COMPARISON OF ZERO FORCING AND MMSE EQUALIZERS IN DIFFERENT AWGN CHANNELS

Harun Durmuş (270206025), Necmettin Yiğit Dübek (260206049), Mustafa Eren Çelebi (270206053)

January 1, 2024

Abstract- This project aims to conduct a comprehensive comparison of the performance of Zero Forcing and MMSE (Minimum Mean Square Error) equalizers in Band Limited AWGN (Additive White Gaussian Noise) channels. Zero Forcing leverages the direct relationships between transmitted and received signals, whereas MMSE optimizes received signals based on the Minimum Mean Square Error principle. The project will employ simulations to investigate performance differences between both equalizer types and compare the results. The primary objective is to determine which equalizer type demonstrates superior performance in a Band Limited AWGN channel.

I. INTRODUCTION

In communication systems, equalization plays a pivotal role in suppressing channel-induced interference and ensuring reliable data transmission. This project delves into the comparison of two common equalization techniques: Zero Forcing and MMSE, and their effect in suppressing intersymbol interference (ISI) in Band Limited AWGN channels.

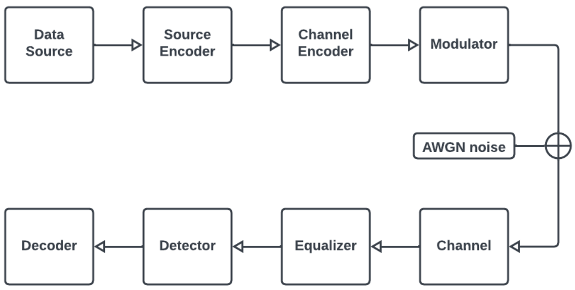

Figure 1: *General block diagram of the process of this project*

A. Equalization Overview

Equalization is a critical component of wireless communication systems that serves to compensate for channel-induced intersymbol interference (ISI). Every channel can be characterized as linear and time-invariant, with specific phase and magnitude responses. These responses influence the quality of signal transmission.

Transversal Filtering

Transversal filtering is a common approach to equalization, involving the use of a filter with adjustable taps. The filter operates on the incoming signal to minimize ISI and noise.

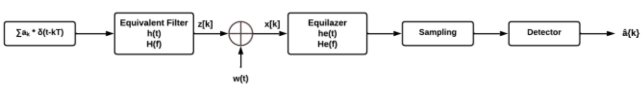

Figure 2: *Detailed Equalization Diagram*

$H(f) = H_t(f).H_c(f).H_r(f)$, where $H_t(f)$ is TX filter, $H_c(f)$ is channel and $H_r(f)$ is Rx filter responses.

Zero Forcing Equalizer

Zero Forcing equalization is achieved by adjusting the filter taps to ensure that the equalizer output is forced to zero at N sample points on each side of the signal. This approach aims to entirely suppress ISI but can result in an increase in noise.

- adjust $\{c_n\}$ (n = -N to N):

so that $z[k] = \begin{cases} 1 & \text{if } k = 0 \\ 0 & \text{if } k = \pm1, \pm2, \dots, \pm N \end{cases}$

Minimum Mean-Square Error (MMSE) Equalizer

The MMSE Equalizer seeks to minimize the mean square error (MSE) of ISI and noise power at the equalizer output. Unlike Zero-Forcing, MMSE Equalization does not entirely eliminate ISI but provides a trade-off by not enhancing noise as much.

- adjust $\{c_n\}$ (n = -N to N), so that $\min(E\{[(z[k] - x[k])^2]\})$

This project represents an opportunity to gain deeper insights into the practical aspects of Zero Forcing and MMSE equalizers in real-world communication scenarios. The results of this study will contribute to our understanding of equalization techniques in communication systems.

II. COMMUNICATION SYSTEM DESIGN

A. Modulation

Basic Concepts of Digital Modulation

Digital modulation[1] is a process used in telecommunications to transfer a digital bit stream over an analog communication channel. This method involves modifying the properties of a carrier wave - such as amplitude, frequency, or phase- to encode digital information. The primary aim of digital modulation is to translate digital data into a form suitable for reliable transmission through a physical medium, typically radio waves, cables, or optical fibers.

Overview of Different Modulation Types: ASK, FSK, and PSK

Digital modulation encompasses various techniques, each with its unique characteristics and applications.

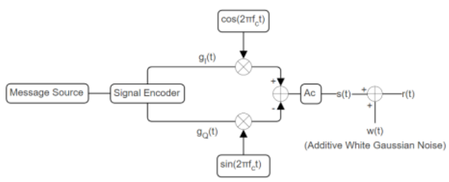

Figure 3: *General Representation of Modulation*

In the figure above, structure of the modulation part in bandpass transmission is shown. Considering binary transmission, three different methods are mentioned by handling an example of binary symbols $s_0(t)$ and $s_1(t)$ below:

- Bit 0, $s_0(t)$, where $0 \leq t \leq T_b$ and also $0 \leq t \leq T_s$ and $T_b = T_s$
- Bit 1, $s_1(t)$

In the cases below $A_c = 1$ and $f_c = 2000$ Hz assumptions are made for obtaining plot figures.

Amplitude Shift Keying (ASK): Amplitude Shift Keying (ASK) is a modulation technique where the amplitude of the carrier wave is varied in direct proportion to the digital data being transmitted. This method encodes binary data: a high amplitude of the carrier wave represents a binary '1', while a low amplitude represents a binary '0'. ASK is straightforward and cost-effective, making it suitable for applications where complexity and cost need to be minimized, such as in RFID systems and infrared remote controls. However, its major drawback is its susceptibility to noise and interference, as amplitude variations can be easily distorted, leading to errors in the received signal. This limitation makes ASK less ideal for noisy environments or long-distance transmission.

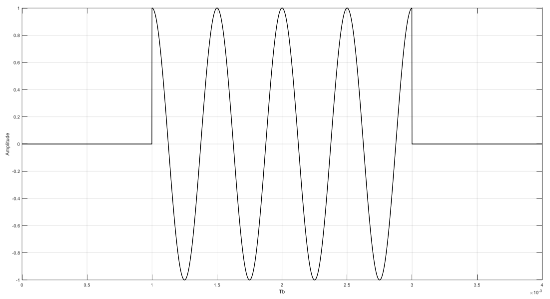

Figure 4: *On-Off Keying with B-ASK[3]*

Orthonormal basis function for the case can be calculated as below and used for signal constellation in further steps:

- $Ø_1(t) = (2/T_b)^{1/2}\cos(2\pi f_c t)$

Frequency Shift Keying (FSK): Frequency Shift Keying (FSK) is a digital modulation technique in which the frequency of the carrier wave is varied to represent the digital data. Unlike ASK, FSK does not alter the amplitude but instead shifts between different frequencies to denote binary '1s' and '0s'. This feature makes FSK more resistant to signal amplitude noise, thus more reliable in noisy environments compared to ASK. FSK is widely used in applications like traditional radio systems, caller ID systems, and modem technology. Its resilience to noise and ability to operate effectively over various transmission media make it a versatile choice. However, it generally requires a larger bandwidth than ASK, which can be a limiting factor in bandwidth-constrained systems.

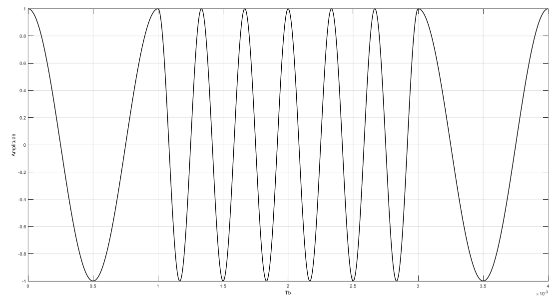

Figure 5*: Signal Representation by using B-FSK[3]*

- $Ø_0(t) = (2/T_b)^{1/2}\cos(2\pi f_0 t)$
- $Ø_1(t) = (2/T_b)^{1/2}\cos(2\pi f_1 t)$

There are two basis functions.

Phase Shift Keying (PSK): Phase Shift Keying (PSK) is a form of digital modulation that involves changing the phase of the carrier wave to convey information. The phase of the carrier signal is shifted in relation to the baseline or unmodulated signal to represent the digital data. PSK is more bandwidth-efficient than both ASK and FSK and offers a higher noise immunity, making it well-suited for high-data-rate applications in wireless communication. The simplest form of PSK is Binary PSK, where only two phase shifts are used, each representing a binary digit. PSK is widely used in technologies like Wi-Fi, RFID, and Bluetooth, where maintaining signal integrity in the presence of noise is crucial. Its main challenge is in the receiver design, which must accurately detect phase changes, a task that becomes more complex in environments with signal reflection and fading.

- $Ø_1(t) = (2/T_b)^{1/2}\cos(2\pi f_c t)$

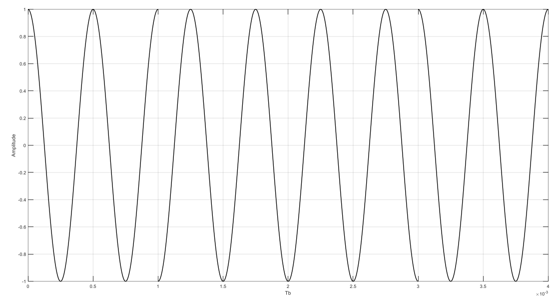

Figure 6: *Signal Representation by using B-PSK[3]*

Obtaining the signal constellation graphs from the calculated orthonormal basis functions for each method, it is observed that B-PSK is the best method since the probability of error is minimum in that case, assuming variance of the noise components are same.

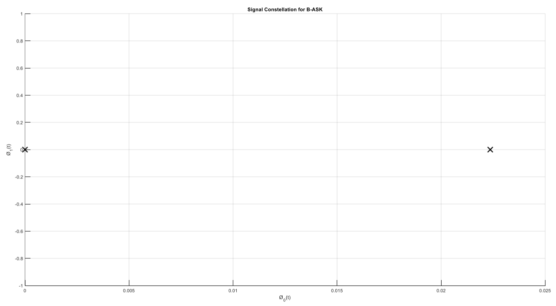

Figure 7: *Signal Constellation for B-ASK[4]*

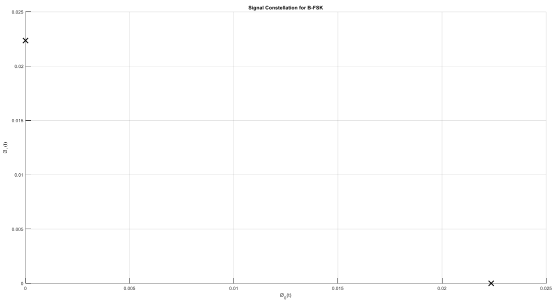

Figure 8: *Signal Constellation for B-FSK[5]*

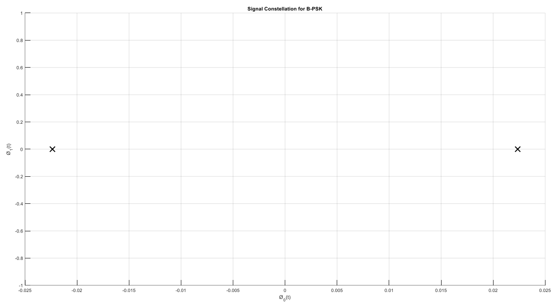

Figure 9: *Signal Constellation for B-PSK[6]*

- $\sqrt{Eb} < \sqrt{(2Eb)} < 2\sqrt{Eb}$

Since the maximum distance of between energies of two bits $s_0(t)$ and $s_1(t)$ in signal constellation is $2\sqrt{Eb}$ which is obtained with B-PSK, the minimum error will be occur. Probability of error can be calculated by using Q functions.

B. Channel

In communication systems[2], a channel serves as the medium for transmitting information from a sender to a receiver. This medium varies depending on the type of communication system. For instance, in wired communication systems like traditional telephone networks or Ethernet connections, the channel consists of physical wires or cables that carry electrical signals. In wireless communication systems such as mobile networks or Wi-Fi, the channel is the air or space through which radio waves or electromagnetic signals travel. Optical communication systems use fiber optic cables, where the channel is the optical fiber through which data is transmitted using light signals. In satellite communication, the channel involves transmitting signals between ground stations and satellites or between different satellites. Lastly, in underwater communication systems like those used for submarine communication, the channel is represented by submerged cables.

The channel is a crucial concept in communication theory, and the characteristics of the channel can have a significant impact on the quality and reliability of the communication. Factors such as noise, interference, and bandwidth limitations can affect how well information is transmitted through the channel. Engineers and researchers study these factors to design communication systems that can effectively overcome challenges and deliver reliable and efficient communication.

Before delving into the other channel types that play a pivotal role in the field of communication systems, it's essential to understand the dynamic nature of signal propagation. In the world of wireless and wired communications, signals don't always travel smoothly from sender to receiver. Instead, they encounter a myriad of challenges along the way, from reflections off buildings in urban settings to the interference caused by multiple signal paths. These challenges give rise to a fascinating array of channel models, each designed to capture the unique characteristics of these real-world scenarios. Let's explore these channel types, from Rayleigh and Rician fading channels to frequency-selective and Nakagami fading channels, shedding light on the intricacies of signal transmission in diverse environments:

Rayleigh Fading Channel: The Rayleigh Fading Channel is characterized by the introduction of random amplitude variations to the transmitted signal, effectively simulating the impact of multipath propagation in wireless communication. A prime example of this occurs in mobile communication within urban environments, where signals can bounce off buildings, resulting in fluctuations in signal strength.

Rician Fading Channel: The Rician fading channel exhibits characteristics which is similar to the Rayleigh fading channel, featuring random multipath fading, but distinguished by the presence of a dominant line-of-sight signal component, making it suitable for modeling scenarios involving a robust direct signal amidst scattered signals. An illustrative example of its applicability lies in wireless communication within open environments with occasional obstructions, where a distinct line of sight to the transmitter coexists with multipath propagation.

Frequency-Selective Fading Channel: The Frequency-Selective Fading Channel exhibits varying levels of attenuation and phase shifts at different frequencies, making it a suitable model for scenarios where the channel response varies with signal frequency. An example of its applicability is in high-frequency radio communication in urban areas characterized by varying building densities, causing different frequency components to experience distinct fading levels.

Nakagami Fading Channel: The Nakagami Fading Channel is characterized by a fading envelope following a Nakagami probability distribution, which can manifest varying fading severity. This channel is particularly relevant in applications like underwater acoustic communication, where the signal experiences different levels of attenuation due to changing water conditions.

Gilbert-Elliott Channel: The Gilbert-Elliott Channel model incorporates both "good" and "bad" states, representing periods of reliable and unreliable transmission, making it a valuable tool for modeling communication in mobile networks. It reflects scenarios where channel conditions may switch between good and bad states due to factors such as shadowing or signal blockage.

Multipath Fading Channel: The Multipath Fading Channel simulates the effects of signals arriving at the receiver through multiple paths, leading to constructive and destructive interference. This channel model is commonly used for indoor wireless communication, where signals may reflect off various surfaces like walls and floors, resulting in multipath propagation.

Slow Fading Channel: The Slow Fading Channel experiences variations in signal strength over relatively extended periods, often attributed to factors like vehicle movement or gradual changes in the environment. This channel model finds application in communication within vehicular networks, where channel conditions change gradually as vehicles move. These channel models collectively aid researchers and engineers in the analysis and design of communication systems that can operate effectively in realistic and challenging conditions.

As a result of examining these channel types, and in addition, as a result of the task given in the project, it is decided to use Additive White Gaussian Noise Channel and observe the response of the system in different signal-to-noise ratios.

AWGN (Additive White Gaussian Noise) Channel

This channel introduces random additive noise to the transmitted signal, simulating the effects of various environmental interferences in communication systems. For instance, in digital communication systems, the AWGN channel can represent the cumulative impact of thermal noise, electronic interference, and other unpredictable factors. For instance, in wireless communication, AWGN may model the collective impact of atmospheric noise, electronic device interference, and other random disturbances that affect the clarity of the received signal. Here are the key effects of the AWGN channel on the received signal:

Additive Noise: The AWGN channel adds random noise to the transmitted signal during its journey from the sender to the receiver. This noise is independent of the original signal and is considered to be unrelated to the communication process.

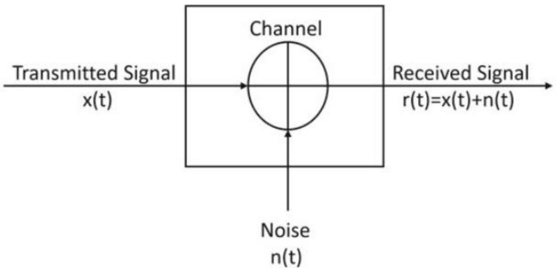

Figure 10*: Noise addition part*

Signal Degradation: The noise introduced by the AWGN channel can degrade the quality of the received signal. As the noise is added to the signal, it can make it more challenging for the receiver to accurately detect and interpret the original information.

Impact on Signal-to-Noise Ratio (SNR): The quality of the received signal is often measured by the Signal-to-Noise Ratio (SNR), which is the ratio of the signal power to the noise power. In the presence of AWGN, the SNR decreases, making it more difficult for the receiver to distinguish the signal from the noise.

Bit Errors: The presence of noise in the received signal can lead to errors in the detection of individual bits. In digital communication systems, this may result in bit errors, where the received bits do not match the transmitted bits.

Reduced Communication Performance: The random nature of the Gaussian noise in the AWGN channel means that the received signal experiences fluctuations that can impact communication performance. This is especially critical in scenarios where a high level of accuracy is required, such as in data transmission or digital audio/video communication.

Uncertainty: The randomness of the noise introduces uncertainty into the received signal. This uncertainty poses a challenge for the receiver to reliably reconstruct the original information sent by the transmitter.

Despite these challenges, engineers use various techniques, such as error-correcting codes, modulation schemes, and signal processing algorithms, to mitigate the effects of AWGN and improve the reliability of communication systems in the presence of noise.

C. Equalization

Zero Forcing Equalizer

Zero-Forcing Equalizer refers to a form of linear equalization algorithm used in communication systems which applies the inverse of the frequency response of the channel. The Zero-Forcing Equalizer applies the inverse of the channel frequency response to the received signal, to restore the signal after the channel[7]. The name Zero Forcing corresponds to bringing down the intersymbol interference (ISI) to zero in a noise free case. This will be useful when ISI is significant compared to noise. For a channel with frequency response the zero forcing equalizer is constructed by:

$$C(f) = \frac{1}{F(f)}$$

Figure 11: *Obtaining Equalizer Formula*

Thus the combination of channel and equalizer gives a flat frequency response and linear phase . In reality, zero-forcing equalization does not work in some applications, for the following reasons:
Even though the channel impulse response has finite length, the impulse response of the equalizer needs to be infinitely long - At some frequencies the received signal may be weak. To compensate, the magnitude of the zero-forcing filter ("gain") grows very large. As a consequence, any noise added after the channel gets boosted by a large factor and destroys the overall signal-to-noise ratio. Furthermore, the channel may have zeroes in its frequency response that cannot be inverted at all.

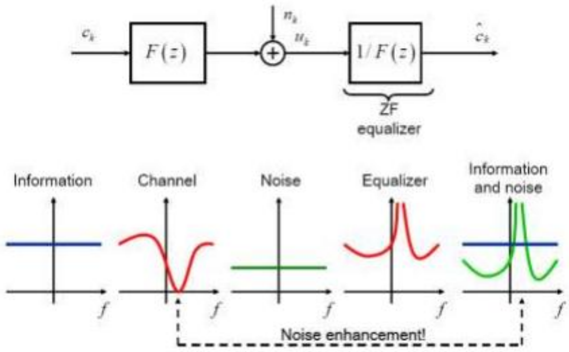

Figure 12: *Zero Forcing Equalizer*[9]

The noise enhancement problem is defined at the begininng, in the equalization overview section.

MMSE Equalizer

To minimize the intersymbol interference and additive noise effects, the equalizer coefficients can be optimized using the minimum mean squared error (MMSE) criterion. When the SNR has elevated values the MMSE equalizer works as Zero-Forcing does, but when the SNR has lower values, the fact that MMSE equalizer takes into account the noise and signal variance, makes to not amplify the noise as Zero-Forcing does.

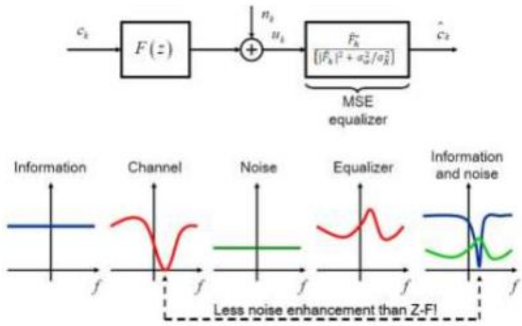

Figure 13: *MMSE Equalizer*[9]

As it can be seen in the figure above, when the Signal to Noise Ratio (SNR) has high values, the MMSE equalizer works as the Zero Forcing does, but for the rest of values that SNR can take, the MMSE equalizer works better in terms of distortion.

That is got for the term $\sigma_w^2/\ \sigma_x^2$, because when a deep null appears in the frequency response of the channel, this equalizer does not amplifies the received signal just multiplying for the inverse of the channel, it takes into account the SNR in order to not amplify so much the noise term.

This is the big difference about the two equalizers and it can be clearly appreciated in the deep nulls of the channel. While Zero Forcing equalizer increases significantly the noise term in order to recover the original signal, MMSE does not do.

D. Demodulation

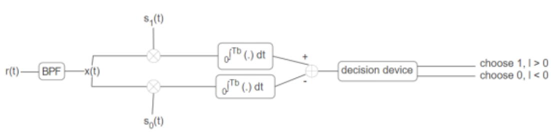

Figure 14: *Demodulation Block Diagram*

Considering two path correlation receiver a general case the demodulation of the B-ASK signal is described above. But for designing optimum receiver, coherent demodulation of the BPSK signal is performed at the receiver. The received signal is multiplied by a carrier with the same frequency and phase as at the transmitter, which is required for coherent demodulation. Phase synchronization is typically achieved using the Phase Locked Loop (PLL) in the receiver. However, perfect phase synchronization is assumed here, and PLL implementation is not undertaken. The block diagram of the BPSK modulator is depicted in the figure above. After multiplication with the carrier (orthonormal basis function), the signal is integrated and sampled over symbol time 'T'. Thresholding is then applied to determine whether a '1' is transmitted a '0' is transmitted.

Coherent receiver of B-PSK has only one branch for correlation since there is only one basis function.

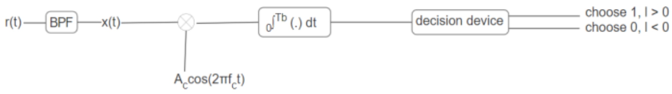

Figure 15: *Coherent Detector of B-PSK*

The demodulation of B-PSK signals involves several key steps. Initially, the demodulator detects the phase of the incoming signal and compares it against a reference phase. This comparison is critical to determine the binary value represented by each phase. Accurate demodulation requires precise synchronization between the transmitter and receiver to ensure that the phase comparison correctly interprets the data. Fluctuations in phase, known as phase noise, or any synchronization mismatches can lead to errors in interpreting the binary data. Despite these challenges, B-PSK remains a preferred choice due to its relative resistance to noise and efficient use of bandwidth compared to other modulation schemes like Amplitude Shift Keying (ASK).

E. Bit Error Rate (BER) Analysis

BER Basics

Such an analysis system is used to determine the efficiency of the system that needs to be created by evaluating each bit error according to time in order to detect all the distortions that occur during the transmission of the data.The difference from many other evaluation systems is that the full end-to-end performance of the data is evaluated.This allows us to measure the actual performance of a fully working system instead of evaluating the component parts one by one.The bit error rate can be easily translated into a simple formula:

- BER = Errors / Total number of bits

If the signal-to-noise ratio between the transmitter and receiver is high, the bit error rate will be small, but if noise is detected in the signal, we need to consider the bit error rate.

Factors Effecting BER

By changing controllable variables, it can be optimized to provide the required level of performance. This is normally accomplished during the design stages of a data transmission system, so that performance parameters can be adjusted during the initial design concept stages.

Interference: A system's interference levels are often determined by outside variables and cannot be altered by the system's architecture. Nonetheless, the system's bandwidth may be adjusted. It is possible to lower the amount of interference by narrowing the bandwidth. Reducing the bandwidth, however, places a limit on the amount of data that can be processed.

Boost transmitter power: Increasing the system's power level can also result in a boost in the power per bit. This needs to be weighed against a number of variables, such as the degree of other users' disturbance, the effect of raising the power output on the size of the power amplifier, total power consumption, battery life, etc.

Reduce bandwidth: Cutting back on bandwidth is another strategy that may be used to lower the bit error rate. The signal to noise ratio will rise as a result of receiving lower noise levels. This again leads to a decrease in the achievable data throughput.
Lower order modulation: Data throughput may be sacrificed in order to employ lower order modulation techniques.

Achieving a bit error rate that is adequate requires balancing all of the relevant elements. Typically, certain compromises must be made in order to meet all of the requirements. Further tradeoffs can be made about the amounts of error correction put into the transmitted data, even at bit error rates that are less than optimal. This can assist hide the consequences of any bit mistakes that occur, so improving the bit error rate overall, even if more redundant data must be provided with greater degrees of error correction.

When deciding which connection characteristics, such as power and modulation type, to utilize, the bit error rate, or BER, parameter is frequently stated for various communications systems.

III. MATLAB DEMONSTRATION

By following the flow diagram that is defined in figure 1, process is demonstrated on MATLAB[8]. First, random bit data which represents the encoded data of an any analog input signal (such as audio) is generated. Then, modulation, which is chosen as B-PSK, is applied. The resultant modulated signal is parametrized as ***"mxsig"***.

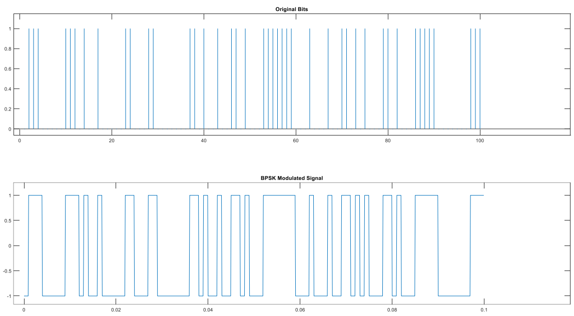

Figure 17: *B-PSK Modulated Signal*

As it is mentioned in the modulation part above in the report, cosinusoidal wave is overlayed on bit data signal in the figure above. Signal constellation is visualized as the figure below. There is no in-phase component since this is B-PSK modulation. There are only two symbols to transmit.

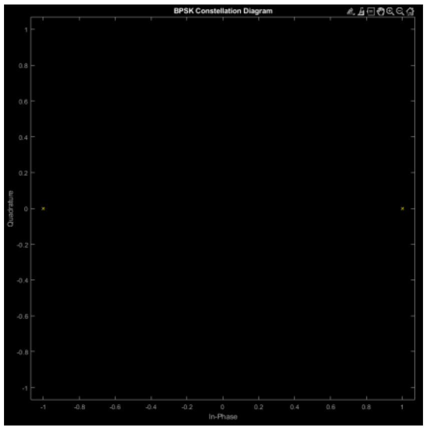

Figure 18: *B-PSK Signal Constellation*

Number of bits parameter ***"N_bits"*** is choosen as 100 in the figure above for better visualization of the modulated signal. Actual number of bits is as much as higher, for example 10000, in order to obtain smooth BER analysis graphs.

Then, the for loop is created for analyzing the process for each SNR values for i = 1,2,3…N, where N = 61. According to the SNR values, additive white Gaussian noise addition is applied.

For different channel coefficients, channel response is used for obtaining channel matrix by usin built-in function toeplitz(.). Number of taps is chosen according to the length of channel. If the maximum delay index in channel is k, then the number of tap is equal to 2*k+1.

channel_matrix_zf
7x7 double

	1	2	3	4	5	6	7
1	0.9000	0.2000	0.1000	0.0500	0	0	0
2	0.3000	0.9000	0.2000	0.1000	0.0500	0	0
3	0.1000	0.3000	0.9000	0.2000	0.1000	0.0500	0
4	0.0500	0.1000	0.3000	0.9000	0.2000	0.1000	0.0500
5	0	0.0500	0.1000	0.3000	0.9000	0.2000	0.1000
6	0	0	0.0500	0.1000	0.3000	0.9000	0.2000
7	0	0	0	0.0500	0.1000	0.3000	0.9000
8							

channel_matrix_zf channel_matrix_mmse
7x7 double

	1	2	3	4	5	6	7
1	1.0150	0.5100	0.2650	0.1400	0.0350	0.0100	0.0025
2	0.5100	1.0150	0.5100	0.2650	0.1400	0.0350	0.0100
3	0.2650	0.5100	1.0150	0.5100	0.2650	0.1400	0.0350
4	0.1400	0.2650	0.5100	1.0150	0.5100	0.2650	0.1400
5	0.0350	0.1400	0.2650	0.5100	1.0150	0.5100	0.2650
6	0.0100	0.0350	0.1400	0.2650	0.5100	1.0150	0.5100
7	0.0025	0.0100	0.0350	0.1400	0.2650	0.5100	1.0150
8							

Figure 19: *7-tap Channel Matrices*

After convolving the noisy output with the channel matrix, matched filter and decoding method that is choosen as hard-decision decoding are applied. Number of errors is detected to simulate BER.

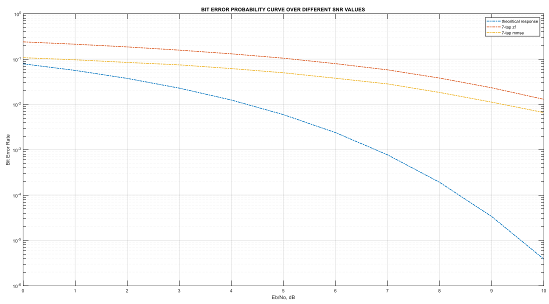

Figure 20: *BER Analysis*

To observe the effect of noise and ISI, eyediagram is plotted for each equalization methods in both 0 dB and 10 dB SNR values.

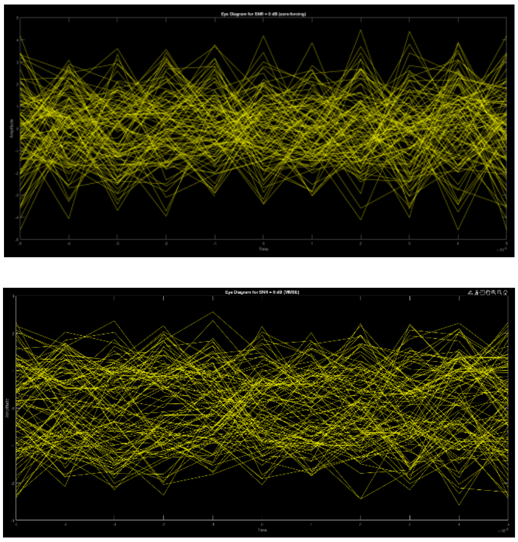

Figure 21: *Eyediagrams for 0 dB SNR value*

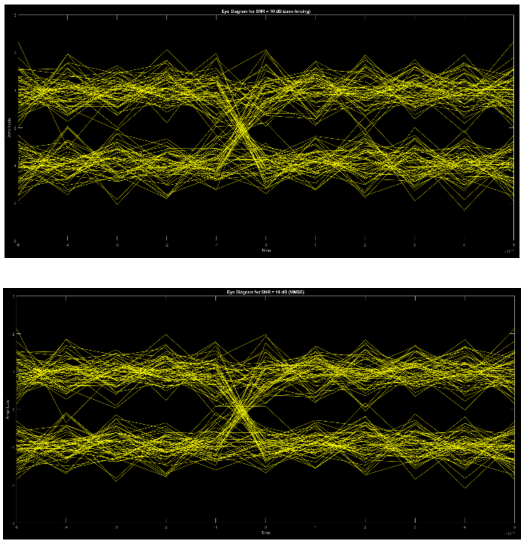

Figure 22: *Eyediagrams for 15 dB SNR value (Zero-Forcing and MMSE in order)*

An eye diagram is an important tool in digital communications for evaluating the quality of a digital signal. It is created by superimposing parts of a signal to create a pattern like an eye. The openness of the "eye" is a critical measure of signal quality. Several factors influence this openness:

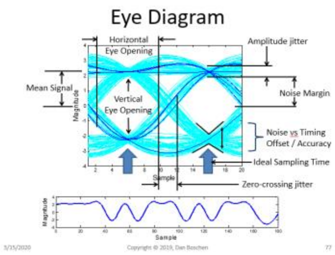

Figure 23: *Eye Diagram Properties*[10]

Amplitude: The magnitude of the signal is reflected in the vertical opening of the eye. A bigger hole indicates clear separation between signal levels, which is required for accurate data interpretation.
Timing Jitter: The horizontal width of the eye represents the degree of timing jitter, or the variability in the timing of signal transitions. Less jitter leads in a wider eye, indicating a sharper signal.
Inter-symbol Interference (ISI): ISI causes the eye to close horizontally. It happens when former signals interact with current ones, which is commonly caused by signal reflections or spreading.
Signal-to-Noise Ratio (SNR): A clearer eye pattern indicates a greater SNR, indicating a stronger signal in comparison to background noise. In this case, as it seems from the figure 21, eyes are more opened when MMSE equalizer is applied. This result is related to noise mitigation ability of MMSE equalizer is more than Zero-Forcing equalizer.
Rise and Fall Times: Faster rise and fall times (transition times between signal levels) can result in a more open eye but may increase ISI.
Bandwidth: The bandwidth of the system influences signal transition sharpness, which impacts the clarity of the eye diagram.
Impairments in the System: Factors such as phase noise and amplitude noise can impair the openness of the eye, with a well-designed system minimizing them to retain clarity.

IV. CONCLUSION

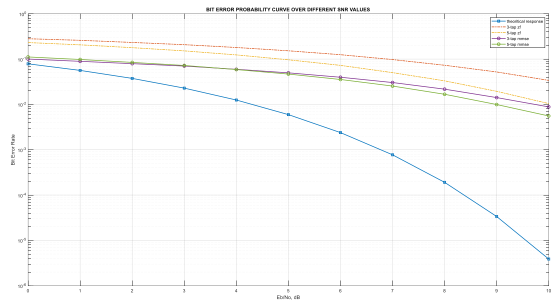

Figure 24: *BER Analysis With Different Tap Equalizer Filters*

Figure 24 shows that, choosing number of tap according to the channel is a crucial parameter during equalization process in digital communication. Maximum index number of delay should be considered while choosing ***"k"*** parameter[8].

To sum up everything that has been stated so far, in the context of digital communication systems, the project effectively compared Zero Forcing and MMSE Equalizers. The main discovery is that the choice of equalization is significantly affected by the signal-to-noise ratio (SNR) in the transmission medium. While the Zero Forcing Equalizer performs better in high SNR conditions by suppressing intersymbol interference (ISI), the MMSE Equalizer performs better in low SNR situations due to its noise mitigation capabilities.

The trade-off between these two strategies is a crucial observation that requires careful attention while optimizing digital transmission systems. This balancing tries to reduce bit error rates while increasing communication efficiency. The project's simulations and analyses provide insight on these equalization techniques, allowing for a better understanding of their usage in different communication environments. This understanding is essential for creating more efficient and dependable digital communication systems.

REFERENCES

[1] *"Digital Communications"*, John G. Proakis , 4th ed. New York: McGraw Hill, 2001.
[2] "*Fundamentals of Communication Systems"*, John G. Proakis and Masoud Salehi, 2nd edition, 2005.
[7] *"Wireless Communications and Networking"*., "Ch.4", Jon Mark and Weihua Zhuang (2003)., Prentice Hall. p. 139.
[9] *"Implementation Of Zero Forcing And MMSE Equalization Techniques In OFDM"*, Ferran Conillera Vilar, December, 2014.
[10] "*Eye Pattern Construction and Interpretation*", Dan Boschen, 2019.

Appendix

[3] MATLAB code for signal representations

```
clc;clear;close all;
tb = 1/1000;
ta= tb/1000;
t = 0:ta:(4*tb)-ta;
s=cos(2*pi*2000*t);
s(1:1000)=0;
s(3000:4000)=0;
plot(t,s,"black")
xlabel("Tb")
ylabel("Amplitude");
%%
clc;clear;close all;
tb = 1/1000;
ta= tb/1000;
t = 0:ta:(4*tb)-ta;
s1=cos(2*pi*3000*t(1001:3000));
s0_1=cos(2*pi*1000*t(1:1000));
s0_2=cos(2*pi*1000*t(3001:4000));
figure(1);
plot(t(1:1000),s0_1,"black")
hold on
plot(t(1001:3000),s1,"black")
hold on
plot(t(3001:4000),s0_2,"black")
xlabel("Tb")
ylabel("Amplitude");
%%
clc;clear;close all;
tb = 1/1000;
ta= tb/1000;
t = 0:ta:(4*tb)-ta;
s0_1= cos(2*pi*2000*t(1:1000));
s1= cos(2*pi*2000*t(1001:3000)+pi);
s0_2= cos(2*pi*2000*t(3001:4000));
figure(1);
plot(t(1:1000),s0_1,"black")
hold on
plot(t(1001:3000),s1,"black")
hold on
plot(t(3001:4000),s0_2,"black")
xlabel("Tb")
ylabel("Amplitude");
```

[4] MATLAB code for signal constellation with B-ASK

```
clc;
clear all;
%%
Fs = 1000;
Eb= (1/sqrt(2))*sqrt(1/Fs);
sz = 250;
figure(1)
scatter(0,0,sz,'x',"black");
hold on
scatter(Eb,0,sz,'x',"black");
grid on
xlabel("Ø_0(t)");
ylabel("Ø_1(t)");
title("Signal Constellation for B-FSK");
%%
```

[5] MATLAB code for signal constellation with B-FSK

```
clc;
clear all;
%%
Fs = 1000;
Eb= (1/sqrt(2))*sqrt(1/Fs);
fi2 = 0:0001:0.5;
fi1 = 0:0001:0.5;
sz = 250;
figure(1)
scatter(Eb,0,sz,'x',"black");
hold on
scatter(0,Eb,sz,'x',"black");
grid on
xlabel("Ø_0(t)");
ylabel("Ø_1(t)");
title("Signal Constellation for B-FSK");
%%
```

[6] MATLAB code for signal constellation with B-PSK

```
clc;
clear all;
%%
Fs = 1000;
Eb= (1/sqrt(2))*sqrt(1/Fs);
fi2 = 0:0001:0.5;
fi1 = 0:0001:0.5;
sz = 250;
figure(1)
scatter(-Eb,0,sz,'x',"black");
hold on
scatter(Eb,0,sz,'x',"black");
grid on
xlabel("Ø_0(t)");
ylabel("Ø_1(t)");
title("Signal Constellation for B-PSK");
%%
```

[8] MATLAB code "P3.m"

```
clc;
clear all;
close all;
%% signal generation
%%
N_bits = 100;
bit_duration = 1e-3;
samples_per_bit = 10;
bits = randi([0 1], 1, N_bits);
%% reshaping
%%
signal = repelem(bits, samples_per_bit);
%% creating time vector
%%
fs = 1 / (bit_duration/samples_per_bit);
ts = 1/fs;
time = 0:ts:(length(bits)*bit_duration) - ts;
%% figures
%%
figure;
subplot(211);
stem(bits, 'Marker', 'none');
title('Original Bits');
subplot(212);
plot(time, signal);
title('Signal with 1 ms Bit Duration');
%% modulation (B-PSK)
%%
fc = fs*10; % Adjust the carrier frequency based on our requirements
mxsig = zeros(1, length(signal));
for i=1:length(signal)
    if signal(1,i) == 1
        mxsig(1,i) = cos(2*pi*fc*time(i)) ;
    elseif signal(1,i) == 0
        mxsig(1,i) = cos(2*pi*fc*time(i)+pi);
    end
end
%% figures
%%
figure;
subplot(2,1,1);
stem(bits, 'Marker', 'none');
title('Original Bits');
subplot(2,1,2);
plot(time, mxsig);
title('BPSK Modulated Signal');
figure;
scatterplot(mxsig, 1, 250, 'yx');
title('BPSK Constellation Diagram');
%% channel model
%%
channel_response = [0.05 0.1 0.2 0.9 0.3 0.1 0.05]; % channel 1. for channel
1, 7-tap filter is ideal
% channel_response = [0.2 0.9 0.3]; % channel 2
chan_out = conv(mxsig,channel_response);
%% noise addition
%%
Eb_N0_dB = 0:1:10;
for i = 1:length(Eb_N0_dB)
    pavg = sum(abs(mxsig).^2)/length(mxsig);
    snr_lin = 10^(0.1*Eb_N0_dB(i));
    var_noise = pavg/snr_lin;
    noise = sqrt(var_noise)*randn(1,length(chan_out));
    noisy_out = chan_out + noise; % additive white gaussian noise
```

```
    %% ZERO-FORCING EQUALIZATION
    %%
    L = length(channel_response);
    k = (L-1)/2; % (2*k+1)-tap equalizer design
    channel_matrix_zf = toeplitz([channel_response((L+1)/2:end)
zeros(1,2*k+1-L+(L-1)/2)], [channel_response((L+1)/2:-1:1) zeros(1,2*k+1-
L+(L-1)/2)]);
    d_zf = zeros(1,2*k+1);
    d_zf(k+1) = 1;
    channel_zf  = transpose(inv(channel_matrix_zf)*transpose(d_zf));
    %% matched filter
    N_msg = length(signal);
    yFilt_zf = conv(noisy_out,channel_zf);
    yFilt_zf = yFilt_zf(k+(L+1)/2:end);
    yFilt_zf = conv(yFilt_zf,ones(1,1)); % convolution
    ySamp_zf = yFilt_zf(1:1:N_msg);  % sampling at time T
    %% receiver - hard decision decoding
    yDecoded_zf = real(ySamp_zf)>0;
    if(Eb_N0_dB(i) == 0)
        yout1_zf = ySamp_zf;
    elseif(Eb_N0_dB(i) == 10)
        yout2_zf = ySamp_zf;
    end
    %% counting the errors
    nErr_zf(i) = size(find(signal - yDecoded_zf),2);
    %% MMSE EQUALIZATION
    %%
    h_autocorr = conv(channel_response,fliplr(channel_response));
    channel_matrix_mmse = toeplitz([h_autocorr(L:end) zeros(1,2*k+1-L)],
[h_autocorr(L:end) zeros(1,2*k+1-L)]);
    channel_matrix_mmse = channel_matrix_mmse + (1/2)*(10^(-
Eb_N0_dB(i)/10))*eye(2*k+1);
    d_mmse = zeros(1,2*k+1);
    d_mmse([-(L-1)/2:(L-1)/2]+k+1) = fliplr(channel_response);
    channel_mmse  = transpose(inv(channel_matrix_mmse)*transpose(d_mmse));
    %%  matched filter
    yFilt_mmse = conv(noisy_out,channel_mmse);
    yFilt_mmse = yFilt_mmse(k+(L+1)/2:end);
    yFilt_mmse = conv(yFilt_mmse,ones(1,1)); % convolution
    ySamp_mmse = yFilt_mmse(1:1:N_msg);  % sampling at time T
    %% receiver - hard decision decoding
    yDecoded_mmse = real(ySamp_mmse)>0;
    if(Eb_N0_dB(i) == 0)
        yout1_mmse = ySamp_mmse;
    elseif(Eb_N0_dB(i) == 10)
        yout2_mmse = ySamp_mmse;
    end
    %% counting the errors
    nErr_mmse(i) = size(find(signal - yDecoded_mmse),2);
end
simBer_zf = nErr_zf/N_msg; % simulated ber
simBer_mmse = nErr_mmse/N_msg; % simulated ber
theoryBer = 0.5*erfc(sqrt(10.^(0.1*Eb_N0_dB))); % theoretical ber
%% figures
%%
figure;
semilogy(Eb_N0_dB,theoryBer,'-.','LineWidth',1.5);
hold on
semilogy(Eb_N0_dB,simBer_zf,'-.','LineWidth',1.5);
% hold on
semilogy(Eb_N0_dB,simBer_mmse,'-.','LineWidth',1.5);
hold off
grid on
legend('theoritical response','7-tap zf','7-tap mmse');
xlabel('Eb/No, dB');
ylabel('Bit Error Rate');
title('BIT ERROR PROBABILITY CURVE OVER DIFFERENT SNR VALUES');

eyediagram(yout1_zf, samples_per_bit, ts, 0);
title("Eye Diagram for SNR = 0 dB (zero-forcing)");

eyediagram(yout1_mmse, samples_per_bit, ts, 0);
title("Eye Diagram for SNR = 0 dB (MMSE)");

eyediagram(yout2_zf, samples_per_bit, ts, 0);
title("Eye Diagram for SNR = 10 dB (zero-forcing)");

eyediagram(yout2_mmse, samples_per_bit, ts, 0);
title("Eye Diagram for SNR = 10 dB (MMSE)");
```