**A STUDY FOR COMPARING DIGITAL MODULATION TECHNIQUES**

**PCM vs DELTA MODULATION**

|  |  |  |
| --- | --- | --- |
| **NAME** | **SURNAME** | **ID** |
| **TALİP EREN** | **DOYAN** | **21243710024** |

**TABLE OF CONTENTS**

**1. Introduction…………………………………………………….3**

**2. Pulse Code Modulation……………………………………….. 5**

**3. Delta Modulation……………………………………………… 8**

**4. Comparison of PCM vs DM…………………………………. 10**

**5. Conclusion…………………………………………………….. 13**

**6. References……………………………………………………… 14**

**7. Appendix………………………………………………………. 15**

**1. Introduction**

Claude Shannon laid down the theoretical background of digital communication in his paper "A Mathematical Theory of Communication" in 1948. After his publishment, it was followed by three new approaches in coding theory which are first nontrivial error correcting code by Golay in 1949, turbo codes by Berrou, Glavieux, and Thitimjshima in 1993, and low-density parity-check (LDPC) codes, which were first described by Gallager in 1962 [1]. Day by day, digital communication systems are improved. There existed new approaches after scientific progress. Digital communication is gathering information and data by transmitting and receiving data in digital form. The term digital form means computer language where everything consists of binary (ones and zeros) representation. It is in a discrete-time domain which means we need to sample the real-world data. Also, there exists analog communication, which is the traditional form. Analog Communication is the process of gathering information data with real-world data (continues time domain). A popular audio data handel.wav and its continuous time domain, frequency domain, and discrete-time domain are illustrated in Figure 1.1.

metin, ekran görüntüsü, çizgi, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure1.1

Modulation is the transfer of an information-carrying signal onto a carrier signal for various reasons. This application is widely used in communication systems. When we look at the modulation process from an engineer's perspective, it is the whole of the work done to transport a message signal easily and efficiently with the help of a carrier signal by consuming less power. It is varying one or more parameters of a carrier signal in accordance with the instantaneous values of the message signal [2].

As there are digital and analog communication systems, there are also digital and analog modulation types. The types of modulation techniques are shown in Figure 1.2.metin, ekran görüntüsü, diyagram, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 1.2

Digital communication systems have some advantages over analog communication systems. Not all but some reasons are brief:

**1.** Noise affects digital systems less than analog systems.

**2.** In digital systems, it is easier to work with different types of data.

**3.** The transmission method can function independently of the source. For instance, a digital transmission method that transmits voice at 10 kbps can also be used to transmit computer data at the same rate of 10 kbps.

**4.** Digital systems are less sensitive to physical effects.

**5.** Digital signals are easier to characterize [3].

This paper will be focused on comparing two different digital modulation techniques that are Pulse Code Modulation and Delta Modulation. The methods include "Modulation" in their name, but they are used for digitizing analog signals. For example, we have an audio signal. We can convert this signal into digital form with the help of Pulse Code Modulation, and Delta Modulation. If we want to transmit these signals between channels, we have to use other digital modulation techniques which are Amplitude Shift Keying, Frequency Shift Keying, Phase Shift Keying, and Quadrature Amplitude Modulation. To sum up, we digitize analog signals for some reasons in a lot of applications. Therefore, we need to convert these analog signals with the help of PCM and Delta Modulation. From now on, we will discuss the mathematical background of PCM, Delta Modulation, and their applications, comparing these techniques on a handheld audio signal.

**2. Pulse Code Modulation**

PCM is a discrete-time, discrete amplitude waveform-coding process, by means of which an analog signal is directly represented by a sequence of coded pulses [1]. Pulse Code Modulation was first introduced by Alex H. Reeves in May 1937 [4]. After his invention, it was followed by some studies that developed PCM in practical applications such as the paper "Pulse Code Modulation" written by H. S. Black and J. O. Edson. There are a lot of application areas of PCM. For example, in the aerospace industry, it is widely used in telemetry applications. A common telemetry military standard IRIG-106 has a chapter about PCM. The flight test data are transmitted with PCM codes. It is just an example of a PCM area. The mathematical background of Pulse Code Modulation is based on sampling and quantization theory. A simple block diagram of PCM is illustrated in Figure 2.1.

metin, diyagram, plan, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 2.1

In a simple PCM receiver firstly there exists an analog message source. It is the information that we will transmit. Then a low pass filter will apply to the signal. A low pass filter is used for anti-aliasing. Aliasing occurs when the sampling rate is less than the Nyquist rate [5]. Nyquist rate tells us sampling frequency should be at least two times of message signal frequency. After filtering the signal, we need to sample it. The sampling process is the first part of digitizing a signal. Digital signals are in discrete time form as mentioned in the introduction. To convert a signal continues to discrete we take samples from the signal. In the following, the signal is quantized by quantizer. Continued signals have an infinite range of amplitude. However, we are dealing with finite amplitudes. It was mentioned that digital signals are both discrete in time and amplitude. Quantization is converting amplitude to discrete from sampled signal. In the quantization process, it is important to know some parameters and mathematical expressions. Necessary equations are given in the following:

Equation 2.1

Equation 2.2

Equation 2.3

Let's consider a basic example to cover. Assume that our signal is sinusoidal and has 1V amplitude. The frequency of the signal is 1/6 Hz. Also, assume we have 2 bits per sample. First of all, we should find a number of quantization levels from Equation 2.1. It can be easily calculated as L = 4. Secondly, stepsize can be found by a formula which is 0.5. Finally, the quantization levels are -0.75, -0.25, 0.25, 0.75.

In the last part of the transmitter, there exists an encoder. The encoder is the component that converts the quantized signal to binary. It was found quantization levels in the previous paragraph. There were four quantization levels and 2 2-bit streams which means there are four binary representations. The representatives are 00, 01, 10 and 11. Assuming we have an x value which is the output of the sin function. Then binary representations are:

After the encoding process, now the signal is turned into a PCM signal. It is ready to transmit over the channels. When the signal is transmitted over the channels, noise may occur. Generally, digital systems are less affected by noise but it can be effective due to noise power. In the receiver part, the transmitted signal first comes to the regeneration circuit. The most important feature of a digital system is the ability to control errors. The regeneration circuit regenerates the signal without noise with the help of an amplifier equalizer, timing circuit, and decision-making device. Theoretically, the signal is the same as the transmitted signal. However, sometimes we cannot control the noise. Bit error occurs when it is happened [3]. Finally, the signal is decoded by the decoder and it is applied to the reconstruction filter. The analog signal is again obtained by these operations. An example of a PCM signal obtained by Handel audio is shown in Figure 2.2. In this figure, it can be seen clearly all the operations made in an analog signal.

metin, ekran görüntüsü, çizgi, öykü gelişim çizgisi; kumpas; grafiğini çıkarma içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 2.2

**3. Delta Modulation**

PCM is a great invention for digitizing analog signals. However, in some applications, we need increased transmission bandwidth. PCM has a complex structure to deal with high transmission bandwidth. Therefore, delta modulation was invented. The basic theory of delta modulation was a French patent issued in 1946 and more detailed studies have been conducted in 1952 by Jager (Philips) and Libois (CNET) [6]. Generally, it is used in audio transmission, telephone, radio communications, etc. It economizes the system by reducing system complexity while increasing transmission bandwidth. The comparison between PCM, DPCM, and DM is given in Figure 3.1.

metin, ekran görüntüsü, yazı tipi, çizgi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 3.1

The theoretical background of delta modulation is staircase approximation. The system complexity is reduced with two approaches; a single-bit quantizer and a single-unit delay element.

Important elements of delta modulation are error signal, (stepsize) and message signal. The relationship between the element and necessary equations is illustrated in the following expressions:

Equation 3.1

Equation 3.2

Also, the block diagram of delta modulation is given in Figure 3.2.

diyagram, metin, plan, teknik çizim içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 3.2

The theory comes from comparing error signals and message signals. If the error signal is positive, the message signal is increased by amount + and vice versa. Like other digital modulation techniques, delta modulation has an encoder in the transmitter. The idea of encoding here is when the message signal increases, it encodes binary "1". If the signal is reduced by -, it encodes binary "0". In every sample, we are making the same process. The important thing is determining a value for the operation. After the signal is encoded, it is ready to transmit over the channels. An example of delta modulation, staircase approximation, encoded signal, and decoded signal is illustrated in Figure 3.3.

metin, çizgi, ekran görüntüsü, öykü gelişim çizgisi; kumpas; grafiğini çıkarma içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 3.3

**4. Comparison of PCM vs DM**

In previous sections, 2 modulation types, their history, application areas, and theoretical background are discussed. While discussing these, it is emphasized that noise can affect the actual signal. Noise is an unwanted value in the signal. It distorts the real values of the signal. In communication systems, we do not want distortions because every sample of the signal is very important. For example, military applications require very high accuracy. Even a simple noise can affect the critical mission of a system. Noise may occur for different reasons. Especially the links, application areas, the distance between the systems, and hardware losses create the noise. Therefore, it is important to reduce noise in applications. As mentioned, digital systems are less sensitive to noise so it is chosen for a lot of applications. In this section, noise effects on the modulations and comparison techniques of 2 modulation types will be discussed on a noisy audio signal. An example of a noisy analog signal can be seen in Figure 4.1.

metin, ekran görüntüsü, öykü gelişim çizgisi; kumpas; grafiğini çıkarma, diyagram içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 4.1

After the encoding process, the signal is transmitted in both modulation types. However, a noise that may occur while transmission called path loss. Assume that a white Gaussian noise occurred between the channels. This noise will affect PCM and DM signals. Therefore, a noise is added to encoded signals theoretically. This noise has a signal-to-noise ratio (SNR) 20dB. The aim of this study is to compare these two modulation types under noise. Figures 4.2 and 4.3 are illustrated in the following to observe the behavior of PCM and DM encoded signals under the white Gaussian noise.

metin, ekran görüntüsü, çizgi, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 4.2

metin, ekran görüntüsü, çizgi, öykü gelişim çizgisi; kumpas; grafiğini çıkarma içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 4.3

The three most common comparison techniques are used in this study. These are signal-to-noise ratio (SNR), bit error rate (BER), and mean squared error (MSE).

Signal to noise ratio is the ratio of signal power to noise power. In communication systems, it is used very commonly. Generally, it is measured in dB scale. It is used for measuring whether signal power is higher than noise power or not. Higher SNR means better signal quality, lower SNR means worse signal quality. The mathematical formula of SNR is given in Equation 4.1.

Equation 4.1

The bit error rate is the ratio of the number of incorrect bits to total number of bits. It is also widely used in communication systems. It shows how many bits are transmitted wrong and how much it affects the system. Higher BER means there are a lot of incorrect bits and lower BER means most of the bits transmitted are correct. The mathematical formula of BER is given in Equation 4.2.

Equation 4.2

Mean squared error is an error of the average squared difference between the actual signal and the noisy signal. Generally, it is used in deep learning and image processing but it is also widely used in communication systems. It shows how similar actual and noisy signals are. Higher MSE means the signals are very different (which means noise affected the system much) and lower MSE means the signals are similar not so distorted. The mathematical formula of MSE is given in Equation 4.3.

Equation 4.3

The noisy signals are shown in Figures 4.2 and 4.3. These signals are compared by 3 metrics that are mentioned. First of all, SNR rates are not very different. The SNR for PCM is 7.02dB and 6.98dB for DM. Secondly, BER rates are compared. Like SNR, these values are also not different. BER for PCM is %5.66 while BER for DM % is 5.74. These values show us both modulations are not very different and they are strong modulation types. Finally, MSE values are compared. In this metric, the MSE of PCM is 0.05 while the MSE of DM is 0.96. Based on these measurements, both modulations were not affected much by noise however DM signal is not similar to the actual signal. It was observed that PCM is a bit better than DM. The comporison metrics belongs to PCM has better quality. To sum up, for this application PCM is more useful than DM. Demodulated signals also can be found in Figures 4.2 and 4.3.

**5. Conclusion**

In conclusion, pulse code modulation and delta modulation are digital modulation techniques. They are used for digitizing analog signals. In daily life, they are widely used for avoiding noise. Noise occurs in various situations. It affects systems accuracy. It is an unwanted condition. The theoretical background of these modulations is different. DM was invented after PCM was invented. This study is focused on comparing these modulations by 3 metrics which are signal-to-noise ratio, bit error rate, and mean squared error. The measurements tell us PCM has a bit better modulation than DM. However, it should not forgotten PCM is more complex and DM is simpler. In some applications, we need cheaper and simpler solutions. Delta modulation is a great tool for these applications. All in all, complex systems require PCM and simpler systems require DM. Future researchers are recommended to study optimizing value in DM, compare with different kinds of signal types, and better compare in different conditions.

**6. References**

**1-** Haykin, S. (2013). Digital Communication Systems. United States of America. Wiley.

**2-** Kabir, Md. H. (n.d.). Experiment on PCM Using MATLAB Software. Adjunct Lecturer, Dept. of Electronic and Telecommunication Engineering, International Islamic University Chittagong.

**3-** Haykin, S. (2000). Communication Systems (4th ed.). United States of America. Wiley.

**4-** Schwartz, M. (2003). Improving the Noise Performance of Communication Systems: 1930s to early 1940s. Charles Batchelor Professor Emeritus of Electrical Engineering, Columbia University, New York, NY 10027 USA. IEEE Communications Magazine, 41(12), 34-39

**5-** Y. Dalveren. Lecture: 9 *Digital/Analog Systems* [PowerPoint Slides]. Unpublished manuscript, EE316, Department of Electrical Electronics Engineering, Atılım University, February 2024, Ankara, Türkiye

**6-** Schindler, H. R. (1973). Delta Modulation. IBM Zurich Research Laboratory. IEEE Spectrum, 10(5), 78-83.

**7. Appendix**

**A) Main Part of Code**

clc;clear;close all;

%Basic Operations with Audio Signal

[y,fs] = audioread("handel\_audio.wav"); %Reading Handel audio file

sound(y,fs) %Sounding audio signal

t = (0:length(y)-1)/fs; %Defining time domain

figure('Name','Normal Signal','NumberTitle','off');

subplot 311;

plot(t,y) %Plotting Handel signal in Time Domain

title("Handel Audio in Time Domain");xlabel("Time(s)");ylabel("y(t)");

subplot 312;

l = length(t);

fz = (-l/2 : l/2 -1) \* (fs/l); %Defining frequency domain

y\_f = fft(y); %Converting time domain signal to frequency domain signal with Fast Fourier Transform

plot(fz,abs(fftshift(y\_f))) %Plotting Frequency domain signal

title("Handel Audio in Frequency Domain");xlabel("Frequency(Hz)");ylabel("Y(f)");

subplot 313;

downsamplingfactor = 100; %Defining downsampling factor

sampledIndices = 1:downsamplingfactor:l; %Defining sampled indices

sampledSignal = y(sampledIndices); %Sampling audio signal

sampledTime = t(sampledIndices); %Sampling time

stem(sampledTime,sampledSignal) %Plotting sampled signal

title("Sampled Version of Handel");ylabel("y(n)");xlabel("n");

%Pulse Code Modulation Signal

figure('Name','Pulse Code Modulation Normal','NumberTitle','off');

[index1,q1,SerialCode1] = PCM(y,t); %Calling PCM function

%Delta Modulation

figure('Name','Delta Modulation Normal','NumberTitle','off');

[dmDecoded1,dmEncoded1] = DeltaModulation(t,y); %Calling Delta Modulation function

%Noisy Signal Operation with PCM

figure('Name','Pulse Code Modulation Noisy','NumberTitle','off');

[index2,q2] = PCM\_noisy(y,t); %Calling PCM function with noisy signal

%Noisy Signal Operation with Delta Modulation

figure('Name','Delta Modulation Noisy','NumberTitle','off');

[dmDecoded2,dmEncoded2] = DeltaModulation\_noisy(t,y); %Calling Delta Modulation function with noisy signal

%MSE Comparison

mse\_pcm = immse(y,q2);

mse\_dm = immse(y,dmDecoded2);

fprintf('Mean Squarred Error (MSE) of Pulse Code Modulation: %.2f\n', mse\_pcm);

fprintf('Mean Squarred Error (MSE) of Pulse Code Modulation: %.2f\n', mse\_dm);

**B) PCM Function**

function [index1,q,SerialCode] = PCM(y,t)

subplot 511

plot(t,y)

title("Handel Audio in Time Domain");xlabel("Time(s)");ylabel("y(t)");

subplot 512

stem(t,y) %Plotting sampled signal

title("Sampled Version of Handel");ylabel("y(n)");xlabel("n");

Vmax = max(y); %Defining maximum amplitude of the signal

Vmin = -Vmax; %Defining minimum amplitude of the signal

nbits = 16; %Number of bits per sample

L = 2^nbits; %Number of Quantization Level

Vpp = Vmax-Vmin;

stepsize = Vpp/L; %Quantization interval size

quantizationlevels = Vmin:stepsize:Vmax; %Quantization Levels

codebook = Vmin-(stepsize/2):stepsize:Vmax+(stepsize/2); %Quantisation Values - As Final Input of quantiz

[index,quantized] = quantiz(y,quantizationlevels,codebook); %Making quantization process

NonZeroInd = find(index ~= 0);

index(NonZeroInd) = index(NonZeroInd) - 1; %MATLAB indexing from 1 to N. However we need to convert it 0 from N-1

BelowVminInd = find(quantized == Vmin-(stepsize/2));

quantized(BelowVminInd) = Vmin+(stepsize/2);

%This is for correction, as signal values cannot go beyond Vmin

%But quantiz may suggest it, since it return the Values lower than Actual

subplot 513

stem(t,quantized)

title("Quantized Handel Audio Signal");xlabel("Time(s)");ylabel("y(n)");

%Since we performed quantization, we can continue with encoder

TransmittedSignal = de2bi(index,"left-msb");

SerialCode = reshape(TransmittedSignal',[1 size(TransmittedSignal,1)\*size(TransmittedSignal,2)]);

subplot 514

grid on;

stairs(SerialCode(1:100));

title("First 100 bit of the PCM Signal");ylim([-2,2]);

%Now we can demodulate our signal

RecievedCode=reshape(SerialCode,nbits,length(SerialCode)/nbits);

index1 = bi2de(RecievedCode','left-msb');

q = (stepsize\*index1); %Convert into Voltage Values

q = q + (Vmin+(stepsize/2)); % Above step gives a DC shifted version of Actual signal

subplot 515

plot(t,q)

title("Demodulated Signal");ylabel("y(t)");xlabel("Time(s)");

end

**C) DM Function**

function [dmDecoded,dmEncoded] = DeltaModulation(t,y)

subplot 411

plot(t,y)

title("Handel Audio in Time Domain");xlabel("Time(s)");ylabel("y(t)");

delta = 0.01; %Defining stepsize

dmEncoded = zeros(size(y)); %Creating encoded signal variable with size actual signal

previousSample = 0; %Defining previous sample as 0

%Creating the encoder

%For loop compares the actual value of signal and previous sample.

%If actual signal is bigger, it increases signal by stepsize

%If actual signal is less, it decreases signal by stepsize

for i = 1:length(y)

if y(i) > previousSample

dmEncoded(i) = 1;

previousSample = previousSample + delta;

else

dmEncoded(i) = 0;

previousSample = previousSample - delta;

end

end

subplot 412

stairs(dmEncoded(1:100));ylim([-2 2]); %Plotting encoded signal

title("Dela Modulated Signal");

dmDecoded = zeros(size(dmEncoded)); %Defining decoded signal variable with size Encoded signal

previousSample = 0; %Defining previous sample 0 again

%Creating Decoder

%If encoded signal is 1, decoded signal is increased by delta

%If encoded signal is 0, decoded signal is decreased by delta

for i = 1:length(dmEncoded)

if dmEncoded(i) == 1

previousSample = previousSample + delta;

else

previousSample = previousSample - delta;

end

dmDecoded(i) = previousSample;

end

subplot 413

plot(t,dmDecoded) %Plotting decoded signal

title("Delta Demodulated Signal")

%Plotting decoded signal vs staircase approximation

subplot 414

stairs(dmDecoded(1:200),"r")

hold on;

plot(y(1:200), 'b');

hold off;

title('Staircase Approximation vs Original Signal');

end

**D) Noisy PCM Function**

function [index1,q] = PCM\_noisy(y,t)

subplot 611

plot(t,y)

title("Handel Audio in Time Domain");xlabel("Time(s)");ylabel("y(t)");

subplot 612

stem(t,y) %Plotting sampled signal

title("Sampled Version of Handel");ylabel("y(n)");xlabel("n");

Vmax = max(y); %Defining maximum amplitude of the signal

Vmin = -Vmax; %Defining minimum amplitude of the signal

nbits = 16; %Number of bits per sample

L = 2^nbits; %Number of Quantization Level

Vpp = Vmax-Vmin;

stepsize = Vpp/L; %Quantization interval size

quantizationlevels = Vmin:stepsize:Vmax; %Quantization Levels

codebook = Vmin-(stepsize/2):stepsize:Vmax+(stepsize/2); %Quantisation Values - As Final Input of quantiz

[index,quantized] = quantiz(y,quantizationlevels,codebook); %Making quantization process

NonZeroInd = find(index ~= 0);

index(NonZeroInd) = index(NonZeroInd) - 1; %MATLAB indexing from 1 to N. However we need to convert it 0 from N-1

BelowVminInd = find(quantized == Vmin-(stepsize/2));

quantized(BelowVminInd) = Vmin+(stepsize/2);

%This is for correction, as signal values cannot go beyond Vmin

%But quantiz may suggest it, since it return the Values lower than Actual

subplot 613

stem(t,quantized)

title("Quantized Handel Audio Signal");xlabel("Time(s)");ylabel("y(n)");

%Since we performed quantization, we can continue with encoder

TransmittedSignal = de2bi(index,"left-msb");

SerialCode = reshape(TransmittedSignal',[1 size(TransmittedSignal,1)\*size(TransmittedSignal,2)]);

subplot 614

grid on;

stairs(SerialCode(1:100));

title("First 100 bit of the PCM Signal");ylim([-2,2]);

%Adding noise

noisy\_signal = awgn(SerialCode,10);

noise = SerialCode - noisy\_signal;

subplot 615

stairs(noisy\_signal(1:100));title("First 100 bit of Noisy Signal");ylim([-2,2]);

%Now we can demodulate our signal

noisy\_signal = double(noisy\_signal>0.5);

RecievedCode=reshape(noisy\_signal,nbits,length(noisy\_signal)/nbits);

index1 = bi2de(RecievedCode','left-msb');

q = (stepsize\*index1); %Convert into Voltage Values

q = q + (Vmin+(stepsize/2)); % Above step gives a DC shifted version of Actual signal

subplot 616

plot(t,q)

title("Demodulated Signal");ylabel("y(t)");xlabel("Time(s)");

%Calculating SNR and BER

[ber\_pcm,ratio\_pcm] = biterr(SerialCode,noisy\_signal); %Obtaining bit errors and bit error rate with built in biterr function for PCM

fprintf('Bit Error of Pulse Code Modulation: %d\n', ber\_pcm);

fprintf('Bit Error Rate (BER) of Pulse Code Modulation: %.2f\n', ratio\_pcm\*100);

SNR\_pcm = snr(SerialCode,noise);

fprintf('Signal to Noise Ratio (SNR) of Pulse Code Modulation: %.2f\n', SNR\_pcm);

**E) Noisy DM Function**

function [dmDecoded,dmEncoded] = DeltaModulation\_noisy(t,y)

subplot 511

plot(t,y)

title("Handel Audio in Time Domain");xlabel("Time(s)");ylabel("y(t)");

delta = 0.01;

dmEncoded = zeros(size(y));

previousSample = 0;

for i = 1:length(y)

if y(i) > previousSample

dmEncoded(i) = 1;

previousSample = previousSample + delta;

else

dmEncoded(i) = 0;

previousSample = previousSample - delta;

end

end

subplot 512

stairs(dmEncoded(1:100));ylim([-2 2]);

title("Dela Modulated Signal");

%Adding noise

noisy\_signal = awgn(dmEncoded,10);

noise = dmEncoded - noisy\_signal;

subplot 513

stairs(noisy\_signal(1:100));title("First 100 bit of Noisy Signal");ylim([-2,2]);

noisy\_signal = double(noisy\_signal>0.5);

dmDecoded = zeros(size(noisy\_signal));

previousSample = 0;

for i = 1:length(noisy\_signal)

if noisy\_signal(i) == 1

previousSample = previousSample + delta;

else

previousSample = previousSample - delta;

end

dmDecoded(i) = previousSample;

end

subplot 514

plot(t,dmDecoded)

title("Delta Demodulated Signal")

subplot 515

stairs(dmDecoded(1:200),"r")

hold on;

plot(y(1:200), 'b');

hold off;

title('Staircase Approximation vs Original Signal');

%Calculating SNR and BER

[ber\_delta,ratio\_delta] = biterr(dmEncoded,noisy\_signal); %Obtaining bit errors and bit error rate with built in biterr function for PCM

fprintf('Bit Error of Delta Modulation: %d\n', ber\_delta);

fprintf('Bit Error Rate (BER) of Delta Modulation: %.2f\n', ratio\_delta\*100);

SNR\_delta = snr(dmEncoded,noise);

fprintf('Signal to Noise Ratio (SNR) of Delta Modulation: %.2f\n', SNR\_delta);

end