Analog to Digital Conversion

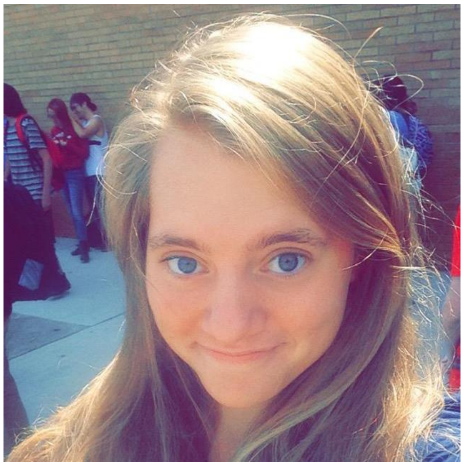
Meredith Matteson*

Mentor: Aysha Shanta

*Lenoir City High School
The University of Tennessee Knoxville

Young Scholars Program, CURENT, EECS, UTK

About the Authors



Meredith Marie Matteson

Meredith Marie Matteson is an upcoming senior at Lenoir City High School (LCHS) in Lenoir City, Tennessee. She is the President of the school's Chapter of National Honor Society as well as the Vice President of Mu Alpha Theta, the Mathematics Honor Society. Meredith has a Varsity Track and Field Letter and is in the Lenoir City High School top Choir Ensemble, The LCHS Singers. She has participated in the Newspaper Tower Competition and the Food Battery Competition at the annual Engineer's Day at the University of Tennessee.

Aysha Siddique Shanta

Aysha Siddique Shanta received the B.Sc. degree in electrical engineering from BRAC University, Dhaka,Bangladesh, in 2011, and is currently working towards the Ph.D. degree in computer engineering at the University of Tennessee,Knoxville, TN, USA. Her research interest include microelectrode design with carbon nanofibers and/or carbon nanospikes for electrochemical interfaces and she is also interested in device modeling.

Abstract

This project is about an analog to digital converter. The task is to write codes in MATLAB that convert an analog voice recording into a digital signal. The original recording has both continuous time and continuous valued the final digital signal will have both discrete time and discrete value. The digital recording can be transmitted over the communication channel and processed digitally at the receiving end.

Table of Contents

I. INTRODUCTION

Analog to Digital conversion (ADC) is a process that has very important applications in the modern world. Since most modern devices are digital, all analog signals must be converted to digital signals. Digital devices are cheaper and more dependable. In addition, digital computers have the ability to manipulate, transmit, and process data in ways never thought possible before. Without this conversion, most of our modern technology would be unusable.

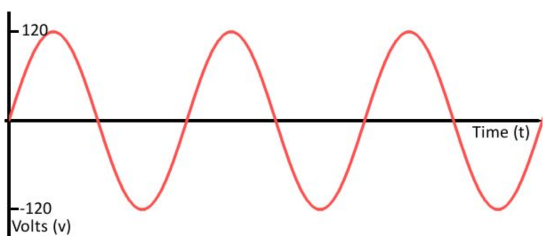


Figure 1: Analog Signal [1]

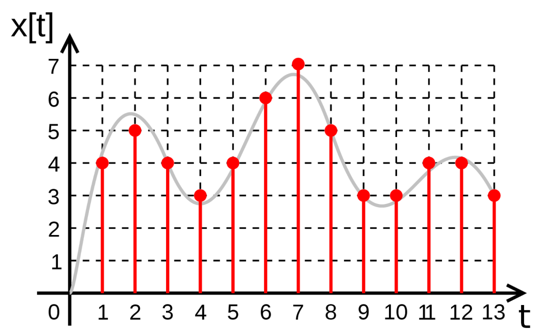


Figure 2: Digital Signal [2]

Analog signals have continuous waves and an infinite number of data points as shown in Figure 1. Digital signals are made up of just enough of those points so that it is possible to reconstruct the wave later at the receiving end. An example of a digital signal is shown in Figure 2. The process takes continuous data and makes it more efficient and practical digital data. This digital data can then be transmitted from one place to another and processed digitally. The analog signal is inputted into an ADC to convert it into a digital input signal which is then inserted into a digital signal processor and changed into a digital output signal. On the receiving end, it is reconstructed into an analog signal using an Digital to Analog converter [6]. Figure 3 represents a block diagram of the mentioned process. .

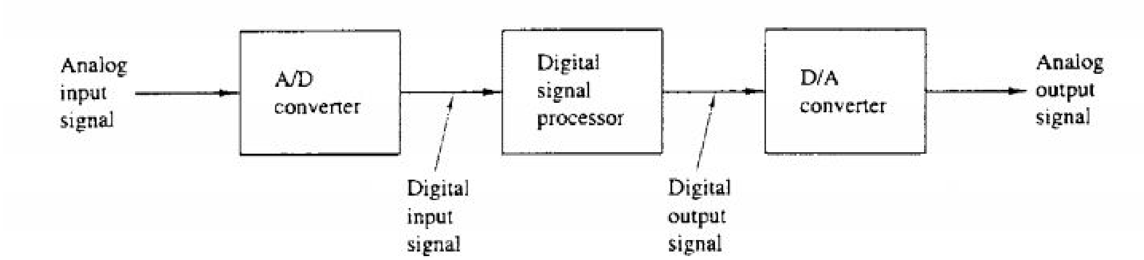


Figure 3: The process of analog to digital conversion and digital to analog reconstruction [6].

A perfect example of this concept is the telephone. When calling someone, the voice acts as the analog signal. It is converted into a digital signal using sampling, quantization, and coding, then transmitted and reconstructed on the other end of the telephone line. This is why a person sounds different on the telephone than he or she does in real life. Some other examples of an ADC are radio, cameras, video recorders, computers, and television. Digital signals are more practical due to the fewer amount of data points. The fewer data points allow the data to be transmitted and processed faster. In order to use any type of digital device, like a computer for example, the signal must be digital. In the modern era, where computers are such a huge part of everyday life, analog to digital conversion is essential.

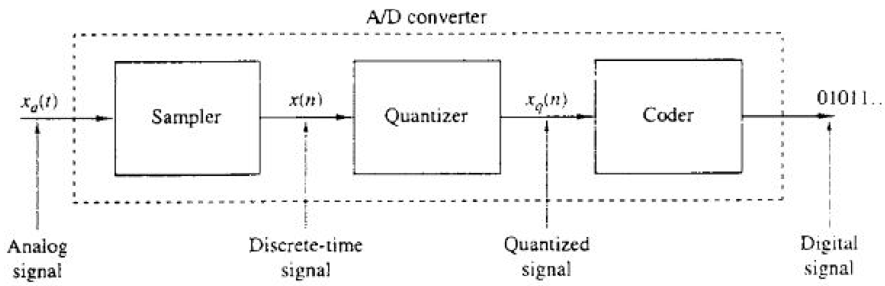


Figure 4:Process of analog to digital conversion [6]

Only the conversion from analog to digital signal has been completed in this project. This portion is made of three different steps: sampling, quantization, and coding. A more detailed description is shown in Figure 4 where each step must be completed before another can begin.

II. LITERATURE REVIEW

Sampling is the process of changing the x-axis quantities from continuously timed to discretely timed, as shown in figure 5. After sampling, the signal will have a definite number of points on the x-axis instead of an infinite number of points.

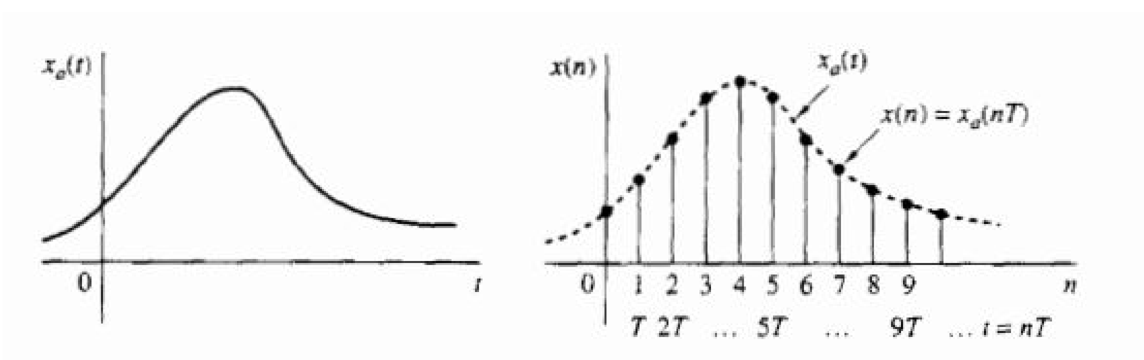


Figure 5: Process of Sampling [6]

The process simplifies the x-axis to a point where it is workable and reasonable. Having millions and millions of data points, when considerably less are needed, is inefficient and unnecessary.

Sampling

Sampling should consist of enough data points so that the signal can be reconstructed accurately. According to the Nyquist - Shannon sampling theorem, this means that the sampling frequency should be at least twice the actual wave frequency [6]. For example, if the frequency of the wave is 100 hertz, the sampling frequency should be at least 200 hertz.

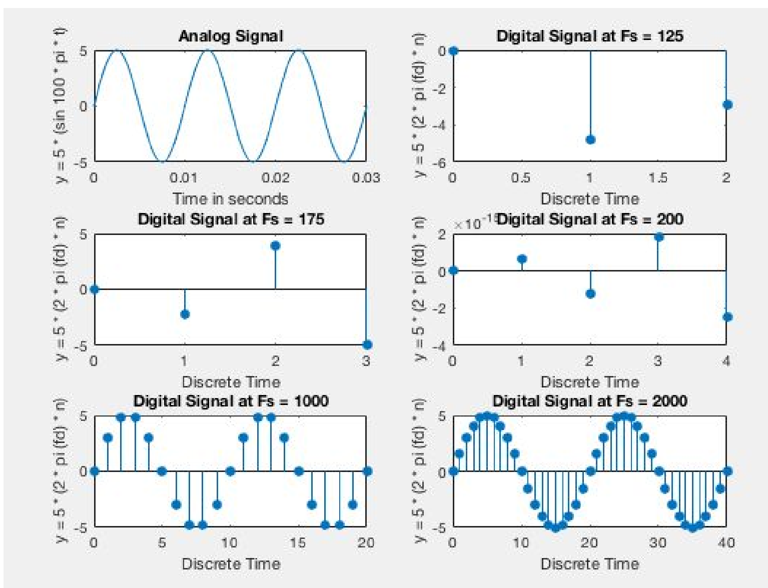


Figure 6: Testing to the Nyquist-Shannon Sampling Theorem

In some cases, it is better to sample at a rate three or four times that of the original frequency at a minimum, as described by Figure 6. Figure 12 in the Appendix is an example of a sampling code. As a result of this concept, as frequency increases, sampling rate must also increase.

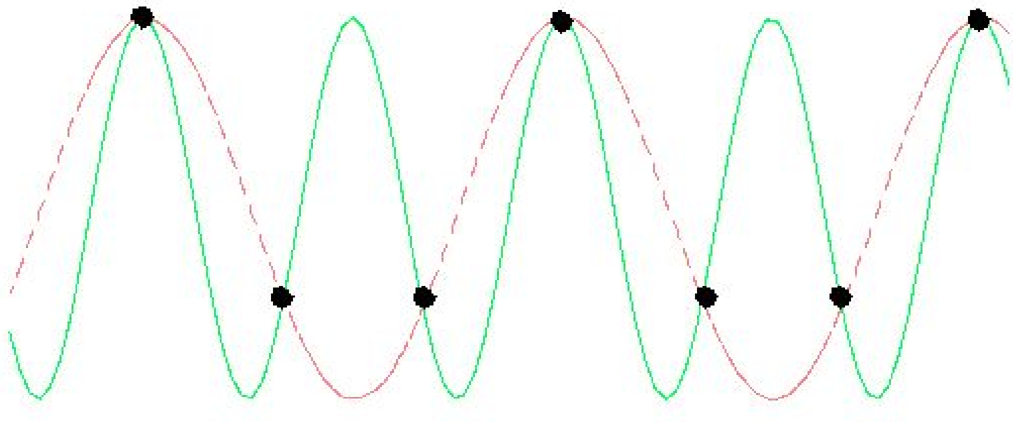


Figure 7: The green line represents the actual wave; the points represent the sample. Notice that the reconstructed signal (red line) is not the same as the green line [5].

If the sampling frequency is too small, the samples cannot be accurately interpolated to get the actual signal at the receiving end. An example is shown in Figure 7. The more data points you have the easier it will be to reconstruct the wave, but having an unnecessary number of data points is counterproductive.

Quantization

The next step is quantization. The digital device can only communicate Y axis data at specific levels. Figure 8 shows a graphic representation of the concept.

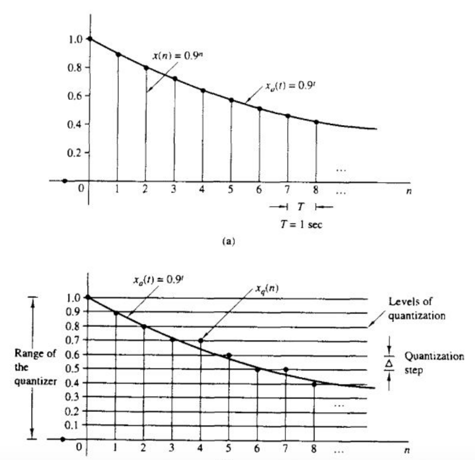


Figure 8: The graphic describes the process of quantization [6].

The more quantization levels there are, the more accurate and reconstructable the plot will be. Number of quantization levels are assigned using the number of bits. The equation for determining the number of levels = $2^{\text{number of bits}}$. So if the number of bits is 8, the number of levels is $2^8 = 256$. The range, from the maximum height to the minimum height, will have 256 different levels. Most telephones use an 8 bit quantizer. This converts the continuously valued y-axis into a discretely valued y-axis. Quantization has one completely inevitable inaccuracy: quantization error. The fact that the computer abandons the original y-axis values and assigns another number in the place of it is a loss of data. However, the only way to reduce the quantization error is to increase the number of levels which in turn reduces the quantization step. Fewer quantization levels makes the process more efficient in transmission and processing but it will cause a larger quantization error. More quantization levels makes the process less efficient but the reconstruction of the data will be more accurate.

Coding

The next step is coding. Coding is a very important step. It converts the quantization level indices into a binary sequence. Digital devices only understand low level languages, like binary for example. As a result, the data points must be converted to a series of 1's and 0's for the computer to process it.

The combination of all three steps converts a signal with continuous time and continuous value to one that has discrete time and discrete value.

III. METHOD

This project's task was to sample, quantize, and code the conversion from an analog signal to a digital signal. The original wave is the analog signal. In this experiment, the wave is a recording of the voice saying, "Hello. My name is Meredith." The computer automatically changed the recording into a sampled digital signal since computers cannot store analog signals. So, the first step of the process is resampling. This audio signal was resampled at 8 KHz as this is the most common sampling rate among telephone communication.

For this experiment, the computer utilized the function *[y, Fs, bits] = audioread('')* to read the audio signal. This built in function reads the audio file and outputs the sample frequency, number of bits used in quantization, and the y-values. Then the program utilized the function *x = resample(y, p, q)* where *p* is a new smaller sampling rate, *q* is the sampling frequency and *y* will be recognized from the audioread equation.

For quantization, a 4 bit quantizer was used. Seeing that number of levels = $2^{\text{number of bits}}$, so there will be 16 distinct levels. The results will show a large difference between the analog recording and digital recording. For a closer look at the program, see Figure 13 in the Appendix.

In this project the computer used the *variable = dec2bin()* built in function. It converts the decimal values into binary numbers. The variable in the parentheses was index, the variable representing the indices of the quantization levels. This process converted the level indices to binary.

All of the method is computerized. For every step, a MATLAB code was written. More details about the final code can be found in Figure 14 in the Appendix.

IV. RESULTS

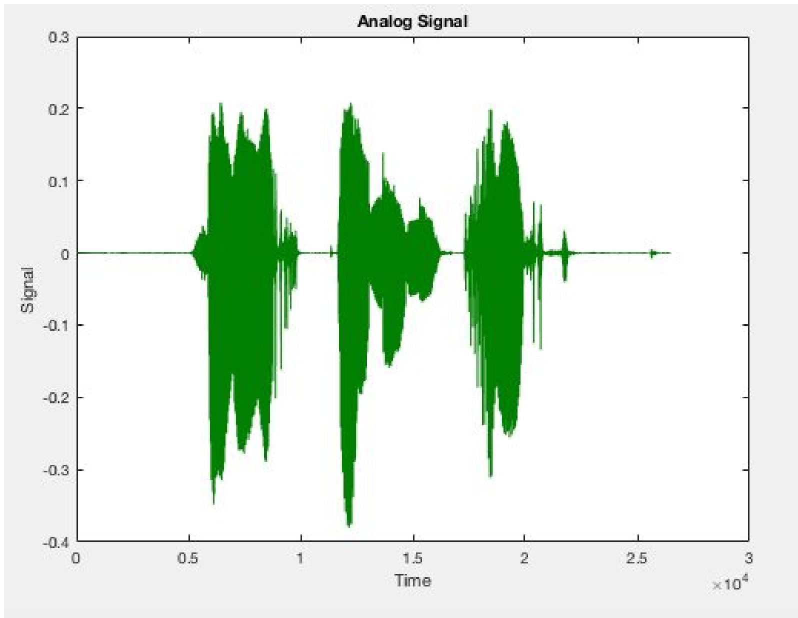


Figure 9: The original voice recording

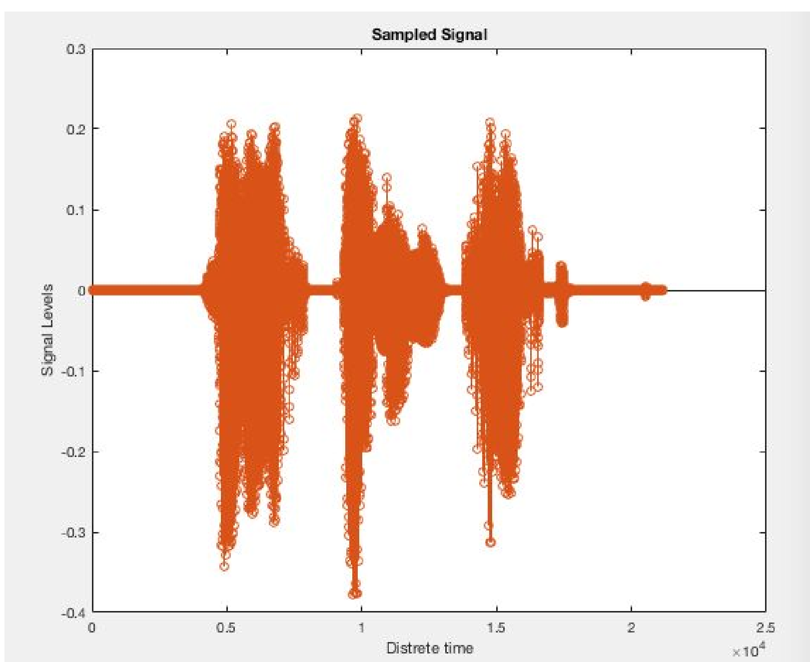


Figure 10: The sampled audio recording

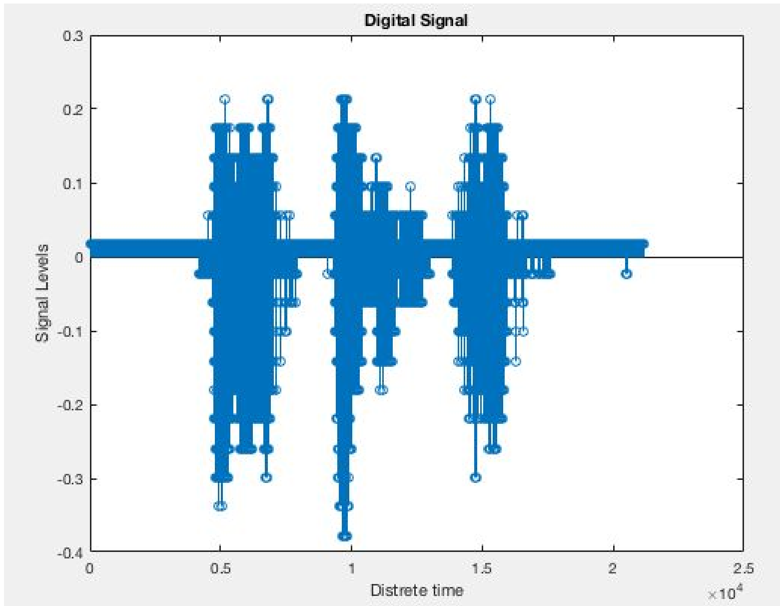


Figure 11: Final digital signal

V. DISCUSSION

The graphic of the original audio signal, as shown Figure 9. It displays an infinite number of points. The sample signal, as shown in Figure 10 and displays the position of each of the thousands of samples. The final digital signal as shown in Figure 11. 16 different levels of quantization can be seen.

The difference between the original analog signal and the sampled digital signal is significant. The only major difference between the two figures is that the sampled recording displays each sample. Each sample is represented by an open circle. Every time one puts any kind of continuous information into a computer, like an image or audio recording, the computer changes it from analog data into digital data. The sampling defines the number of points and where they will occur.

Figure 11 shows the digital signal where the previously sampled signal has been divided into 16 different levels after using the 4 bit quantizer.

VI. CONCLUSION

The process that so much of modern life is based around is a mystery to many people. The MATLAB coding involved has been engineered for maximum efficiency. The process of sampling and quantization is supposed to simplify on the x and y-axises so that the data can be made digital. If the data was not transferred to digital, it would not be able to be used with any digital technology: rendering it nearly useless in today's society.

VII. ACKNOWLEDGEMENTS

I would like to thank, my mentor, Aysha Shanta for all of her valuable help during the process; the success of this project would not have been possible without her. I would like to thank all of the CURENT faculty and staff members. I would like to thank my fellow Young Scholars for helping and supporting me throughout the various challenges associated with this project.

This work was supported in part by the Engineering Research Center Program of the National Science Foundation and the Department of Energy under NSF Award Number EEC-1041877 and the CURENT Industry Partnership Program.

VIII. REFERENCES

[1]. Analog vs. Digital [Graphic of just the analog signal]. (n.d.). Retrieved July 5, 2016, from https://learn.sparkfun.com/tutorials/analog-vs-digital/digital-signals

[2]. Digital.signal.discret.svg [Graphic of a digital signal]. (2014, August 27). Retrieved July 5, 2016, from https://commons.wikimedia.org/wiki/File:Digital.signal.discret.svg

[3]. Experiment 01: Sampling and Quantization of signals using MATLAB. (n.d.). BRAC University. Retrieved July 13, 2016.

[4]. [Graphic of a low frequency curve and a high frequency curve]. (n.d.). Retrieved July 5, 2016, from http://metaldetectorsa.co.za/frequently-asked-questions/frequency-and-your-metal-detector/

[5]. Marshall, D. (2001, April 10). Nyquist's Sampling Theorem. Retrieved July 13, 2016, from https://www.cs.cf.ac.uk/Dave/Multimedia/node149.html

[6]. Proakis J.G. and Manolakis D. G. (1996). Digital Signal Processing: Principles, Algorithms, and Application. New Jersey: Prentice-Hall, Inc.

IX. APPENDIX

```
function [sample_signal, analog, time] = sample_funct(freq, freq_s, A)

%This fucntion will make a regular sine wave then sample from the wave at
%regular intervals.

per = (1/freq);

t_plot = 0 : (per/freq_s) : ( 3 * per) ;
analog = A * sin(2 * pi * freq * t_plot);

fd = (freq/freq_s);
td = 1/fd;

    time = 0:1:(2 * td);
    sample_signal = A * sin( 2 * pi * fd * time);

subplot (2,1,1), plot(t_plot, analog), title('Analog Signal'), xlabel('Time in seconds'), ylabel('y = A * sin(2 * pi * freq * t')
subplot (2,1,2), stem(time, sample_signal), title('Digital Signal'), xlabel('Discrete time'), ylabel('y = A * sin(2 * pi * freq * t')

%[sample_signal, analog] = sample_funct(freq, freq_s, A)
```

Figure 12: Sampling Code

```
function [ index, xq, q_e ] = quantization(x, b)

%This function takes sampled data and quantizes it.

L = 2.^b;
lev_id = 0: 1 : L - 1;

sigmax = max(x);
sigmin = min(x);

Qstep = (sigmax - sigmin) / (L - 1);

siglevel = sigmin : Qstep : sigmax;

    for i = 1 : 1 : length(x) ;
        dif = abs(x(i) - siglevel);
        id = find(dif == min(dif));
        id = id(1);
        xq(i) = siglevel(id);

        index(i) = lev_id(id);
        e(i) = x(i) - xq(i);

    end

q_e = (sum(e.^2) / length(e));
```

Figure 13: This code was actually used in the quantization of the voice.

```
[y, Fs] = audioread('Voice_recording.wav.m4a');

p = 8000;
q = Fs;

x = resample(y, p, q);

[ index, xq, q_e ] = quantization(x, 4);

figure(88);
subplot(3,1,1), plot(x), title('Analog Signal'), xlabel('Time'), ylabel('Signal')
subplot(3,1,2),stem(x), title('Sampled Signal'), xlabel('Distrete time'), ylabel('Signal Levels')
subplot(3,1,3),stem(xq), title('Digital Signal'), xlabel('Distrete time'), ylabel('Signal Levels')
```

Figure 14: Entire program