ZEWAIL CITY

UNIVERSITY OF SCIENCE AND TECHNOLOGY

DIGITAL COMMUNICATIONS SIGNAL RECOVERY USING MATCHED FILTER

CIE 327

Under Supervision of
Dr. Ahmed Eltrass
Eng. Hassan, Eng. Retaj

Omar Ayman 202100443
Mohamed Morshedy 202000372
Ziad Behairy 202100154

August 2023

Contents

1 Introduction

a mathematical foundation for quantifying likelihood, finds essential application in modern digital communications. This project directly demonstrates how probability principles are employed to combat noise in digital communication systems. Through simulation, a random signal is modulated using M-ary Pulse Amplitude Modulation (M-PAM), subjected to Additive White Gaussian Noise (AWGN), and recovered via a matched filter. Varying Signal-to-Noise Ratios (SNRs) reveals their influence on Bit Error Rate (BER), culminating in a SNR-BER relationship that unveils vital insights. This project exemplifies the tangible impact of probability in noise mitigation for robust digital communication.

2 Background information

2.1 M-ary Pulse Amplitude Modulation (M-PAM)

The sole purpose of communication systems is to send a message signal from one point to be received in another. The problem is that a message signal is weak sothat it could not be transmitted for long distances, or it will fade away, and this is where modulation is needed. Modulation is a process in which the desired weak low-frequency message signal is combined with a high-frequency carrier signal by choosing a parameter of the wave carrier to be varied according to variations in the message signal. The combined signal is then sent and received at the receiving end, where it is divided in the two original signals again (called demodulation), saving all information in the message signal from getting lost. Modulation is composed of continuous-wave modulation and pulse modulation. In Pulse Modulation, the signal's information is transmitted by pulses. Our concern is M-ary Pulse Amplitude Modulation (M-PAM), which is a type of pulse modulation, in which pulses can have one of M-levels (amplitudes), represented by N bits, where $N = log_2(M)$.

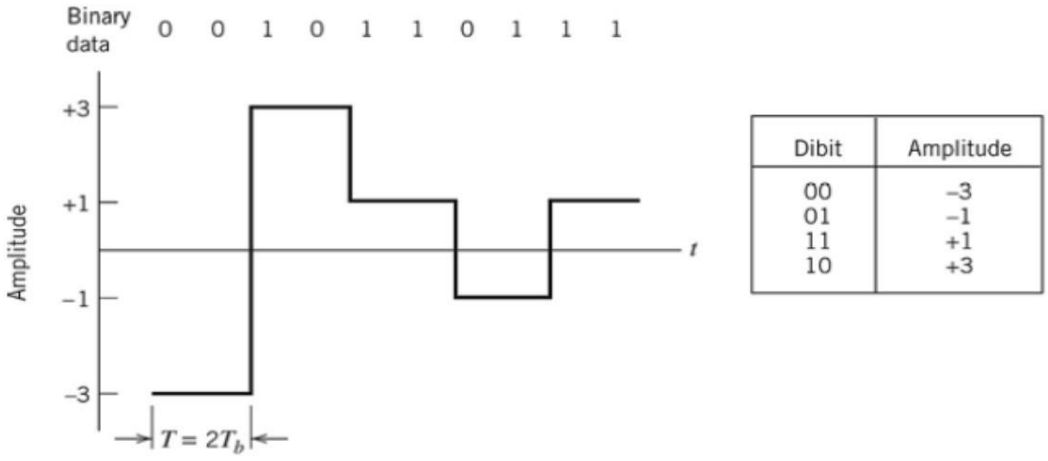


Figure 1: 4-PAM System

2.2 Additive White Gaussian Noise (AWGN)

Additive White Gaussian Noise is a model used to imitate the presence of natural noises and random processes that might interrupt a signal.

Additive means that the noise is added to the transmitted signal. For example, it the signal is $S(t)$, and the noise can be represented by $u(t)$ then the resulting noisy signal is $N(t) = S(t) + u(t)$.

White means that the noise has uniform power over the whole frequency band. The name is analogous to white color being composed of all frequency components in the visible spectrum.

Gaussian means that the noise values (u) follow Gaussian distribution with zero mean ($\mu = 0$), meaning that they are concentrated around the zero and are less likely to appear away from the zero.

$$u \sim Gaussian\ (\mu = 0, \sigma^2) \tag{2.2.1}$$

AWGN is also known as thermal noise because its power is directly proportional to the equivalent temperature at the receiver.

Gaussian (or Normal) distribution is a probability distribution that falls under the category of continuous probability distributions. A normally distributed random variable (x), like AWGN noise we consider here, has a probability density function $f(x)$ given by:

$$f(x) = \frac{1}{\sigma\sqrt{2}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \tag{2.2.2}$$

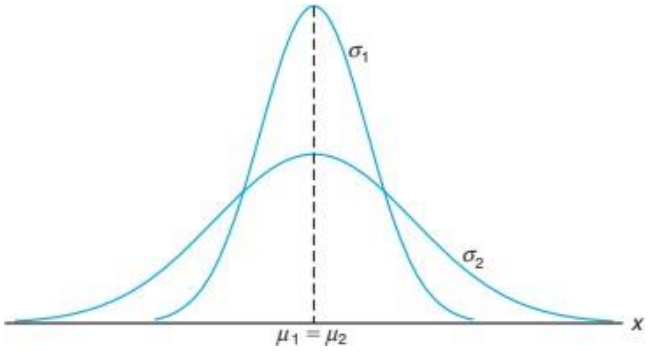


Figure 2: Normal curves with $\mu_1 = \mu_2$ and $\sigma_1 < \sigma_2$

2.3 Matched Filter

A matched filter is a linear receiver filter used to maximize the Signal to Noise Ratio (SNR). It is obtained by applying the process of *cross-correlation* between a known signal (called *template*) and the unknown received signal to detect portions of similarity between the signal and the template. For example, a received signal that is supposed to contain only bits of zeros and ones could cross-correlated with a single pulse to determine the segments containing zeros and the segments containing ones. *Cross-correlation* is a mathematical operation that provides a measure of similarity between two signals as a function of shifting amount of one relative to another. It is similar to convolution with a single difference, being that in convolution the sliding signal is time-reversed. Given two discrete-time signals $x[n]$ and $y[n]$ defined of the interval $[-n, n]$, the cross-correlation between them R_{xy} is given by:

$$R_{xy} = \sum_{n=-N}^{N} x[n]y[n+k] \text{ , where } R_{xy} \text{ is a function of } k \quad (2.3.1)$$

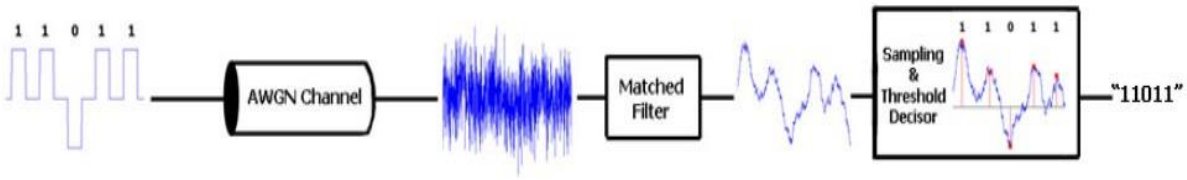


Figure 3: System uses Matched Filter

2.4 Bit Error Rate (BER)

To measure the system efficiency, the transmitted and received signals are compared to calculate the Bit Error Rate *(BER)*, which is defined as the number of Bit Errors divided by total number of transmitted bits. A bit error is a mismatch in a certain bit between the transmitted and received signals. The concept of BER is used in 2-PAM and is extended in a general M-PAM to Symbol Error Rate *(BER)*, since a level is now represented by a symbol (multiple bits) and not only one bit.

3 Methodology and Results

Two simulations were applied using a Simulink model and a MATLAB script.

3.1 Simulink

3.1.1 Used Blocks

1. **Random Integer Generator:** generates random integers that represent the transmitted signal.

Figure 4: Random Integer Generator

2. **M-PAM Modulator:** modulates the transmitted signal following M-PAM, where M is a variable that must be provided in the workspace before running the simulation.

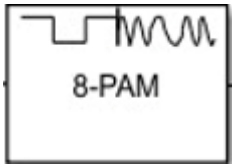


Figure 5: M-PAM Modulator

3. **AWGN channel** adds additive white Gaussian noise to the transmitted signal.

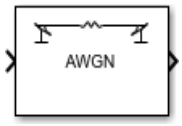


Figure 6: AWGN Channel

4. **M-PAM Demodulator:** demodulates the received modulated signal.

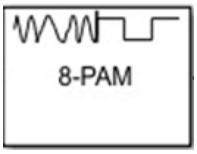


Figure 7: M-PAM Demodulator

5. **Error Rate Calculation:** calculates bit error rate.

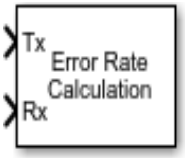
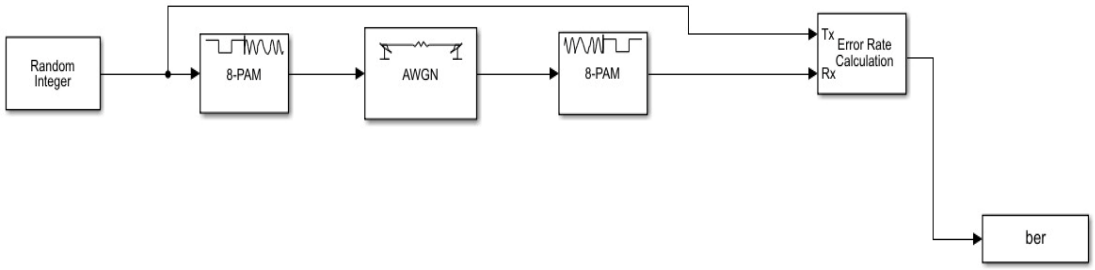


Figure 8: Error Rate Calculation



3.1.2 Results

The dotted curves in fig.5 represent the relation between Eb/N0 and SNR of simulated signal at different Ms and the continuous curves represent the same relation, but theoretically. As M increases, the BER (SER for M > 2) falls more slowly with increasing the Eb/N0, meaning that errors increase, which is consistent as increasing number of symbols also results in increasing the probability of having an error per symbol.

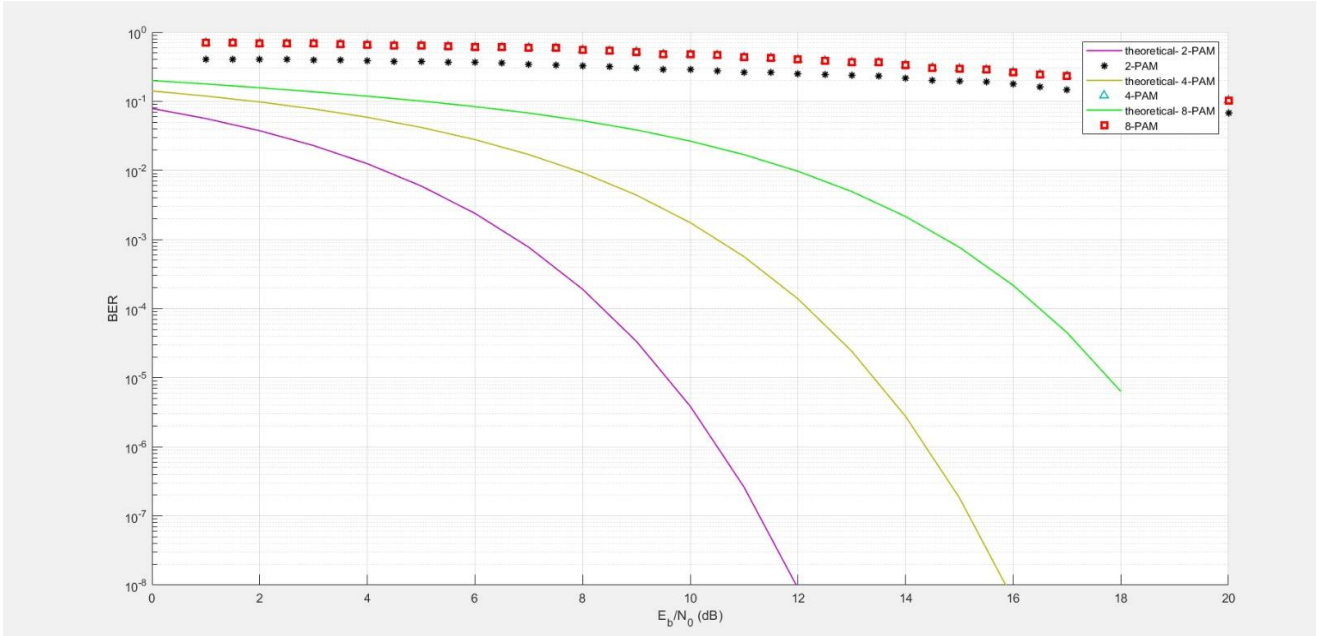


Figure 11: Simulink Results

3.2 MATLAB

3.2.1 Results

The same case of Simulink can be noticed here, as M increases, the BER (SER for M > 2) decreases as a slower rate with increasing Eb/N0, implying a higher probability of error per symbol.

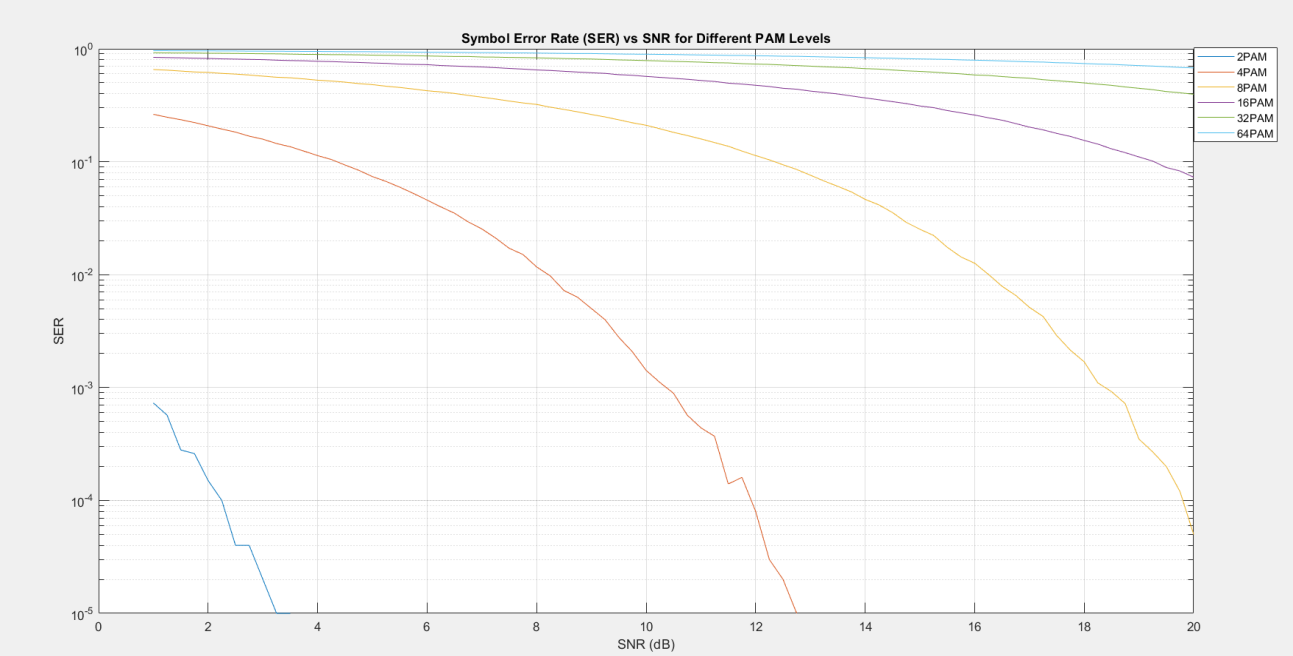


Figure 12: MATLAB Results

4 Conclusion

Probability of having a higher SER increasing with increasing M. This conclusion in strengthened with the agreement between the results of MATLAB and Simulink.

5 References

1. Haykin, S. S. (1994). 4. In Communication systems (4th ed.). Chapter, Wiley.
2. Proakis, J. G., & Salehi, M. (2008). Digital Communications (5th ed.). McGraw-Hill.
3. Haykin, S. (2001). Communication Systems (4th ed.). Wiley.
4. AWGN Channel - Simulink - MathWorks. [Online]. Available: https://www.mathworks.com/help/comm/ref/awgnchannel.html
5. M-PAM Modulator Baseband - MathWorks. [Online]. Available: https://www.mathworks.com/help/comm/ref/mpammodulatorbaseband.html
6. MATLAB randi - Array of random integers - MathWorks. [Online]. Available: https://www.mathworks.com/help/parallel-computing/distributed.randi.html

6 Work Distribution

Task	Done By
MATLAB Code	Ziad , Mohamed
Simulink Simulation	Omar
Writing Report	Omar , Mohamed , Ziad

7 Appendices

7.1 Appendix A: MATLAB Code Implementation

```
%% Array of different M values to be drawn for comparison
M_ALL = [2 4 8 16 32 64];
SNR_range = 1:0.25:20;

% Initialize BER matrix
BER = zeros(length(SNR_range), length(M_ALL));

%% Loop over each M value
for m = 1:length(M_ALL)
    M = M_ALL(m);             % Set the value for M with each iteration
    L = 4;                    % Number of bits in a pulse (length of
pulse)
    pulsesCount = 100000;     % Total number of pulses in the input
signal

    % Generate pulse and its reversed version for matched filtering
    pulse = ones(1, L) ./ sqrt(L);
    reversedPulse = fliplr(pulse);

    % Generate random signal to be modulated
    Signal = randi([0, M - 1], 1, pulsesCount) .* 2 - 1;

    % Prepare the modulated signal for matched filtering
    modulated_signal = repelem(Signal, L) ./ sqrt(L);

    % Simulate matched filtering for different SNR values
    for snr_idx = 1:length(SNR_range)
    SNR = SNR_range(snr_idx);  % Set SNR value with each iteration

    % Introduce AWGN (Additive White Gaussian Noise) to the modulated
signal
    noisySignal = awgn(modulated_signal, SNR, 'measured');
    % The 'measured' option adjusts noise power based on the desired
SNR.
    % This adds simulated noise to the modulated signal to simulate
real-world transmission.

    % Perform convolution for matched filtering
    Result = conv(noisySignal, reversedPulse);
    % Convolve the noisy signal with the reversed pulse using linear
convolution.
    % This is the process of matched filtering, where we try to match
the pulse shape to recover the transmitted symbols.

    Output = Result(L:L:end);
    % Extract symbols after matched filtering.
    % We take every Lth sample of the Result array starting from index
L.
    % This step removes the redundant convolved samples that are not
aligned with the original signal.

    % Calculate the number of symbol errors (BPSK: +1 for errors)
    errorsNum = sum(abs(round(Output) - Signal) > 1);
    % Round the matched filter output to obtain detected symbols.
```