SIGNAL DIGITISATION AND RECONSTRUCTION
ID: 20493425
DEGREE: MSC ELECTRONIC COMMUNICATIONS AND COMPUTER ENGNEERING

MODULE: DIGITAL COMMUNICATIONS EEEE3007

Table of Contents

Introduction

Pulse Code Modulation (PCM) is a concept in digital communication that is used in a several applications, such as digital audio and video processing, telecommunications, and data compression. The process of PCM involves converting continuous-time signal into a discrete-time signal or analogue signals into digital form, which can be represented by a sequence of numbers. This is achieved by five steps namely: sampling, quantization, encoding, transmission and decoding. [1][2]

The sampling process is the selection of a set of values at specific intervals to produce a series of discrete samples. The sampling rate is then chosen to be high enough to capture the essential information in the signal. [6]

Quantization is the process of approximating the amplitude of a sampled signal to a finite number of levels. The number of quantization levels determines the resolution of the PCM system. The greater the number of quantization levels, the higher the resolution. [4]

Then the quantized samples are encoded into a digital format for transmission or storage. In PCM, the most used encoding scheme is binary, where each sample is represented as a sequence of binary digits or bits. [5]

The digital signal is then transmitted over a communication channel or stored in digital memory.

The stored signal is decoded back to the original analogue signal by the receiver or playback device. The decoding process involves converting the binary values back into their corresponding quantized amplitudes and then reconstructing the continuous analogue signal by connecting the discrete amplitudes with a smooth curve through a process called interpolation.[6]

The objective of this project is to utilize the principles of sampling theory in the context of digital communication by demonstrating the process of signal digitization and reconstruction using MATLAB. It involves generating an analogue signal composed of two sinusoidal components with frequencies of 8Hz and 11Hz and has a finite duration of 2.2 seconds. The analogue signal is then sampled using a proper sampling frequency to generate a sampled signal. The sampled signal is then quantized using uniform quantization with 16 quantization levels.

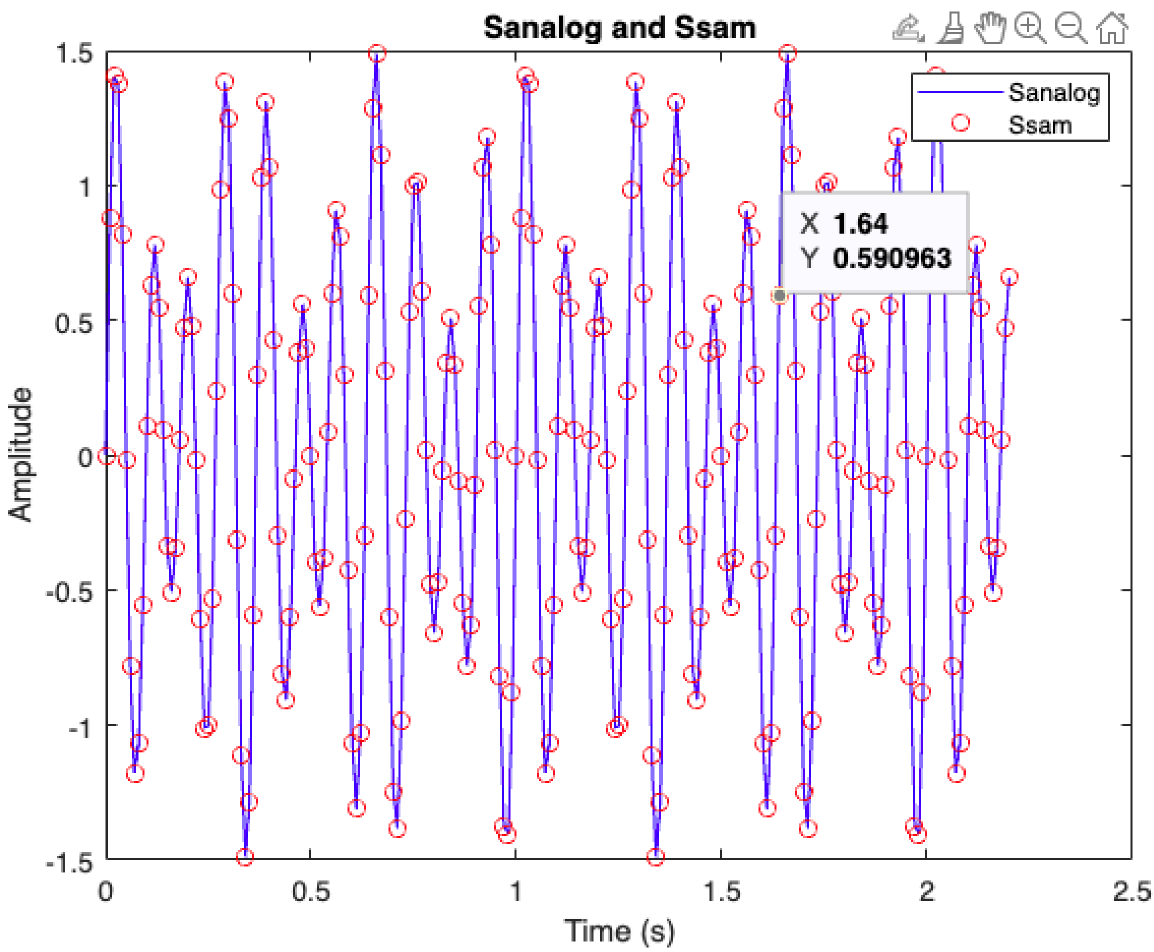
The spectrum of the analogue and sampled signals is then plotted in the frequency domain. Finally, a low pass filter is used to remove any high-frequency components and get the original signal in the frequency domain, which is then transformed into the time domain to produce the reconstructed signal.[6]

Methods and Results

2.1 Sampling Theorem:

In this section, an analogue signal called Sanalog is generated by combining two sinusoidal signals using the built-in “sin” function with frequencies of 8Hz and 11Hz, respectively. A time vector is defined with a sampling frequency of 100Hz which determines the frequency range that can be represented in the digital signal and a finite duration of 2.2 seconds. The sampling theorem to determine the sampling frequency to generate the sampling signal Ssam is determined by the Nyquist-Shannon theorem which states that if a continuous-time signal has a maximum frequency of W Hz, it can be reconstructed perfectly from samples taken at intervals of 1/2W seconds or a rate of 2W samples per second. [6]. The analogue signal is then sampled using the defined time vector to generate a discrete-time signal. Both signals are plotted on the same graph in the time domain in Figure 1, and the analogue signal is the blue continuous wave, and the sampled signal is the red set of discrete points. [7][3]

Figure 1 showing the sampled signal plotted with the original analogue signal



2.2 Quantization:

Quantization involves mapping the continuous amplitude levels of an analogue signal to a finite set of discrete amplitude levels. This process is used in digital signal processing to convert analogue signals into digital signals that can be processed by a computer. [8]

The uniform quantization method divides the range of the signal into equal intervals, with each interval representing a quantization level. The number of quantization levels used is determined by the bit depth of the system, which is specified as 16 in this case.

To ensure that the lowest and highest quantization levels are exactly mapping the lowest and highest sampled signal values, the range of the sampled signal needs to be determined. The range is found by subtracting the minimum value of the sampled signal from the maximum value of the sampled signal. [8]

Once the range of the signal has been determined, it is divided into 16 equal intervals. The size of each interval is calculated by dividing the range of the signal by the number of quantization levels. An array of quantization levels is created from min_val to max_val with a spacing of quant_step between each level.

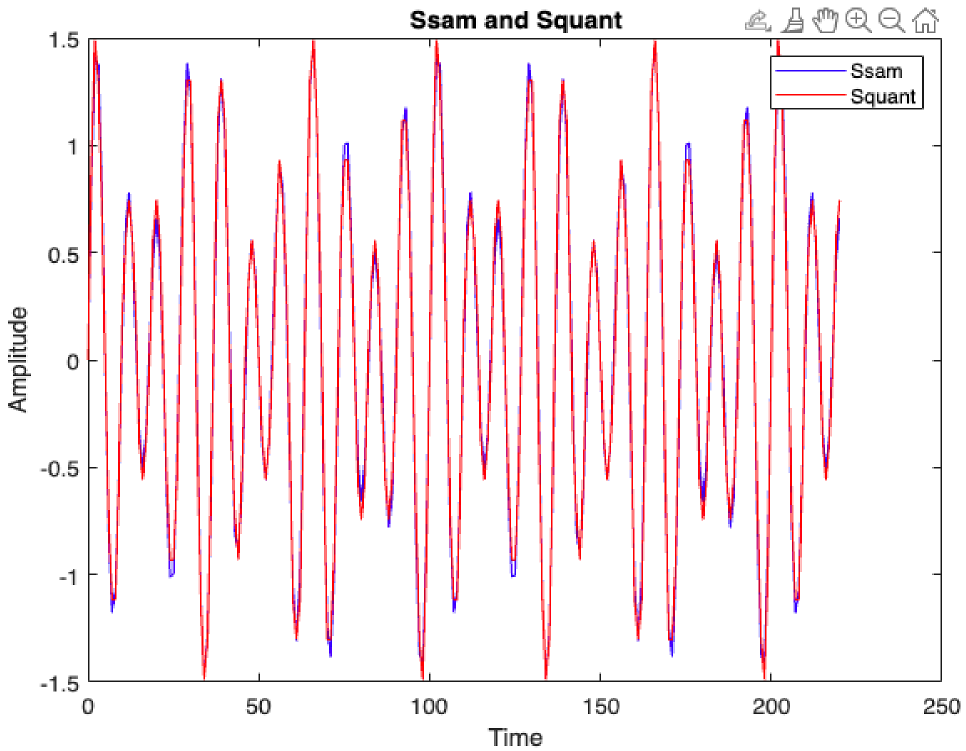
In the code, a zero matrix Squant is created with the same size as Ssam using the zeros() function. It then loops through each element of Ssam using a for loop to find the index of the nearest quantization level to the current sample value using the min() function and abs() function. The min() function returns the smallest element in an array, and the abs() function computes the absolute difference between the quantization levels and the current sample value and it assigns the nearest quantization level to the corresponding element in Squant. [9]

After the loop completes, the quantized signal is generated, where each element is replaced by its nearest quantization level. This process is used in quantization, where a continuous signal is approximated by a finite set of discrete values, resulting in a loss of information. [4]

After the quantized signal Squant has been generated using the uniform quantization method, I plot it in the time domain on a new graph. The time domain plot will show the amplitude of the quantized signal at different time intervals.

To calculate the signal-to-quantization noise ratio (SQNR)which is the quality of the quantized signal, the power of the original signal and the power of the quantization error (or noise power) needs to be determined. The power of the original signal can be calculated by taking the mean of the squared amplitude values of the sampled signal. The power of the quantization error can be calculated by taking the mean of the squared difference between the original sampled signal and the quantized signal. The SQNR can be calculated by dividing the power of the original signal by the power of the quantization error(noise) multiplied by 10*log10.[4] The calculated SQNR result is 22.99 DB.

Figure 2 showing the result of the quantized signal plotted with the original analogue signal



2.3 Spectrum:

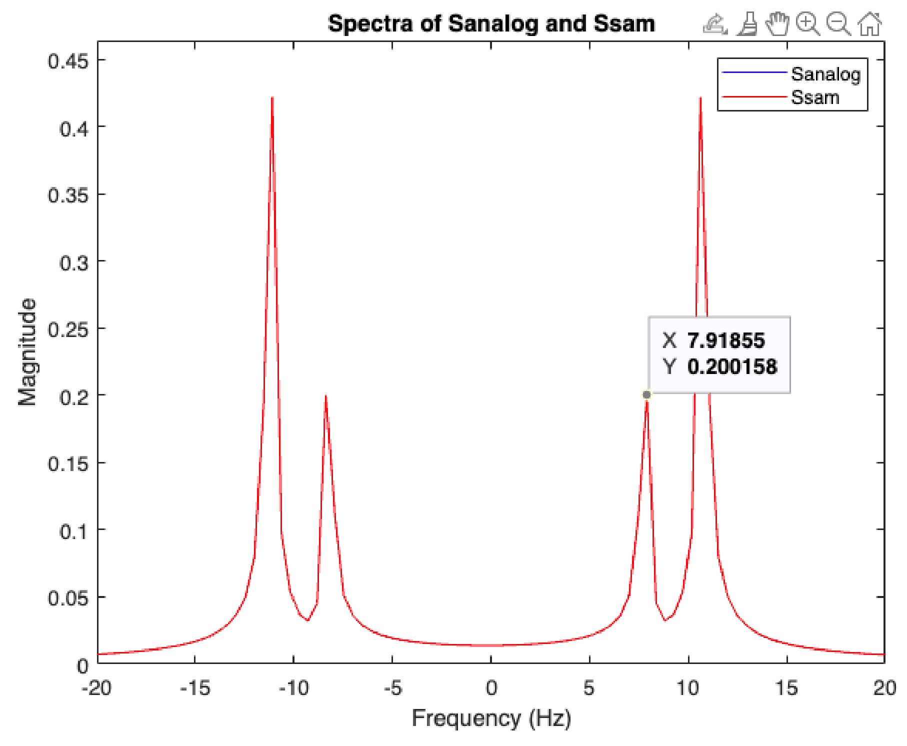
This section required the observation of the original analogue signal, Sanalog, and the sampled signal, Ssam, in the frequency domain. This is done by taking the Fourier transform of both signals using the Fast Fourier Transform (FFT) algorithm because it is a fast and efficient algorithm for computing the Discrete Fourier Transform (DFT) of a discrete-time signal. The goal is to plot the spectrum of both signals on a single graph and ensure that the negative spectrum is displayed as well.

First, it calculates the length of the sampled signal Ssam using the length () function and assigns it to the variable N. Then it generates a frequency vector using the sampling frequency and the length of the sampled signal.

Then, it calculates the spectra of the analogue and sampled signals using the Fast Fourier Transform (FFT) function and normalizes the result by dividing by N. The fftshift() function is used to shift the zero frequency component to the centre of the spectrum.

The plot result of this is shown in Figure 3 below with magnitude on the y-axis and frequency on the x-axis.

Figure 3 showing the spectrum plot of the analogue signal and sampled signal(a)



2.4 Signal Reconstruction:

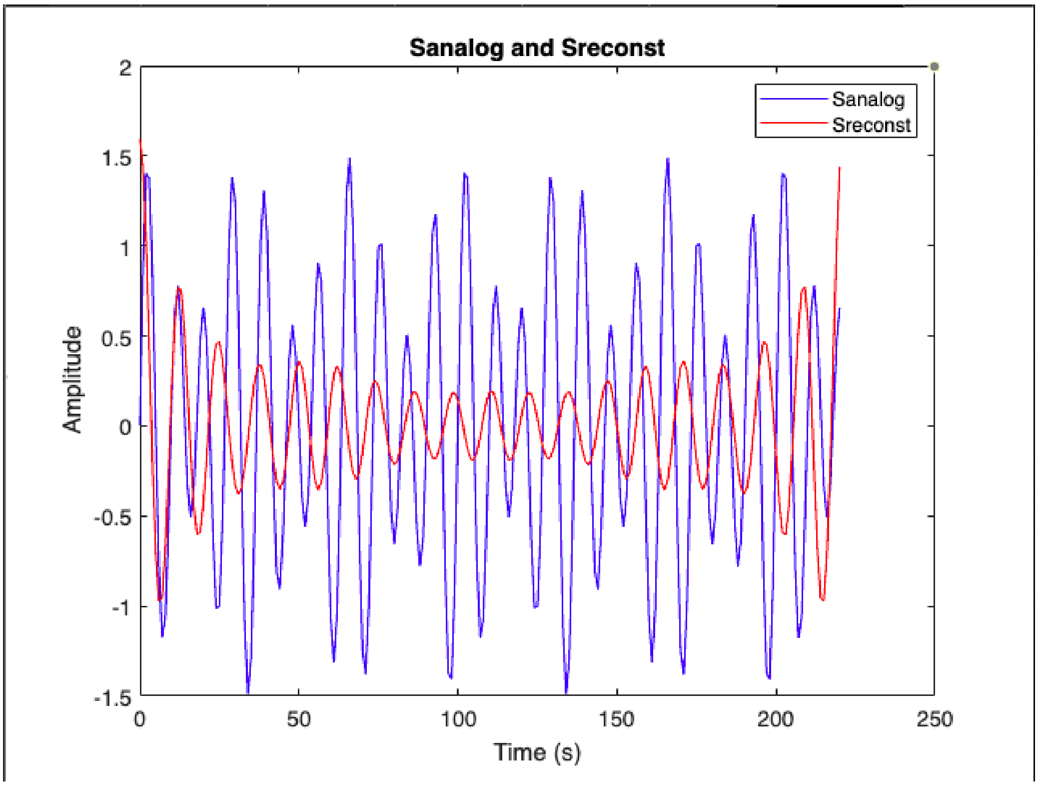
This part gets the original analogue signal from the sampled signal using a low pass filter. This is done by observing the spectrum of the analogue signal, Sanalog, and identifying its maximum frequency component. The maximum frequency component represents the cutoff frequency of the ideal low pass filter.

After identifying the cutoff frequency, I designed an ideal low pass filter in the frequency domain. This filter will pass all frequencies below the cutoff frequency and suppress all frequencies above it. This filter is applied to the FFT of the analogue signal, Sanalog, to extract the frequency domain representation of the original signal. [6]

The filtered spectrum Sanalog_spectrum_filtered is then used to reconstruct the filtered signal in time domain. This is done by taking the inverse Fourier transform of the filtered spectrum, which is achieved with ifft(ifftshift(Sanalog_spectrum_filtered*N)). The ifftshift function is then used to centre the zero frequency component of the spectrum, which is necessary for proper inverse Fourier transformation. N is the length of the signal and is used to normalise the inverse Fourier transform. The resulting signal will be the reconstructed signal, Sreconst. [10]

The plot in Figure 4 shows the original analogue signal, Sanalog, in blue and the reconstructed signal, Sreconst, in red on the same graph in time domain.

Figure 4 showing the analogue signal and reconstructed signal



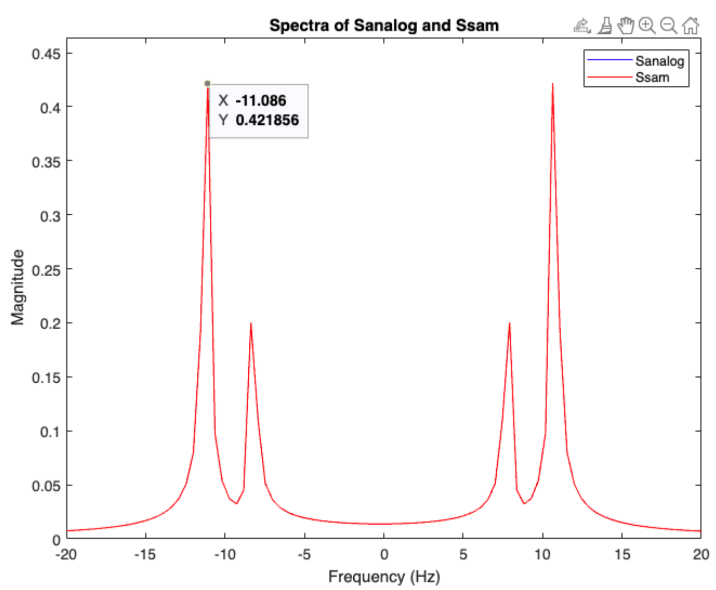
Discussion of Results

Section 2.1 Figure 1, the analogue signal is the blue wave and the sampled signal is shown as the red points on the graph. In the time domain, the analogue signal appears continuous and the sampled signal is a set of discrete points, with a frequency range determined by the sampling frequency 100Hz. As seen in the graph the signal has a finite duration of 2.2 seconds.

Section 2.2 Figure 2, where the quantized signal is plotted against the original analogue signal. With the amplitude on the y-axis and time on the x-axis, the quantization levels are represented by the red Squant lines with the highest and lowest quantization levels corresponding to the highest and lowest values of the original signal. The graph shown has a lower quality than the original sampled signal as the quantized signal has a reduced dynamic range and some level of quantization noise. It has shown the quantized signal as a series of discrete steps with a limited number determined by the bit depth which is specified as 16 rather than the continuous waveform of the original signal. The more levels used in the quantization, the closer the quantized signal will be to the original signal and the SQNR is lower with more levels. However, using more levels will also result in a larger file size and more complex processing requirements. The SQNR result is 22.99dB and indicates the quality of the quantized signal.

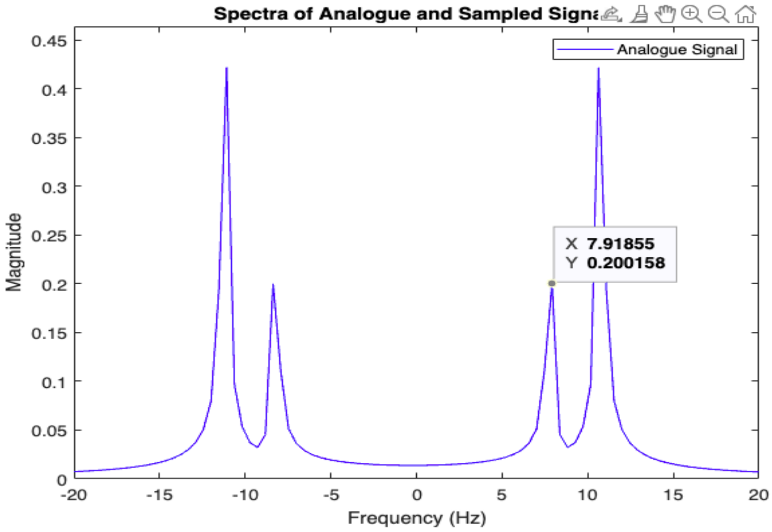
Section 2.3 Figure 3 shows the frequency spectrum of the original analogue signal and sampled signal in the frequency domain. As seen in Figure 3, the signal peaks at about 8Hz and as seen in Figure 5 below it also peaks at 11Hz which corresponds to the frequency components of the signal.

Figure 5 showing the spectrum plot of the analogue signal and sampled signal(b)



The graphs display the negative spectrum as well and I shifted the zero frequency to be exactly in the centre of the graph.
In Figure 3 the blue spectra is not visible hence in commenting the red (sampled signal) graph out, the analogue signal showed and it turns out that the analogue signal spectra is the same as the sampled signal spectra seen in Figure 6.

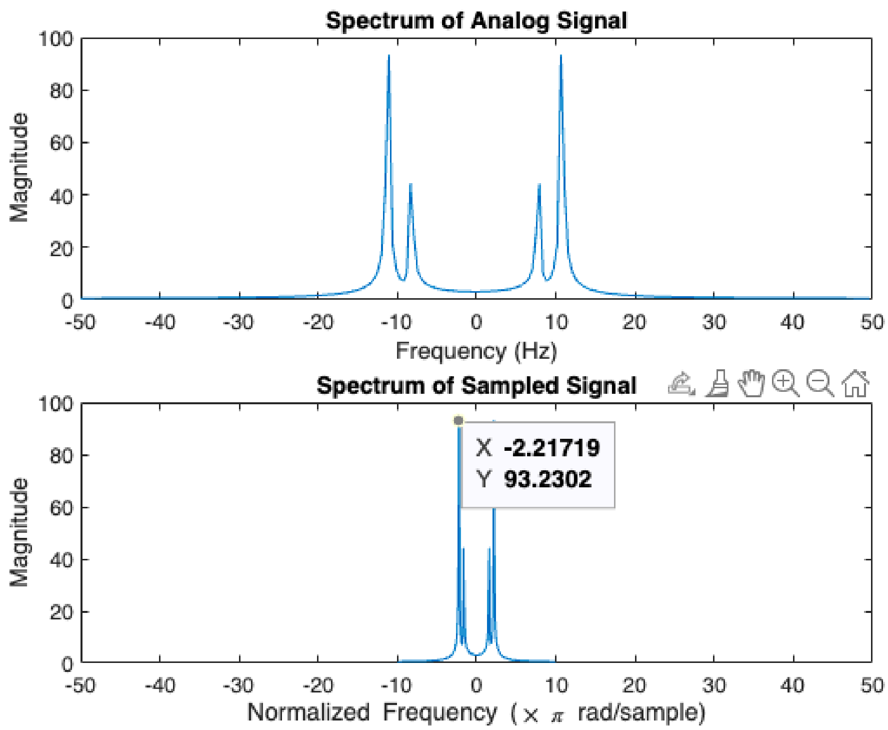
Figure 6 showing the analogue spectra alone



This seems to be because of under sampling [6] which results in spectral replicas of the original signal in the sampled signal. This is due to the process of sampling itself, where the continuous-time signal is multiplied by a train of impulses to create a discrete-time signal. This multiplication in the time domain corresponds to convolution in the frequency domain, resulting in periodic copies of the frequency spectrum centred at multiples of the sampling frequency.

I added an extra code to redo 2.3 and got Figure 7 below. The additional code created a sampling frequency of the sampled signal ‘fsam = Fs/10;’ and then divided by 2 to get the Nyquist Frequency of the sampled signal ‘fsam_nyq = fsam/2;’ Then created a normalised frequency vector for this sampled signal ‘fsam_norm = f/fsam_nyq;’. In this, the spectrum of the sampled signal does not have the same x, y figures as the spectrum of the analog signal.

Figure 7 showing Spectrum plot of analogue and sampled signal



Section 2.4 Figure 4, shows the reconstructed analogue signal and is represented by a continuous wave that looks like the original analogue signal. The ideal low pass filter removed all frequency components above the cutoff frequency as determined by the Nyquist-Shannon theorem.[6] The reconstructed signal has fewer high-frequency components and a smoother curve than the original sampled signal. As seen, the low-pass filter successfully removed any high-frequency components from the signal.

Conclusion

Section 2.1 discusses the process of sampling, and the application of the Nyquist-Shannon theorem to determine the sampling frequency. The generated analogue signal was composed of two sinusoidal components with frequencies of 8Hz and 11Hz and had a duration of 2.2 seconds. A time vector was defined with a sampling frequency of 100Hz, and the analogue signal was sampled using the defined time vector to generate a discrete-time signal. The analogue and sampled signals were then plotted on the same graph in the time domain. The graph showed that the analogue signal was a continuous wave, while the sampled signal was a set of discrete points. This section demonstrates that the analogue signal can be accurately reconstructed from the sampled signal if the sampling frequency is chosen appropriately based on the Nyquist-Shannon theorem.

Section 2.2 discusses the process of quantization, which involves mapping the continuous amplitude levels of an analogue signal to a finite set of discrete amplitude levels. The uniform quantization method was used in this section, which divides the range of the signal into equal intervals, with each interval representing a quantization level. The number of quantization levels used was determined by the bit depth of the system, which was specified as 16 in this case. The quantization process results in a loss of information since a continuous signal is approximated by a finite set of discrete values. The quantized signal was then plotted in the time domain on a new graph. The graph showed the amplitude of the quantized signal at different time intervals. The signal-to-quantization noise ratio (SQNR) was calculated using the formula SQNR = 10* log10 (signal_power/noise_power) and the SQNR was calculated to be 22.99 dB.

Section 2.3 I observed the spectra of Sanalog and Ssam in the frequency domain using the FFT algorithm and plotted them on the same graph, ensuring that the negative frequencies were also displayed. I shifted the zero frequency to the centre of the graph by using the fftshift function. Then observed that Ssam has spectral replicas of the original signal Sanalog, caused by undersampling.

Section 2.4 shows the process of reconstructing the continuous analogue signal by using an ideal low pass filter to remove any high-frequency components and extract the original signal. This was in the frequency domain which was then transformed into the time domain to produce the reconstructed signal. The reconstructed signal was then plotted in the time domain on a new graph, which showed that the reconstructed signal closely matched the original analogue signal. This section demonstrates that the PCM system can accurately reproduce the original signal from the quantized and encoded samples.

References

[1] R. O. N. A. L. D. E. CROCHIERE and L. A. W. R. E. N. C. E. R. RABINER, "Multirate Digital Signal Processing - [PDF document]," *fdocuments.us*. [Online]. Available: https://fdocuments.us/document/multirate-digital-signal-processing-586df500d773d.html?page=3. [Accessed: 15-Mar-2023].

[2]"Pulse-code modulation," Wikipedia, 07-Mar-2023. [Online]. Available: https://en.wikipedia.org/wiki/Pulse-code_modulation. [Accessed: 15-Mar-2023].

[3] A. V. Oppenheim and R. W. Schafer, "Discrete-Time Signal Processing."(2010) Page 140

[4] Korposh, D.S. (2023) "Seminar 2 - Part 2- Digital Encoding-quantisation ." Nottingham.

[5] Admin, "Pulse code modulation - modulation, types, advantages and disadvantages, applications," *BYJUS*, 16-May-2022. [Online]. Available: https://byjus.com/physics/pulse-code-modulation/. [Accessed: 15-Mar-2023].

[6] Korposh, D.S. (2023) "Seminar 2 - Part 1- Digital Encoding-sampling theorem ." Nottingham.

[7] Page 3 Lyons, R. G. (2010). Understanding digital signal processing. Pearson Education.

[8] N. Shlezinger, "Deep Task-Based Quantization ," *Entropy|Free Full-Text|Deep Task-Based Quantization* , 13-Jan-2021. [Online]. Available: https://www.mdpi.com/1099-4300/23/1/104. [Accessed: 15-Mar-2023].

[9] page 6 Y. Wang, "EE3414 Multimedia Communication Systems-1 Quantization ," *quantization.pdf*. [Online]. Available: https://eeweb.engineering.nyu.edu/~yao/EE3414/quantization.pdf. [Accessed: 16-Mar-2023].

[10] "Inverse zero-frequency shift-MATLAB ifftshift-Mathworks," *Inverse zero-frequency shift - MATLAB ifftshift - MathWorks United Kingdom*. [Online]. Available: https://uk.mathworks.com/help/matlab/ref/ifftshift.html. [Accessed: 16-Mar-2023].