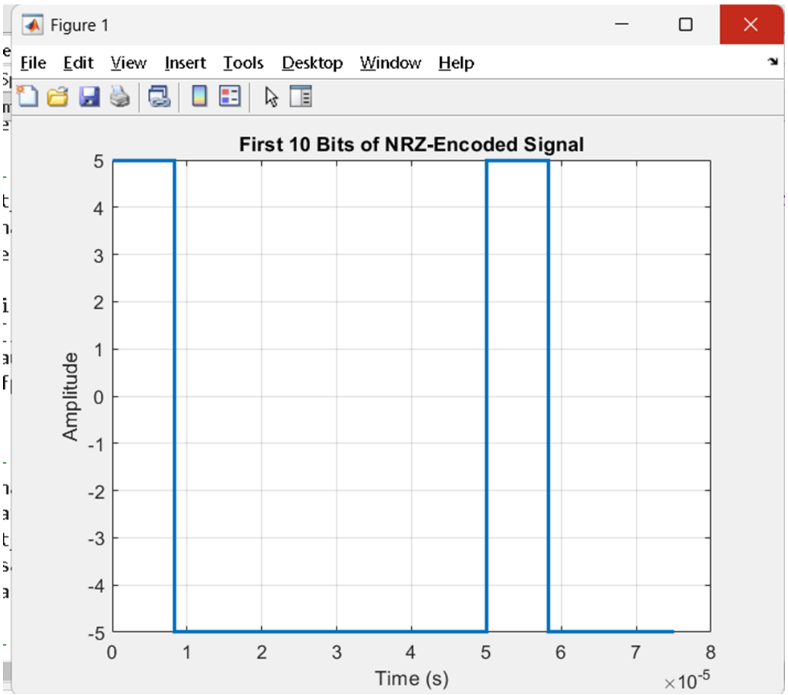

Figure 29: Polar Line Code

2.2.10 f_s =20KHz ,L=4,UniPolar Signaling :

- Sampling is the SAME..
- The Only Difference Between This and the exact one with polar representation is the Line Code , We Will Just Display The New Unipolar Line Code, Due to the Ideal Case Of the Absence of Noise There Will be no significant Differences.

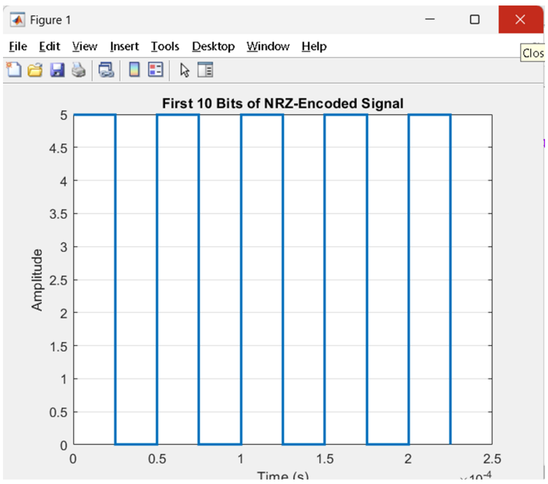

Figure 30: This is the first 10 bits With Amplitudes of 0 and 5 Volts.

2.2.11 f_s=20KHz,L=8, UniPolar Signaling

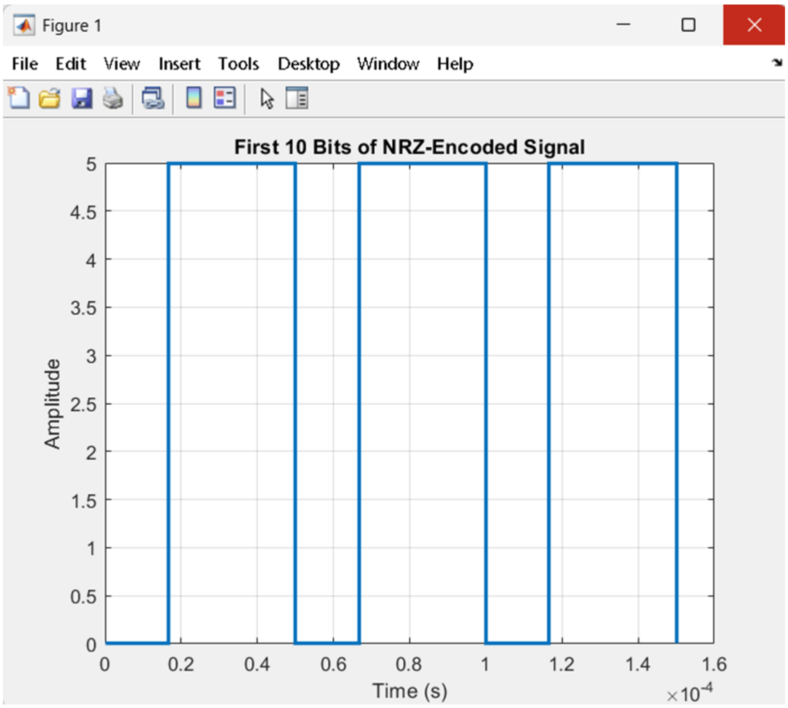

Figure 31: We Clearly See differences between this one and the one for L=4 Due to the increase of levels

2.2.12 f_s=20KHz,L=64,UniPolar Signaling

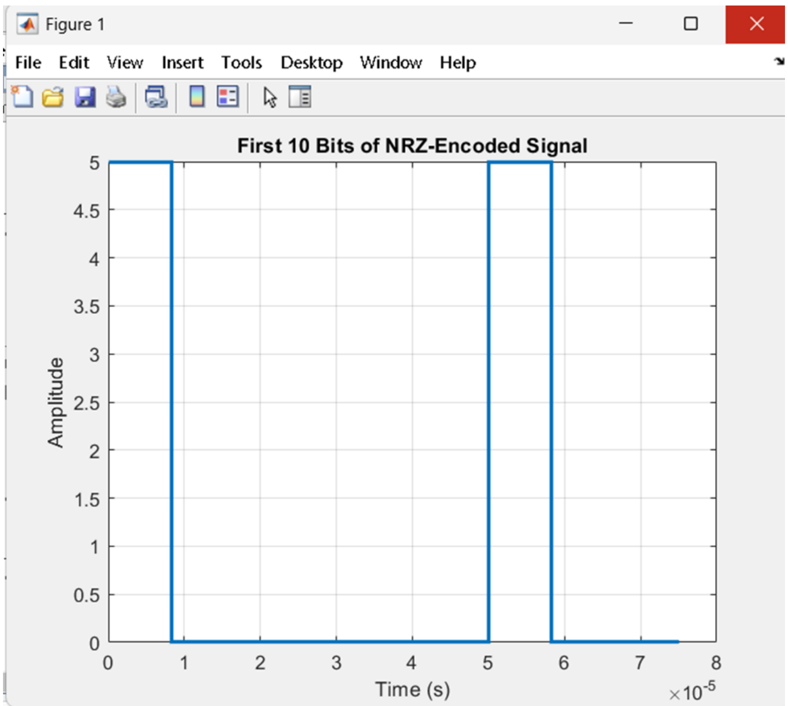

Figure 32: Difference Due to more levels = different values for mapping.

2.2.13 f_s =40KHz ,L=4,Polar and UniPolar Signaling :

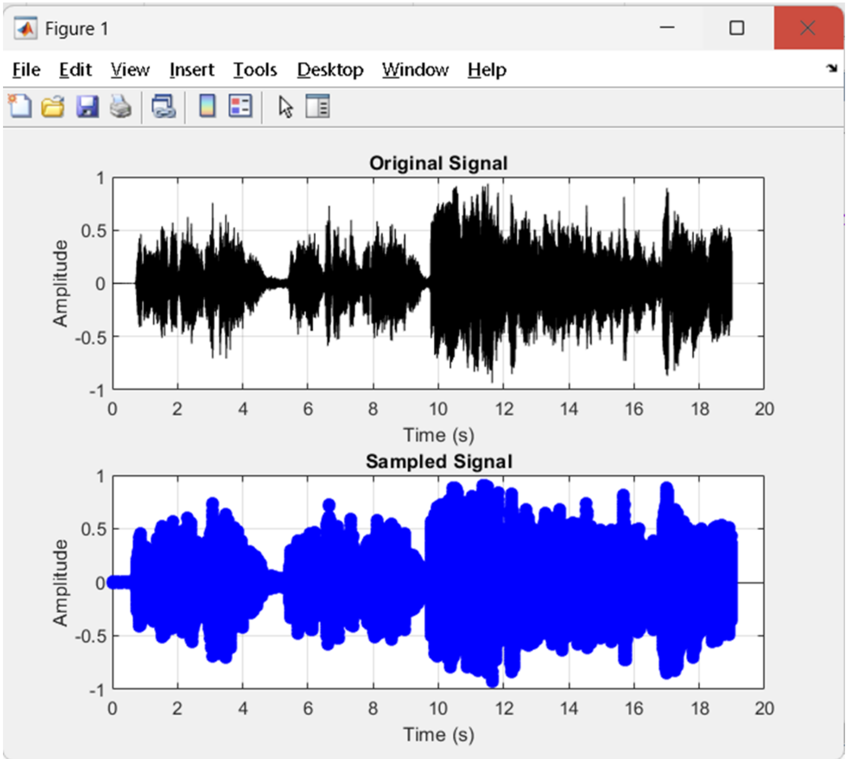

Figure 33: Increasing Sampling Frquency,Leads TO Better Audio Quality,Larger Bitstream, and in this case it might be more and more better ,CD Quality Audio is around 44K Samples per second and we are taking 40K .

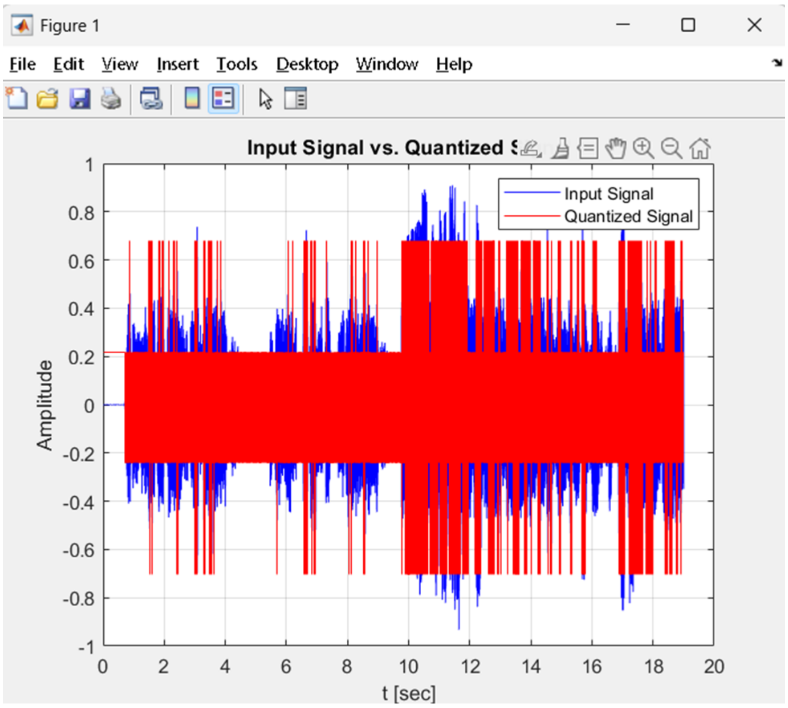

Figure 34: This is the output of the Quantizer, we clearly see that due to low Number of levels the approximations are high, and some components are missing.

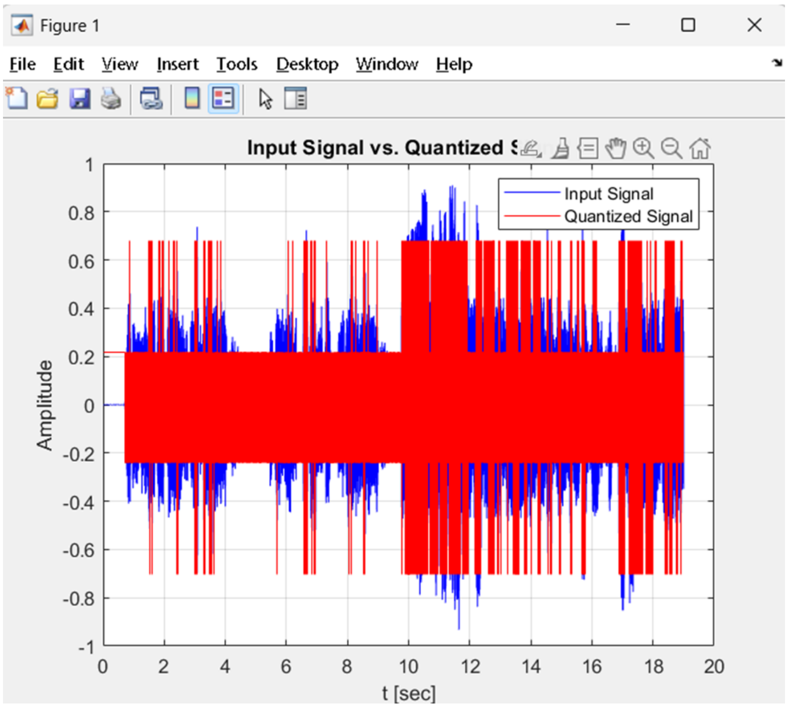

Figure 35: This is the Decoded Output of the Quantizer ad it will of course be missing main Components Due to Low Number OF Levels

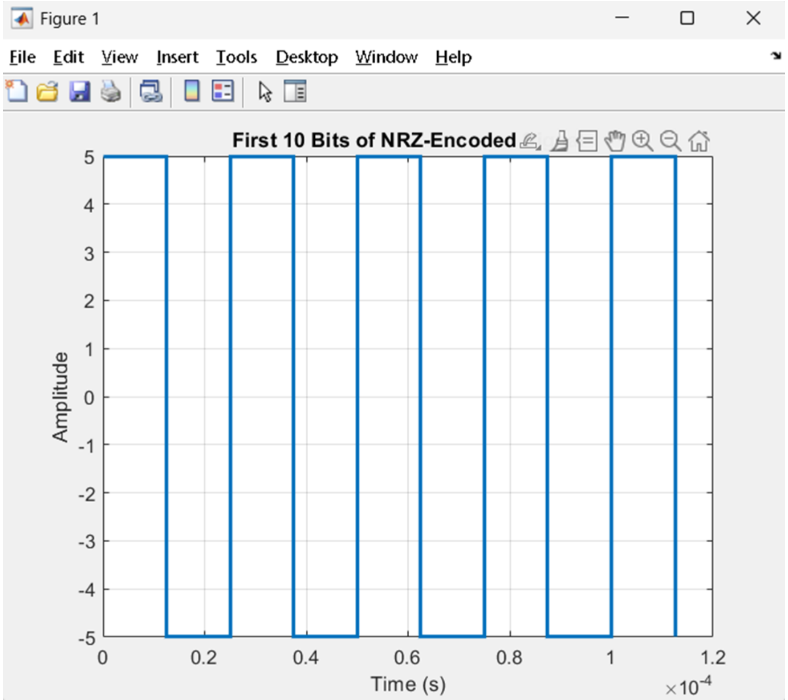

Figure 36: Encoded Signal in Polar

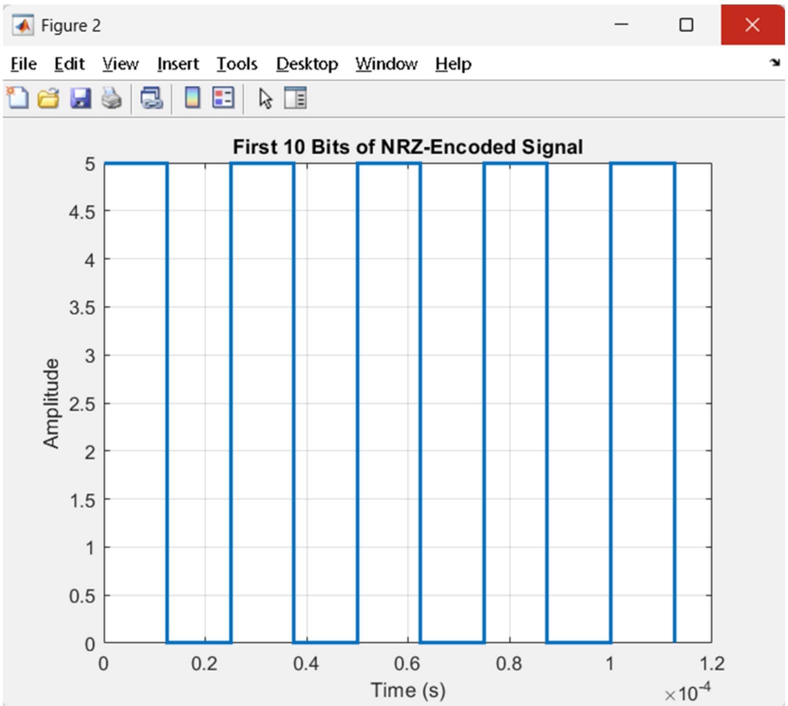

Figure 37: Encoded Signal in UniPolar

2.2.14 f_s=40KHz,L=8, Polar and UniPolar Signaling

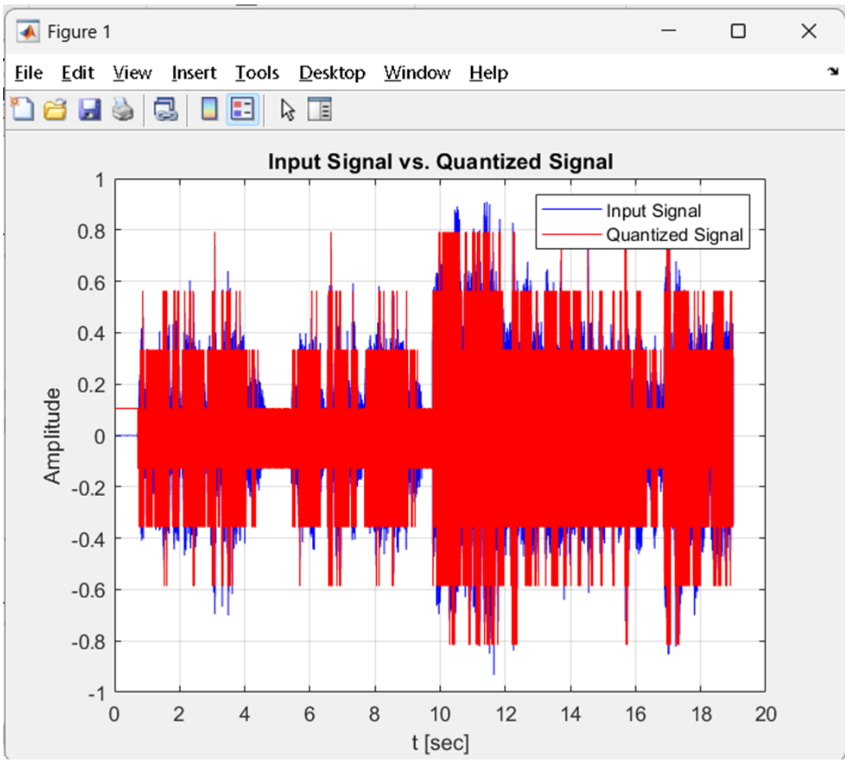

Figure 38: Adding 4 Levels to the 4 we already had gave the quantizer flexibility and takes more exact values or near exact ones.

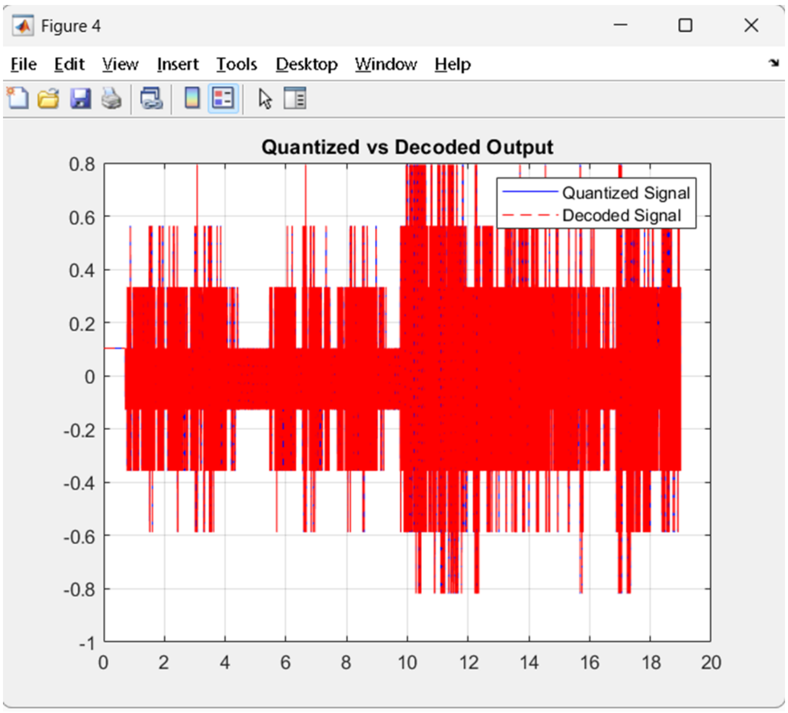

Figure 39: Adding levels Will also Affect on the decoding as more of the original signal will be retrieved.

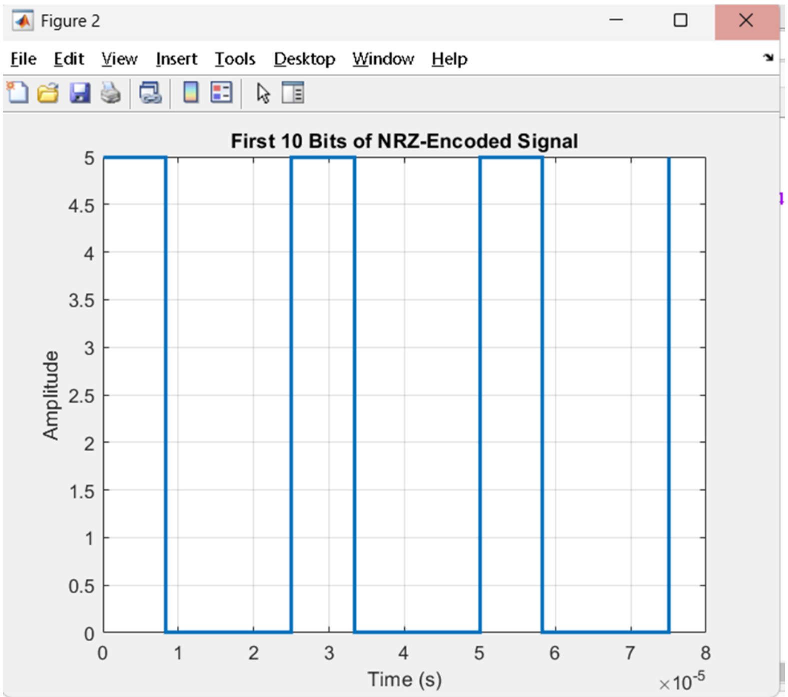

Figure 41: Unipolar Representation

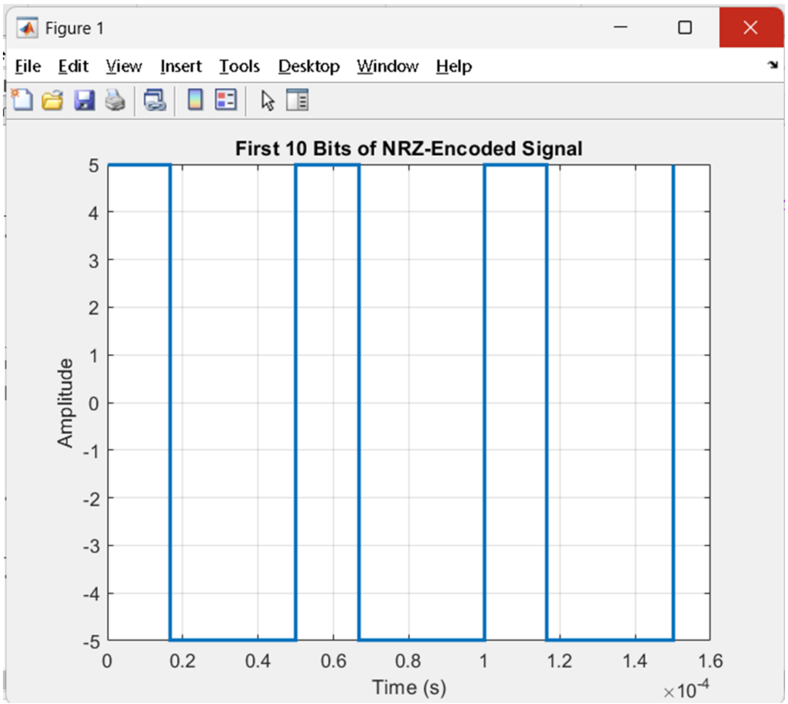

Figure 40: Line Code in Polar

2.2.15 f_s=40KHz,L=64, Polar and UniPolar Signaling

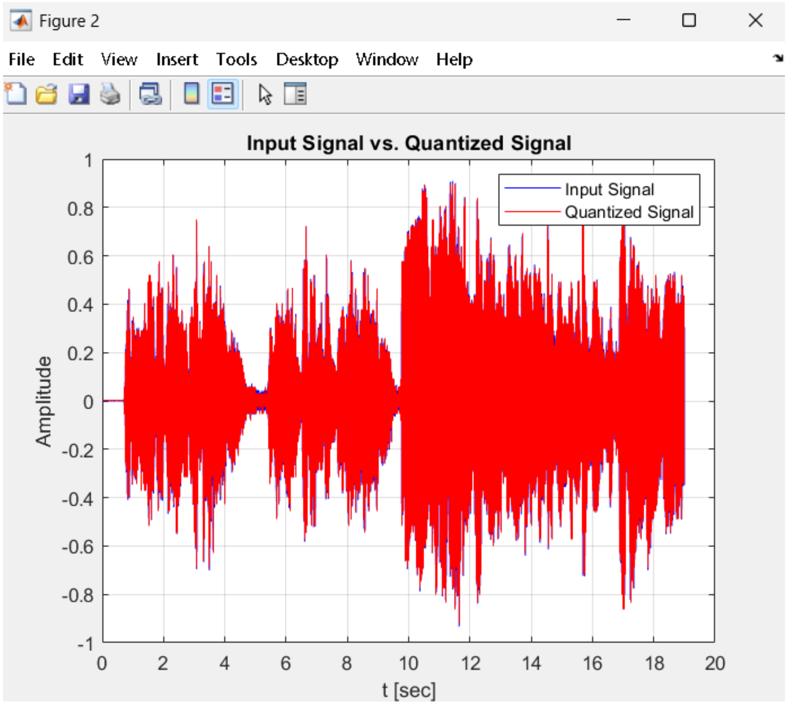

Figure 42: Combining More Levels With This f_S will be the most Clear Output, And in fact this is the best case out of all the project, Highest f_s with highest L

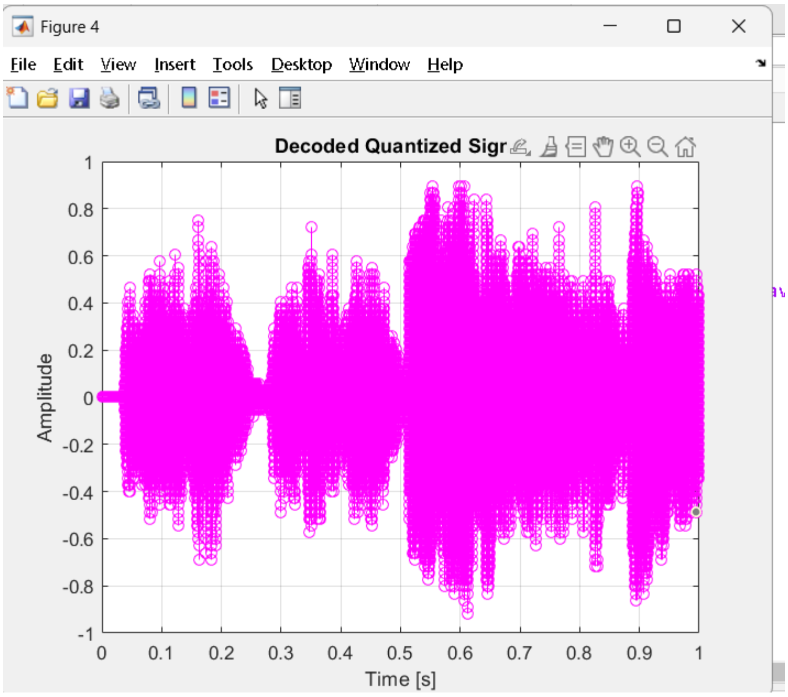

Figure 43: Adding levels Will also Affect on the decoding as more of the original signal will be retrieved, This is also the best audio outptut Result.

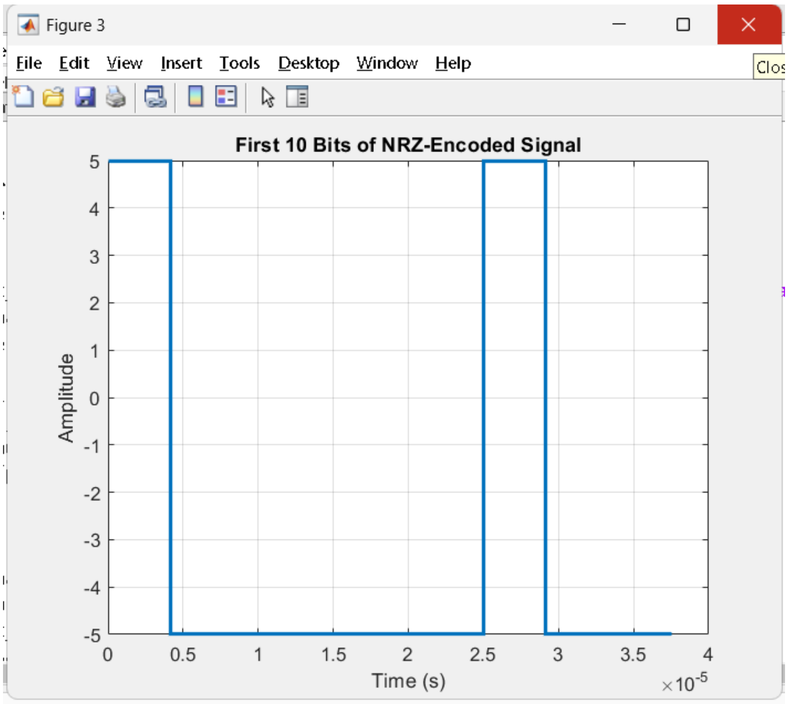

Figure 44: Polar Line Code

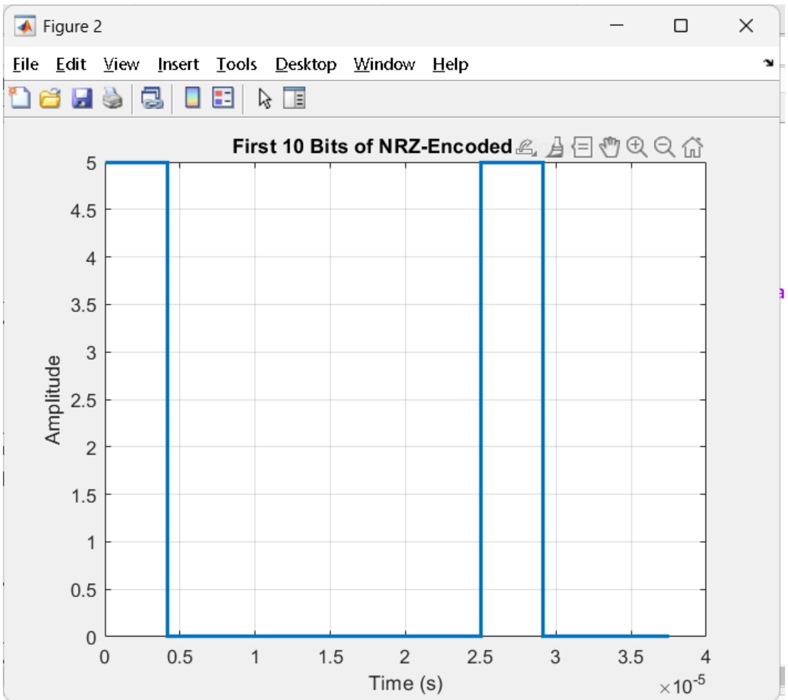

Figure 45: UniPolar Representation

3 Test Case Two

3.1 Setting Up

- Setting Up the Quantizer Amplitude to be 2 Volts.

```
%%
%--Encoder Function Definition--%
function [waveform, t_pulse, Tb, amplitude, type, n] = encoder(bitstream, fs, L)
    bits_per_sample = ceil(log2(L));
    default_Tb = 1 / (fs * bits_per_sample);

    fprintf('Auto-calculated bit duration Tb = %.6f s\n', default_Tb);
    use_default = input('Use this value? (y/n): ', 's');

    if strcmpi(use_default, 'y')
        Tb = default_Tb;
    else
        Tb = input('Enter your custom NRZ bit duration (in seconds): ');
    end

    amplitude = 2;
    type = input('Choose NRZ type (unipolar / polar): ', 's');

    n = 100; % number of samples per bit (oversampling factor)
    waveform = zeros(1, n * length(bitstream));
```

- L is set to 64 and Fs To 40000Hz.

```
sound(Signal,44100)

% --- Parameters ---
fs = 40000;
L = 64;
```

3.2 Generated Noise Channel

```
%%
%--Noise Generator--%
function [noisy_signal] = AWGN_channel(t, signal, N0)
    signal_power = mean(signal.^2);
    snr_linear = signal_power / N0;

    noisy_signal = signal + sqrt(N0) * randn(size(signal));

    figure;
    plot(t(1:100*20), noisy_signal(1:100*20));   % Show 20 bits worth if n=100
    xlabel('t [sec]');
    ylabel('Amplitude');
    title('Noisy PCM signal first 20 bits');
    legend('Channel output');
    grid on;
end
```

Figure 46: The Noise Power in dB is to be inputted when calling the function

3.3 Implementation of the Repeater

```
%--Repeater%--%
function [regenerated_PCM_signal] = regenerative_repeater_NRZ(t, PCM_signal, n, pulse_amplitude)

    A = pulse_amplitude;
    regenerated_PCM_signal = zeros(1, length(PCM_signal));
    prompt = 'Enter NRZ line code type (polar or unipolar): ';
    line_code = lower(input(prompt, 's'));

    if strcmp(line_code, 'polar')
        for i = 1:n:length(PCM_signal)-n+1
            segment = PCM_signal(i:i+n-1);
            corr_pos = dot(segment, A * ones(1, n));
            corr_neg = dot(segment, -A * ones(1, n));

            if corr_pos > corr_neg
                regenerated_PCM_signal(i:i+n-1) = A;
            else
                regenerated_PCM_signal(i:i+n-1) = -A;
            end
        end

    elseif strcmp(line_code, 'unipolar')
        for i = 1:n:length(PCM_signal)-n+1
            segment = PCM_signal(i:i+n-1);
```

Figure 47: The Repeater Function With Different Cases and comparisons whether the user chose Unipolar or Polar

```
            segment = PCM_signal(i:i+n-1);
            corr_1 = dot(segment, A * ones(1, n));
            corr_0 = dot(segment, zeros(1, n));

            if corr_1 > corr_0 + 0.1  % small margin to account for noise
                regenerated_PCM_signal(i:i+n-1) = A;
            else
                regenerated_PCM_signal(i:i+n-1) = 0;
            end
        end

    else
        error('Invalid line code type. Use "polar" or "unipolar".');
    end

    % Plotting
    nexttile
    plot(t(1:min(end, 20*n)), regenerated_PCM_signal(1:min(end, 20*n)));
    xlabel('t [sec]');
    ylabel('Amplitude');
    title(['Regenerated PCM Signal (First 20 Bits) - ', upper(line_code), ' NRZ']);
    legend('Regenerated Signal');
    grid on;
end
```

The repeater Is being called before the decoder.

```
% --- Regenerative Repeater ---
regenerated_PCM_signal = regenerative_repeater_NRZ(t_pulse, noisy_signal, n, pulse_amplitude);

% --- Decoder ---
x_dec = decoder(regenerated_PCM_signal, quant_levels, amplitude, type, n);
```

Figure 48: The Output of the Repeater is passed to the input of the decoder

3.4 Testing

3.4.1 For $\mathcal{N}_0 = 1$

1. For Polar Signaling

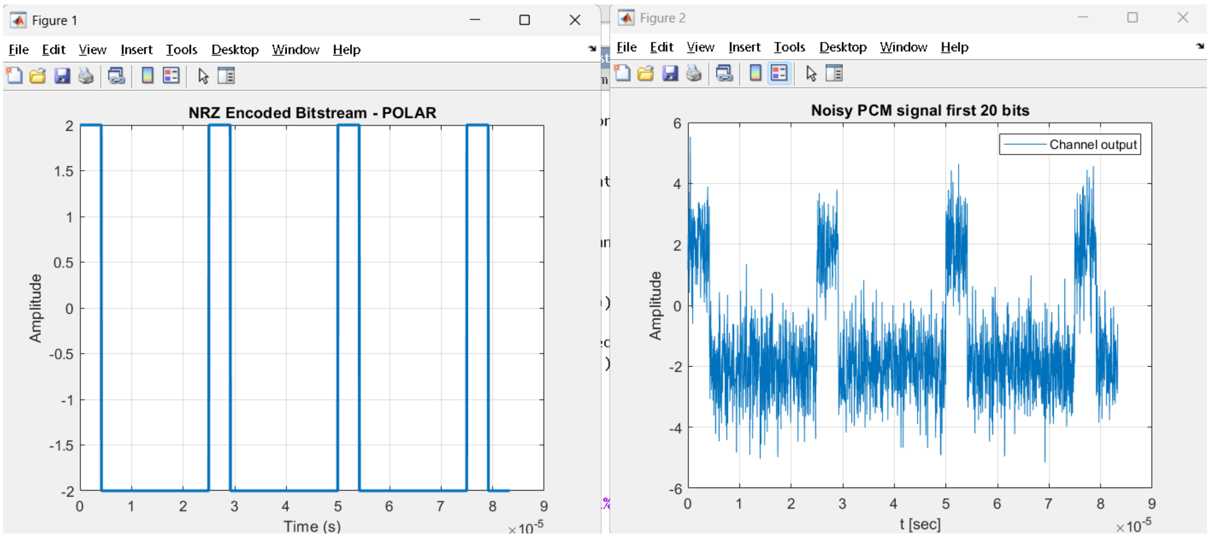

Figure 49: The output of the encoder after being affected by the noise is clearly seen that it has distortions while maintaining the +ve and -ve Amplitudes

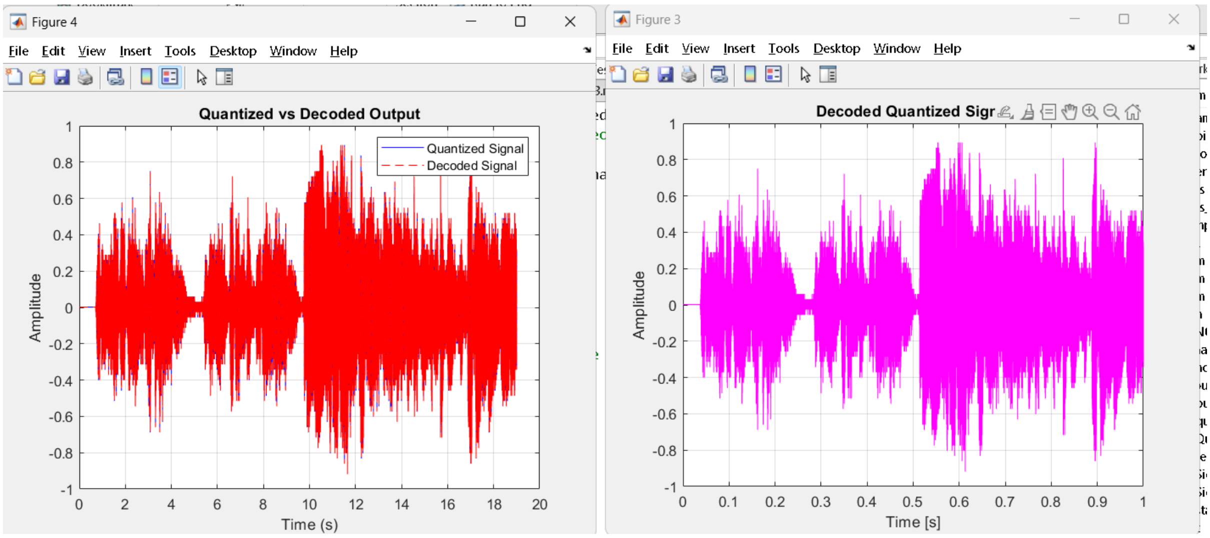

Figure 50: Quantized Vs Decoded Signal

2. For Unipolar Signaling

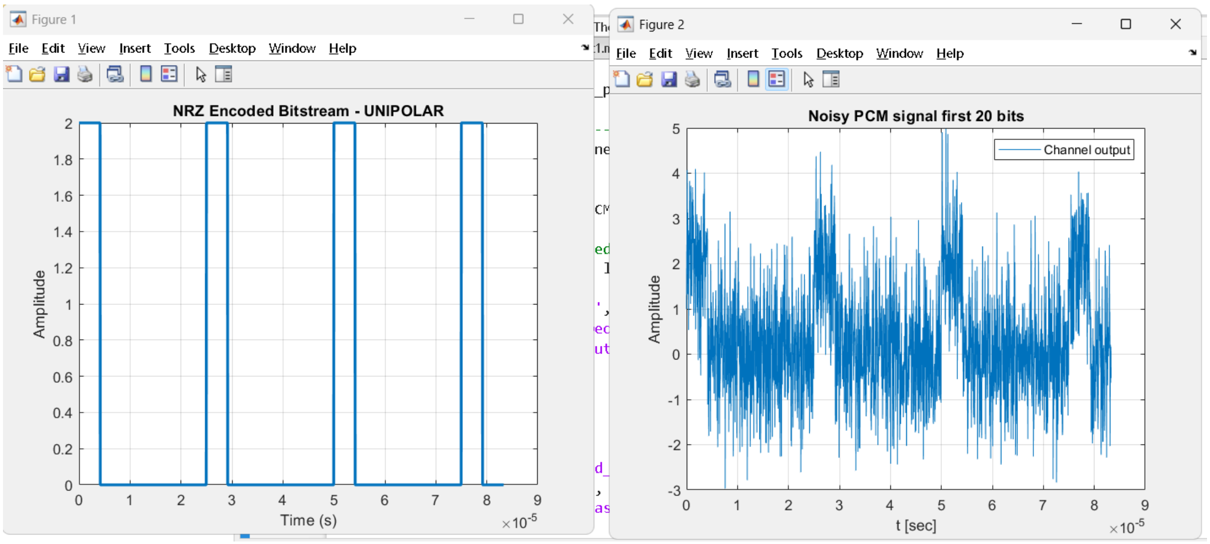

Figure 51: We can See very clearly similar behaviour but with the amplitudes centered arounf 0 and +2

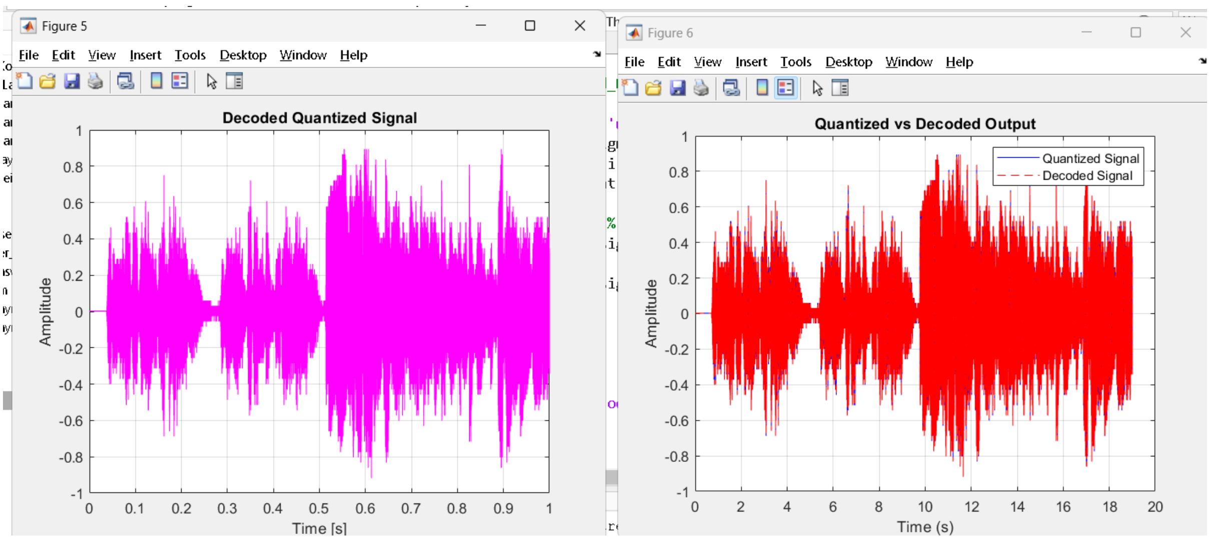

Figure 52: Also very similar behaviour

3.4.2 For $\mathcal{N}_0 = 4$

1. For Polar Signaling

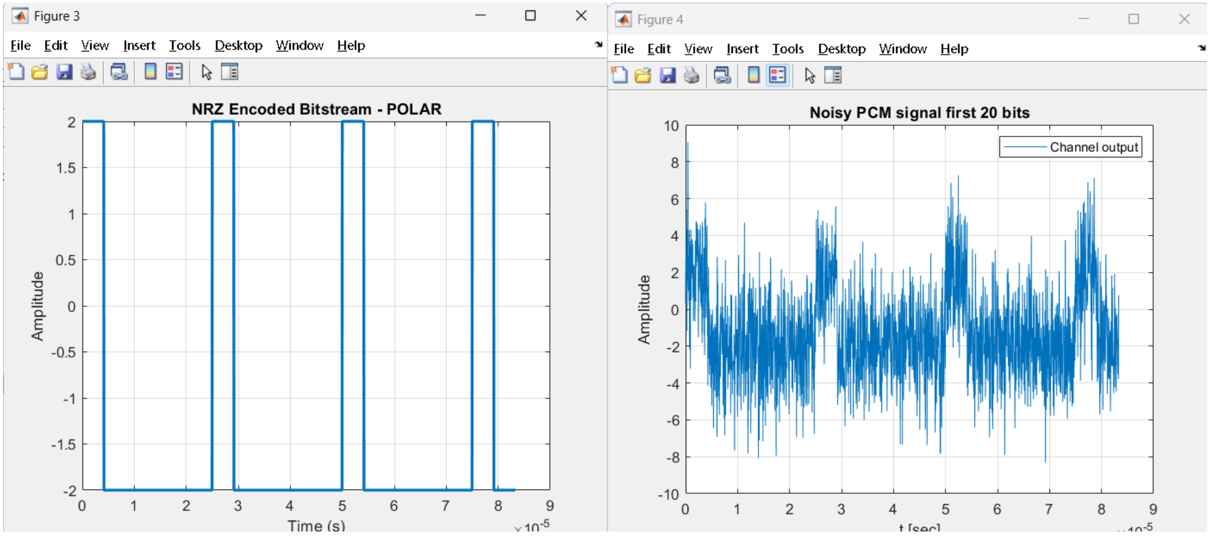

Figure 53: We Can Clearly see and Identify that in comparison with $\mathcal{N}_0 = 1$ more noise is added AKA The Amplitudes Differed and Are Shifted(Clearly Understood Because it is additive Noise).

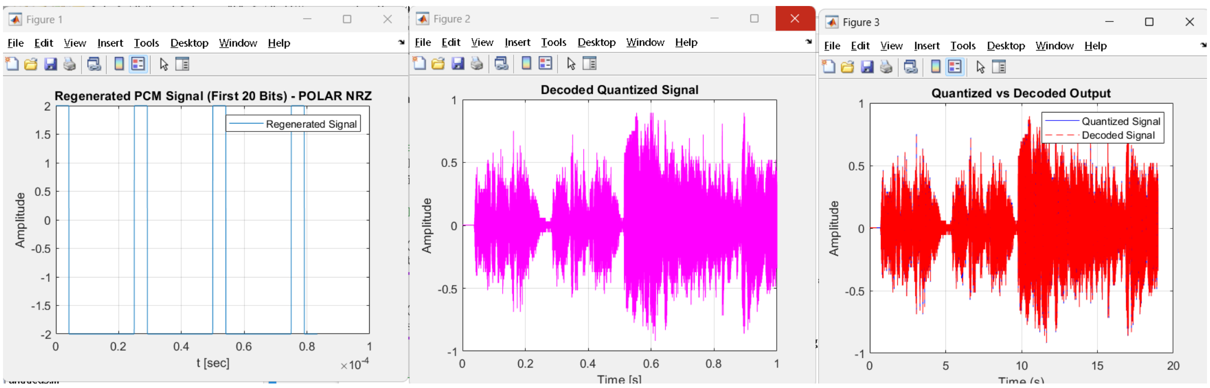

Figure 54: The Reconstructed Signal Is constructed Perfectly Thanks To the Regenerative Repeaters

2. For Unipolar Signaling:

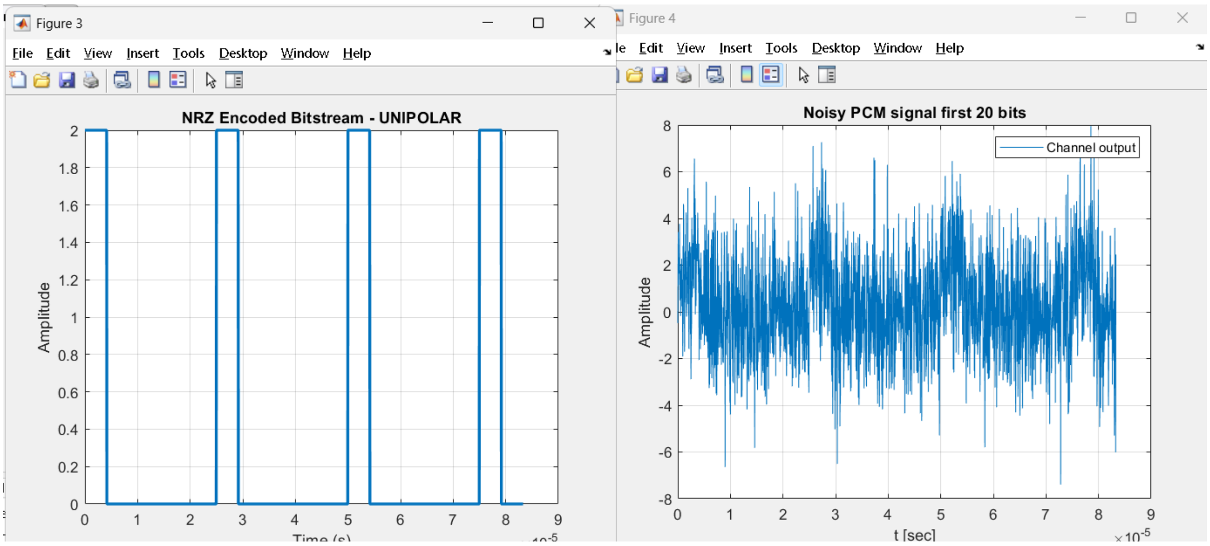

Figure 55: This Time Because of Unipolar signaling it begins to be weak against relatively Larger Noise Power

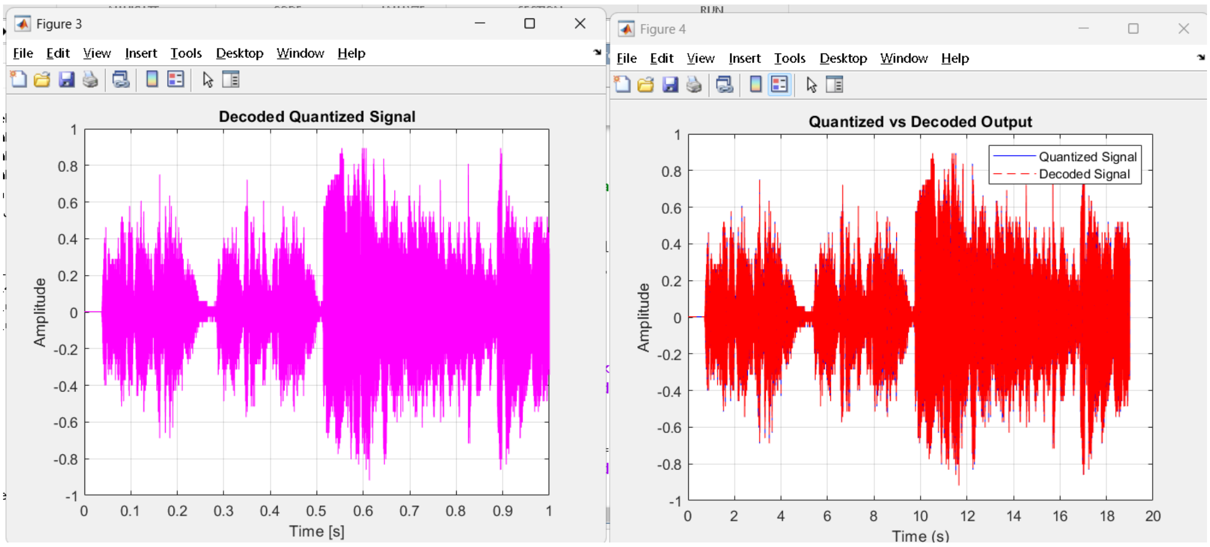

Figure 56: Reconstructed Successfully

3.4.3 For $\mathcal{N}_0 = 16$

1. For Polar Signaling:

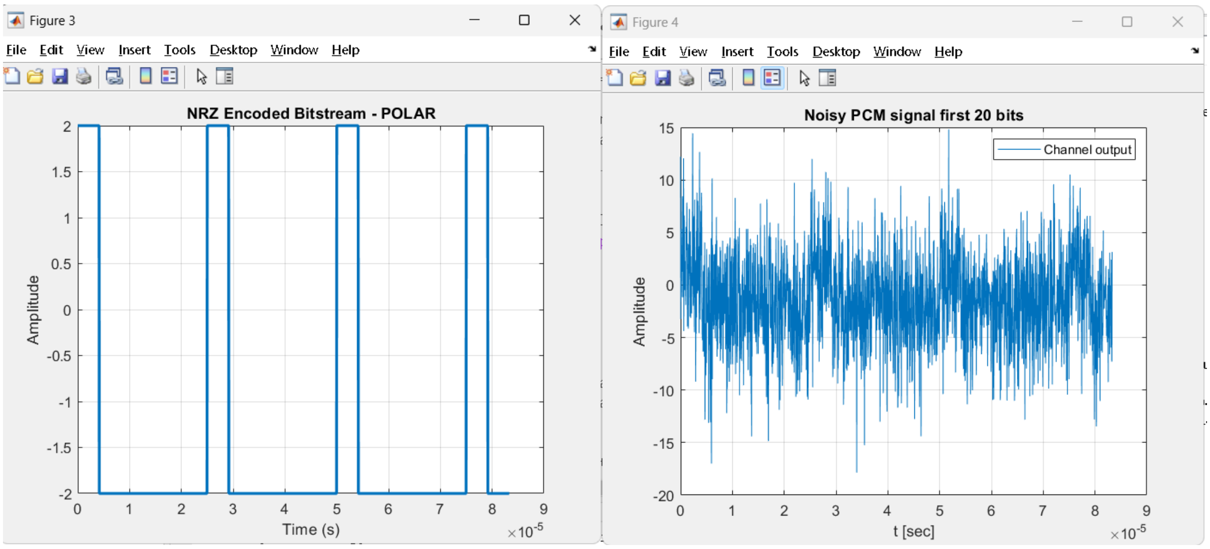

Figure 57: This time with larger noise power it added more noise AKA more Amplitudes

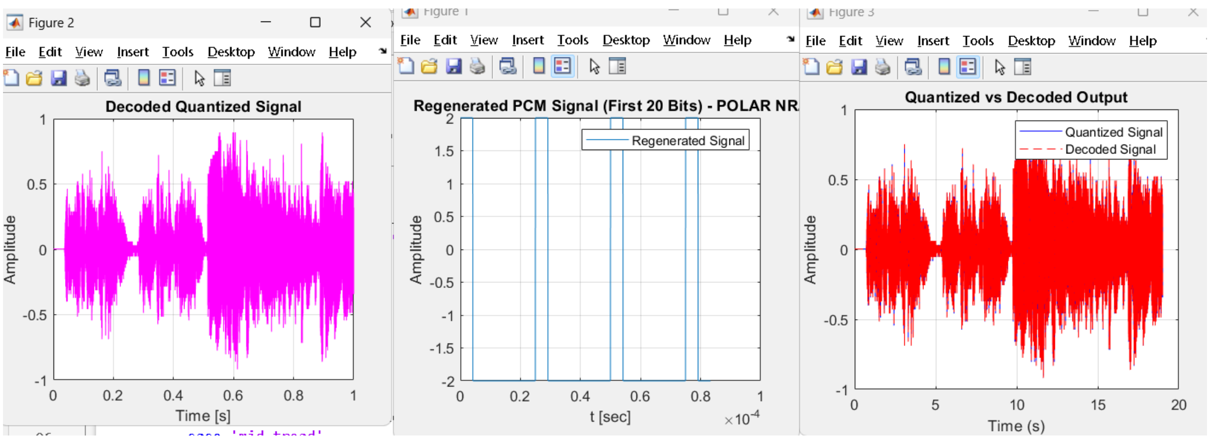

Figure 58: Reconstructed Successfully

2. For Unipolar Signaling:

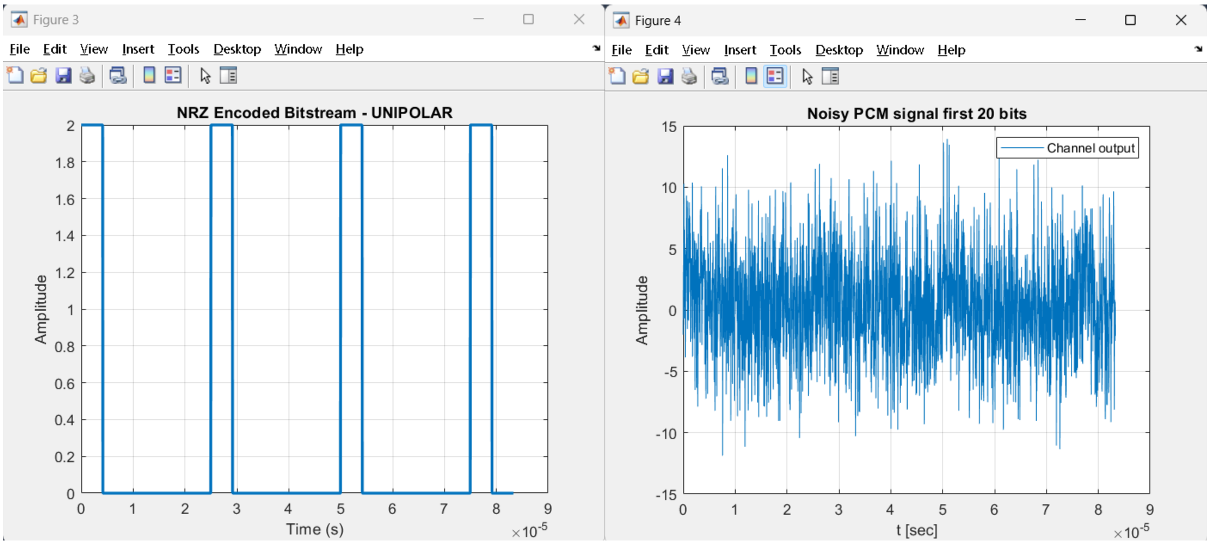

Figure 59: Obviously With Large Noise Power Unipolar is worse than Polar Because It becomes harder for the repeater to do the guess whether it is an A amplitude or A 0