Laboratory 3: Digital Transmission Techniques

ECSE308 - Introduction to Communication Systems and Networks

Group C9

Ismail Faruk (260663521)

Marine Huynh (260743073)

I. INTRODUCTION

There exists different techniques to convert data to a signal. Four combinations are possible. Passing from digital to digital, analog to analog, analog to digital and lastly digital to analog. This lab will take a closer look on how techniques to pass from digital data to an analog signal. We will firstly, observe the conversion from analog to digital and vice versa as well as its impact on the signal to noise ratio (SNR) and the bit error rate (BER) that is carried when converting. Then the different techniques of digital modulation will be compared. Lastly the concept of M - Quadrature amplitude modulation will be observe to see its impact on a signal. We will conclude on weather or not this technique is efficient. This lab will be done by using a tool provided by MatLab called Simulink that permits to simulate systems by generating signals specified by their parameters.

II. ANALYSIS

i. Baseband Digital Transmission

Part 1 of the lab is to understand how conversion between digital to analog or analog to digital actually works. Our team constructed a analog to digital converter system (ADC system). *Figure 1* show the system built using Simulink.

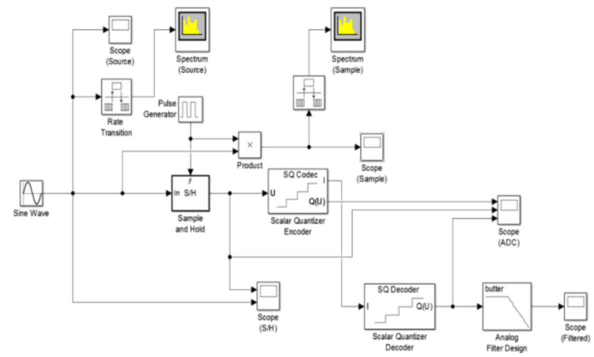

Figure 1: Analog to Digital Converter (ADC) System [1]

After constructing the system, we simulated a sinusoidal input signal and observe the scope from the sample and hold (S/H) and the output signal from the analog to digital converter (ADC) scope, as seen respectively in *Figure 2* and *3*.

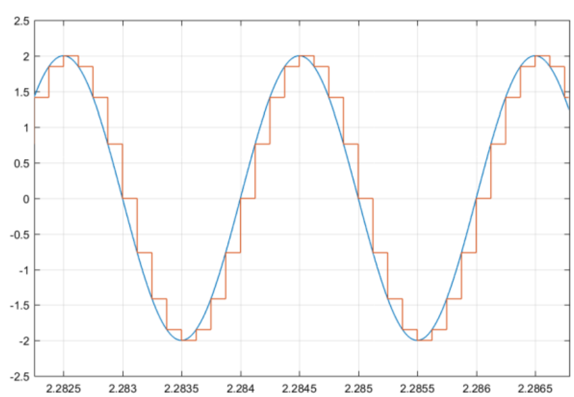

Figure 2: Scope of the sample and hold superimposed with the input sinusoidal signal.

From *Figure 2*, we can see clearly that the orange signal has steps in it as it looks like stairs. This is due to the fact that it is sampling points from the input signal every 0.000125s. This is how the scalar quantizer encoder converts from an analog signal to a digital signal.

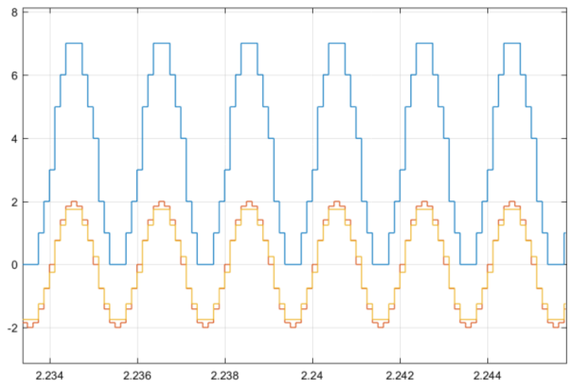

Figure 3: Quantizer encoder, sample and hold, and quantizer decoder scopes.

By looking at *Figure 3*, we can see that the blues line, which is from the encoder, takes the decoder one and changes its range. The decoder will start at 0 and at each level, will increment by an integer of one. This permits to see clearly the amount of levels, but also so that we do not have negative values.

We then compared the spectrums of the source and and spectrum sample. This can be seen in *Figure 4* and *5* respectively.

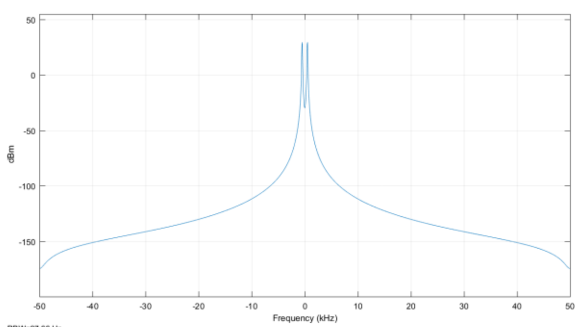

Figure 4: Spectrum of the signal.

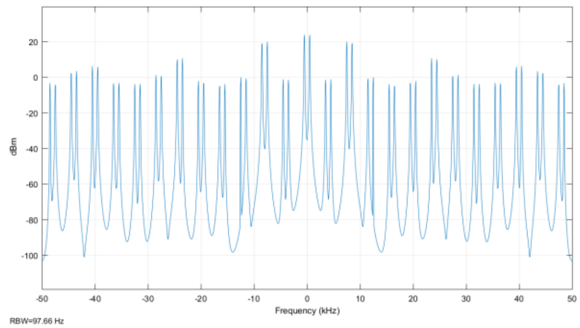

Figure 5: Spectrum of the samples.

Figure 4 shows the spectrum of the input signal. We can then observe the bandwidth that is clearly defined on *Figure 4*. *Figure 5* shows more bandwidths along the sine wave. This repetition comes from the fact that every time the pulse function is high, it will sample, and display a bandwidth at that point.

From *Figure 3*, we see the output from the quantizer encoder and decoder, in blue and yellow respectively. *Figure 3* has parameters for the boundary points of the encoder set to -2:0.5:2 and codebook to -1.75:0.5:1.75. We can observe the number of quantization levels by counting the number of plateaux for one period. Here the number of plateaux is 7. From the number of levels, we can find the number of quantization bits utilized. *Equation 1* below show how it can be calculated.

$$\text{Nb bits} = \log_2(L)$$

eq 1.

Where L is the number of quantization levels. Thus, knowing that the number of levels is 7, and therefore the number of bits calculated is approximately 2.8 bits.

Then we changed the parameters of the encoder to -2:2 for the boundary points and -1.5:1:1.5 for the codebook values. Then we also changed the parameters of the decoder to -1.5:1:1.5 for the codebook values. The output of scope with the different parameters set can be seen in *Figure 6.*

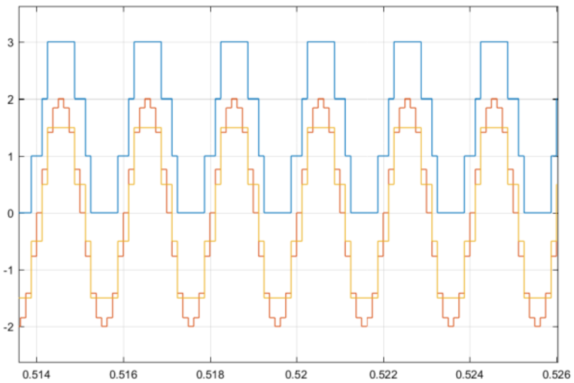

Figure 6: Encoder and Decoder scope for different values of boundary point and codebook.

We can see that compared to *Figure 3*, the encoder signal in *Figure 6* has less quantization levels. This is due to the codebook values set on the encoder. The encoder has now points from -1.5 to 1.5, which gives a range of 3. Therefore, the quantization level will be 3 plus 1, which is the ground level on the x-axis. We can confirm this by looking at the blue line representing the encoding signal in *Figure 6* and counting the number of quantization levels. On the other hand, the scope for the sampling and the decoder have not changed when the boundary points and the codebook values are changing. By lowering the number of levels, the signal will use less bits, and therefore this will be more performante.

We then constructed another system for digital to analog converter as seen in *Figure 7.*

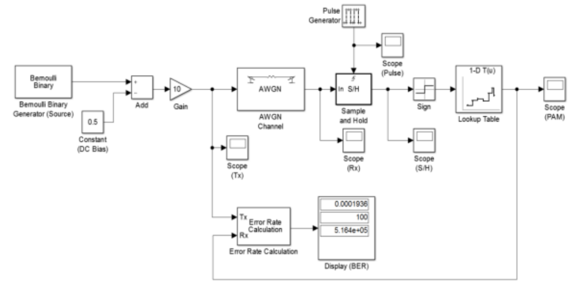

Figure 7: Digital to Analog Converter (DAC) System [1]

The system uses a Bernoulli generator to randomly create a digital signal that contains 0s and 1s. We then increase its gain and constant to have a signal that has a range from -5 to 5 instead of 0 to 1. This will permit to distinguish better the signal once we demodulate it as 1s are represented by positive values and 0s by negatives values. Thus even though we will add white noise to the signal, the initial signal will be retrieved. From *Figure 8* below, we can see the relationship of the bit error rate (BER) vs the amount of carrier-to-noise ratio in dB.

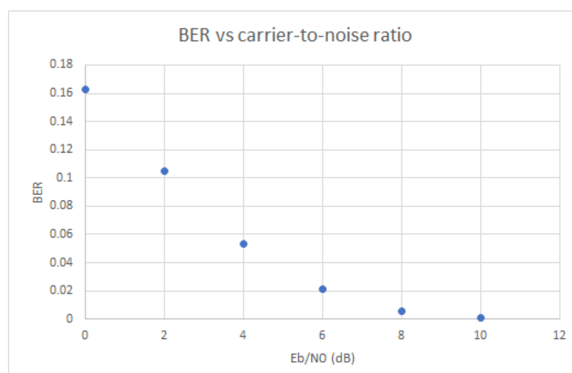

Figure 8: Bit error rate (BER) vs the carrier-to-noise ratio (Eb/N0).

From the graph in *Figure 8*, we can see that the BER decreases rapidly when the Eb/N0 is small and as Eb/N0 tend to infinity, the value of BER reaches the value of zero. This is due to the fact that as you increase Eb/N0, it will take more time to find an error from the samples that you gave. If you increase too high the value of Eb/N0, let's say infinity, the BER will be very small, almost insignificant. The simulation will continue until to collects some errors, but since the BER is too small it will continue and this will crash the software that we are using. The relationship between the BER and the Eb/N0 can be expressed as in *Equation 2*.

$$\mathrm{BER} = \frac{1}{2}\,\mathrm{erfc}\left(\sqrt{E_b/N_0}\right) \quad \text{eq2}$$

ii. Basic Digital Modulation Schemes

The next part of the lab looks at different digital modulation techniques. It will look respectively at binary Amplitude Shift Keying(ASK), Phase Shift Keying (PSK), Frequency Shift Keying(FSK) and 4-Quadrature Amplitude Modulation (4-QAM) [1].

Binary ASK: In ASK, for binary value of 0 and 1, we assigned different amplitude levels, in this case 0 and 1 respectively to the transmitted signal. Thus we only get the transmitted signal for binary values of 1 and zero signal value for binary values of 0, as shown in *Figure 9*.

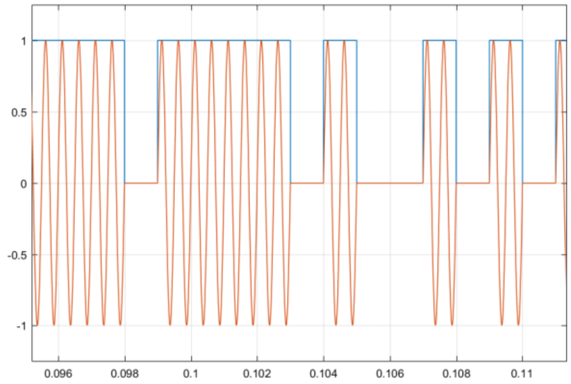

Figure 9: BASK

Binary PSK: In PSK, for binary value of 0 and 1, we shifted the phase of the transmitted signal by 0 rad and π rad respectively. Thus we can observe a shift in phase when the binary value shifts from 0 to 1 and 1 to 0, due to the phase shifting of the transmitted signal, as shown in Figure 10.

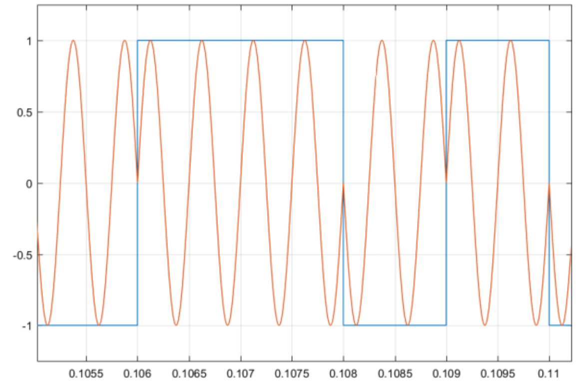

Figure 10: BPSK

Binary FSK: In FSK, we shifted the carrier frequency for binary value of 0 and 1, by adding frequency spacing and subtracting frequency spacing, respectively. Thus we see a higher frequency in the transmitted wave for binary value of 0 and lower frequency of that for binary value of 1, as shown in *Figure 11*.

The signal frequency for binary value of 0 was 3500 Hz. The signal frequency for binary value of 1 was 2500 Hz. Thus, the carrier frequency is 3000 Hz and the frequency separation was 500 Hz.

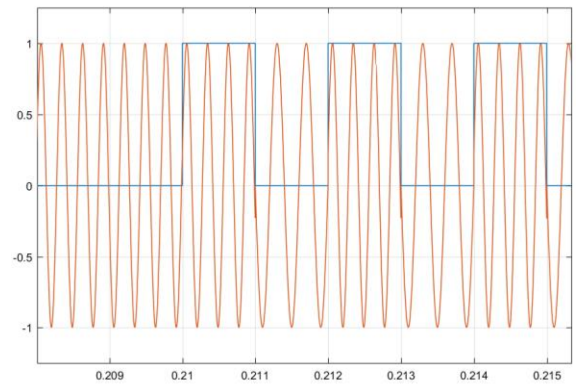

Figure 11: BFSK

4-QAM can be obtained by grouping two corresponding binary bits together and assigning a corresponding phase shift for each combination of bits in the group.

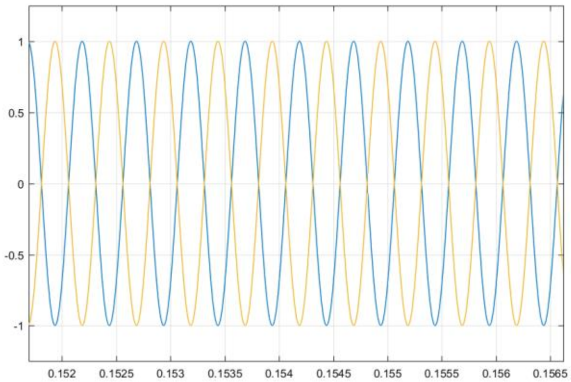

Figure 12: 4-QAM

To implement 4-QAM, we would need to transmited the bits by the separating every corresponding bits, using a serial to parallel converter. One bit is modulated by the carrier frequency and the corresponding bit is modulated by the carrier frequency, phase shifted by 90 degrees. Then the total sum of this modulated signal would become a 4-QAM. The carrier signal is also amplitude modulated.

To implement 4-QAM from BPSK, we would add onto it a change in amplitude to get a QAM.

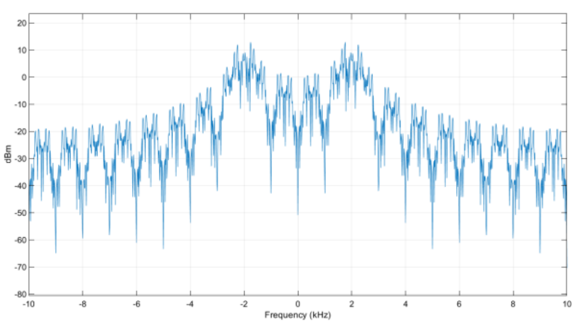

Figure 13: BPSK Spectrum

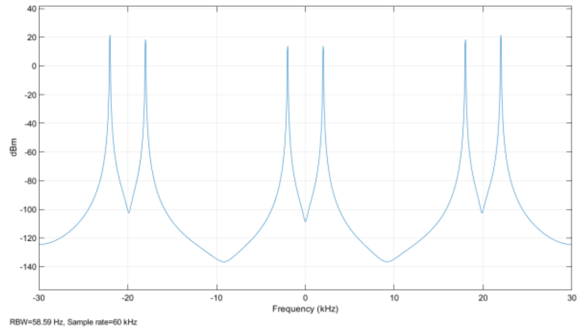

Figure 14: 4-QAM Spectrum

Power spectrum of 4-QAM is related to that of binary PSK, because frequency difference between the peaks of the 4-QAM spectrum is the equal to the bandwidth of BPSK.

Transmission bandwidth of BASK was 3.891 kHz, with Channel Power 20.784 dBm and Frequency error -2.746 kHz.

Transmission bandwidth of BPSK was 5.011 kHz, with Channel Power 23.979 dBm and Frequency Error -2.227 kHz.

Transmission bandwidth of BFSK was 2.255 kHz, with Channel Power 23.979 dBm and Frequency Error = -2.003 kHz.

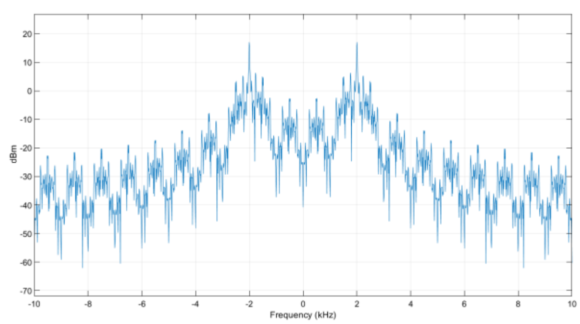

Figure 15: BASK Spectrum

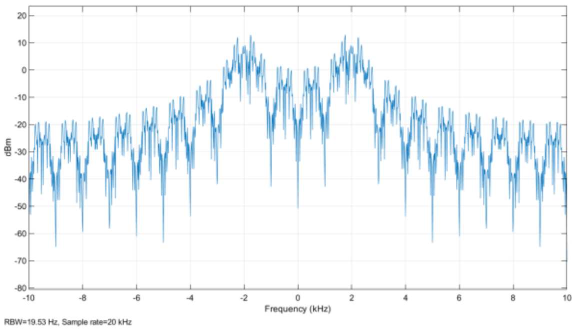

Figure 16: BFSK Spectrum

PSK is the most bandwidth efficient, as it is more error free than ASK and has a higher data rate over the same bandwidth when compared to FSK.

iii. M-QAM Modulation and Demodulation

We then observed in a deeper way the M-Quadruple Amplitude Modulation (QAM). In part ii, we looked briefly at a 4-QAM. Now let's look at a 16-QAM that can carry 4 bits per symbol. In *Figure 17*, the bandwidth for one sample can be seen. The spectrum has been zoomed in for better clarity.

From the simulink software, the bandwidth was estimated to be 22.5692dB. *Figure 17*.

By superposition of ASK with four different amplitude levels and PSK with four different phase levels, we can produce 16-QAM. Based on the value of the random integer, it is assigned an amplitude and a phase that will map it onto the constellation diagram.

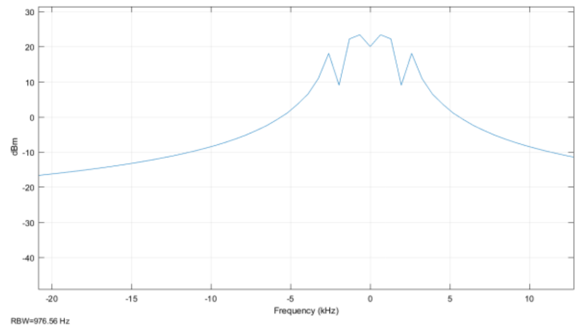

Figure 17: Bandwidth spectrum.

In order to see the influence of the white noise on the constellation diagram, we have given different values of Es/N0 and observed the outcome. We concluded that when the value of Es/N0 is small the sample points are close to their associated point. Increase in SNR reduces the distance between the occuring symbols from their ideal positions. Therefore, as SNR increases, the white noise decreases and the BER also decreases.

In order to have a better understand on the impact of passing more bits/symbol, we observed 16-QAM, 64-QAM and 256-QAM, that has respectively 4 bits/symbol, 6 bits/symbol and 8 bits/symbol. The relationship between the BER and the Es/N0 were plotted in *Figure 18* for all 3.

From *Figure 18*, we can see that the higher the number of bits/symbol, the higher the BER for a same Es/N0 value. Since we have more bits, more levels of amplitude and phase will need to be present to distinguish points from the constellation. Therefore, there is a higher potential error rate.

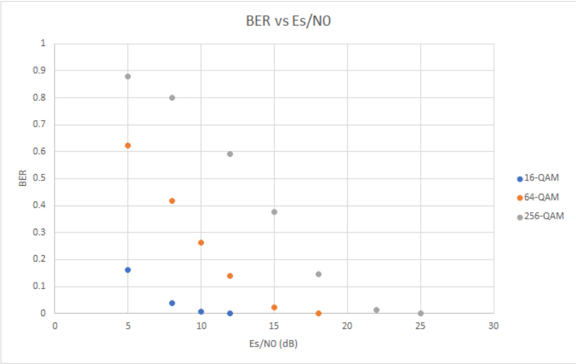

Figure 18: Relationship of BER and Es/N0 for 16, 64 and 256 QAMs.

The BER is greater for higher QAM than lower QAM. With increase in transmission bandwidth, the BER will increase, because transmission signal would include more noise. With lower distance between constellation symbols, the BER would be higher, because symbols would be close to each other and may even overlap due to the noise. Thus, increasing the distance between symbols, would decrease BER.

III. CONCLUSION

After these analysis we can see the different transmission techniques used to pass signals. We also looked at conversion from digital to analog and vice-versa. Based on these experiment the tradeoff between performance, bandwidth and cost can be derived.

IV. REFERENCES

[1] W. Stallings, “Data and Computer Communications,” Pearson Education, Inc. 10th ed. 2014. [pdf], [Accessed Oct. 12, 2018].