Implementation of M-ary PSK in AWGN Environment with Huffman Coding

Qiaoyu Xu
ECE dept. of Concordia
ID:40292010

I. Abstract

This report explores the implementation of M-ary Phase Shift Keying (M-PSK) modulation in an Additive White Gaussian Noise (AWGN) environment with Huffman coding. The study aims to analyze the performance of various M-PSK schemes (BPSK, QPSK, 8PSK, 16PSK, 32PSK, and 64PSK) by evaluating their Bit Error Rate (BER) and Symbol Error Rate (SER) with and without Huffman encoding. Huffman coding is applied to compress the data before transmission, with its impact on system performance being a key focus. Simulations were conducted to observe the effect of different Eb/N0 values on the performance of each modulation scheme. Results show that higher-order PSKs experience higher sensitivity to noise and require more energy to maintain the same BER. Additionally, Huffman coding introduces further challenges by increasing the complexity of the system, which leads to higher BER and SER, particularly at lower Eb/N0 values. The findings suggest that while Huffman encoding can achieve data compression, it is not ideal for high-order PSK schemes due to their increased sensitivity to noise. The report concludes that for systems where source coding is applied, low-order PSK schemes such as BPSK or QPSK are preferable.

II. Introduction

In modern digital communication systems, modulation is a essential part that enable efficient data transition. Amount those techniques, M-ary Phase Shift Keying (M-PSK) is a common and important one widely used in wireless communication.

Besides modulation, a compression coding or source coding scheme is used to further reduce the bits that need to be transmit. The Huffman Coding scheme is used in the implementation. The Huffman coding has a feature to create variable-length code based on the occurrence of symbol. A more frequent symbol would use lesser bits. [1]

This report aims to analyze the BER and Huffman SER of M-ary PSK in Addictive White Gaussian Noise (AWGN) channel over varies number of Eb/N0. The Matlab simulaton should show a deeper understanding of the fundamental combination of digital modulation and source coding.

Identify applicable funding agency here. If none, delete this.

III. Method

As shown in Figure 1, the simulation begins with text reading, where the system reads a .txt file containing a section from the book Harry Potter. This is done using fileread, which imports the entire text as a string. The next step converts the characters into their 8-bit ASCII values, using uint8 and then de2bi to create a stream of binary values. These form the original bitstream that will be encoded and transmitted.

Instead of generating random binary data, actual text is used. The reason for this is rooted in the nature of Huffman coding: the compression efficiency relies on non-uniform symbol distributions. A purely random binary stream would give every 8-bit symbol (0–255) equal likelihood, making Huffman coding ineffective. Real-world text, however, includes naturally frequent characters (like ”e”, ”A”, ”,”, ”.”) which get assigned shorter codes by Huffman encoding, providing true lossless data compression.

Next, Huffman encoding is applied. The binary stream is regrouped into 8-bit symbols, which are frequency-analyzed using accumarray and unique. A Huffman dictionary is generated via huffmandict, and huffmanenco encodes the symbol stream into a compressed binary sequence.

The encoded bitstream is then modulated using M-ary PSK where M = 2, 4, 8, 16, 32 or 64. The rest of the method section will use M = 64 as the example. Since 64-PSK transmits 6 bits per symbol, the encoded bitstream is padded to ensure its length is a multiple of 6. The bits are then grouped and mapped into symbols using Gray coding, which minimizes bit errors during demodulation by ensuring that adjacent constellation points differ by only one bit.

The 64-PSK constellation is constructed on the unit circle using exp(1j*angles) with an angular offset (π/64) for improved detection performance. Transmission through the AWGN channel is simulated by adding complex Gaussian noise with variance calculated from the desired Eb/N0 ratio. The simulation iterates over Eb/N0 values from 0 dB to 20 dB in steps of 1.

The demodulation process involves calculating the minimum Euclidean distance between the received noisy symbol and all ideal constellation points. The nearest match is taken as the estimated symbol, and inverse Gray coding is used to recover the transmitted bits.

Notably, this simulation in Figure 1 does not perform Huffman decoding at the receiver end. Instead, the demodulated bitstream is directly compared to the original encoded bitstream for BER (Bit Error Rate). This is a commonly used and effective approach in communication simulations where the focus is on transmission errors rather than full source reconstruction.

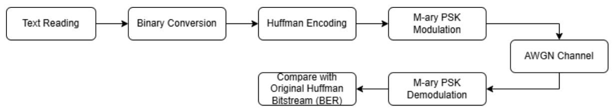

Fig. 1. Block diagram of the BPSK communication system without Huffman Decoding

In Figure 2, the system includes Huffman decoding, the Huffman Symbol Error Rate (SER), and the true end-to-end Bit Error Rate (BER). The term true end-to-end BER refers to the fact that the received bits are Huffman-decoded before being compared with the original input bits. This measurement is particularly challenging for Huffman coding due to its variable-length nature, which can result in discrepancies between the lengths of the decoded bitstream and the original bitstream.

The BER is calculated using the following approach:

1. The total number of bit errors is updated per Huffman symbol transmission, rather than per PSK symbol. This also allows for accurate calculation of the Huffman SER.

2. The minimum length between the original and decoded bitstreams is determined, and bit mismatches are counted only within this range.

3. Any length mismatch—whether the decoded bitstream is shorter or longer than the original—is added to the total number of error bits.

For both implementations, the BER and Huffman SER are averaged over 100 iterations.

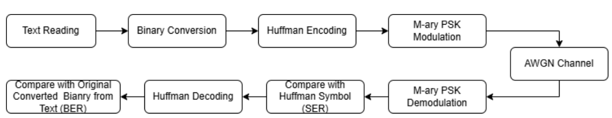

Fig. 2. Block diagram of the BPSK communication system with Huffman Decoding (End to End)

IV. Result

From Figure 3 to Figure 6, the constellation diagrams of QPSK are shown for Eb/N0 values of 20, 10, 7, and 5, respectively. The red dots indicate the ideal positions of the modulated signal values, while the blue dots represent the noise cloud formed by the distorted signals.

From Figure 7 to Figure 10, the BER performance of each PSK implementation without Huffman decoding is shown for BPSK, 8PSK, 16PSK, 32PSK, and 64PSK. The x-axis represents Eb/N0 values ranging from 0 to 20 dB, and the y-axis shows the corresponding BER. The simulated results with Huffman coding are shown in blue, while the theoretical BER of each PSK scheme without Huffman coding is shown in red.

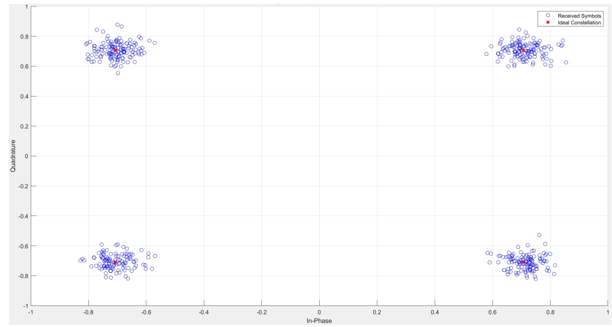

Fig. 3. Constellation diagram of QPSK for Eb/N0 = 20

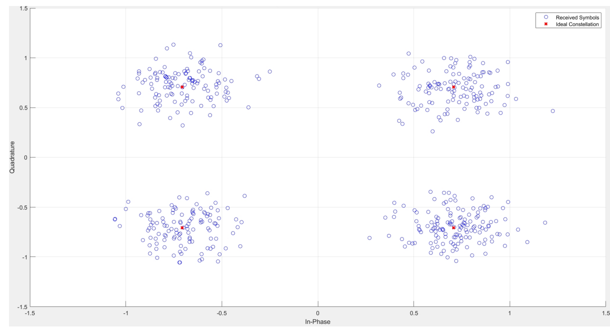

Fig. 4. Constellation diagram of QPSK for Eb/N0 = 10

The BER without Huffman decoding for each PSK is shown in the comparison in Figure 11. The lighter blue line represents 64PSK, green represents 32PSK, purple represents 16PSK, yellow represents 8PSK, orange represents QPSK, and finally, the darker blue represents BPSK.

The BER with Huffman decoding for each PSK is shown in the comparison in Figure 12, using the same color configuration as before.

Finally, the Huffman SER is shown in Figure 13, with the same color configuration.

V. Discussion

First, let's take the constellation diagram of QPSK for various values of Eb/N0 as an example. From Figure 3 to Figure 6, we observe that a high Eb/N0 ratio results in fewer errors, with the noise cloud staying small enough to remain within the decision area. When Eb/N0 = 20 and 10, no errors occur, which is also reflected later in the BER diagram. As Eb/N0 decreases, some spikes in the cloud begin to trespass into other decision areas, and the situation worsens at Eb/N0 = 5. Since the noise is AWGN, the noise cloud forms in such a way that the distorted signals in the constellation diagram are within a circular shape, with the radius of the circle being proportional to the noise level.

The constellation diagram provides an intuition that for PSKs with higher M, the points of modulated signals become more crowded. As Eb/N0 drops, PSKs with higher M experience much higher BER than PSKs with lower M.

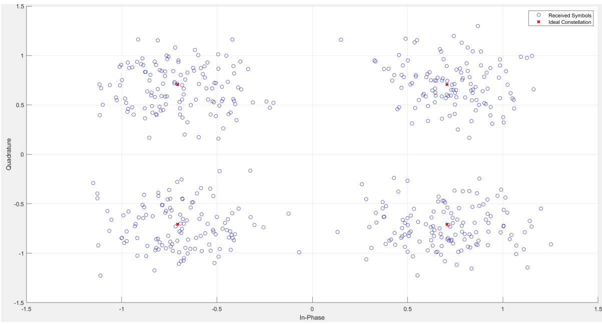

Fig. 5. Constellation diagram of QPSK for Eb/N0 = 7

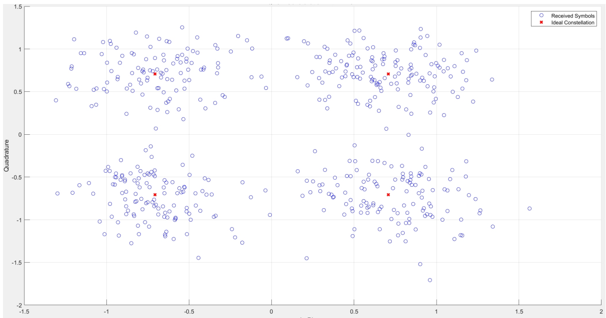

Fig. 6. Constellation diagram of QPSK for Eb/N0 = 5

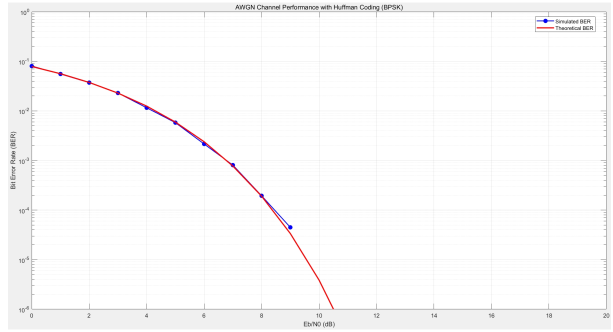

Fig. 7. BER plot of BPSK without Huffman Decoding

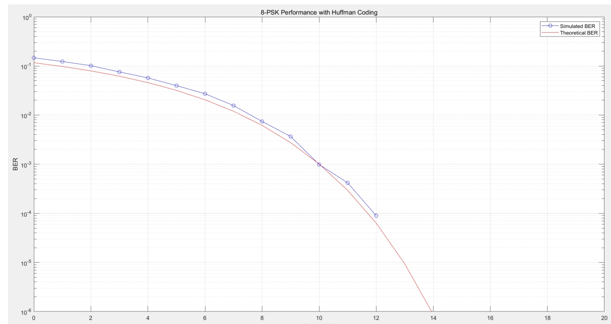

Fig. 8. BER plot of 8PSK without Huffman Decoding

Next, the BER performance of each PSK without Huffman decoding is shown from Figure 7 to Figure 10. The first observation is that the simulated results for PSKs with M bigger and equal to 8 are slightly shifted to the right. This indicates that, in order to achieve the same performance as the theoretical uncoded systems, these systems require more Eb/N0. The reason is that the AWGN added to the channel causes more errors. The higher the M of PSK, the more the BER shifts to the right from the theoretical BER, meaning PSKs with higher M are more sensitive to noise.

To achieve a certain BER, for example, 10^{-5}, the 8PSK requires approximately 1.5 times more energy than QPSK and BPSK to maintain the same BER. The 16PSK also requires about 1.5 times more energy than 8PSK to achieve the same BER. In the case of 32PSK and 64PSK, they are not recommended for practical use due to their high sensitivity to noise.

Theoretically, the BER without Huffman decoding is equivalent to that of PSKs over an AWGN channel, since it compares the encoded Huffman bitstream with the decoded Huffman bitstream. Notably, as Figure 11 shows, BPSK and QPSK have nearly the same BER.

Now, the BER extends to include Huffman decoding in Figure 12. Due to Huffman coding, a single bit error in the Huffman symbol can lead to a significantly incorrect decoded bitstream, which is 8-bit ASCII in this case. Without advanced techniques like bit correction, the BER for each PSK is much worse than the BER without Huffman coding. Therefore, the BER remains high and flat at lower Eb/N0 values. Fortunately, the BER still drops below 10^{-5}, which is a common target, in the simulation. For BPSK and QPSK, the Eb/N0 is still manageable in practice. Interestingly, BPSK and QPSK show different BERs in this case.

As for the Huffman SER shown in Figure 13., the PSKs with higher M not only suffer more from the crowded constellation diagram, they would also have to send more bits due to the padding for 8 bits of Ascii bit stream can not be divided by M, therefore, they would send more bits which increase the possibility of making error.

VI. Discussion

Data compression is advantageous, as it allows for a reduction of 20-30 percent in the number of bits to transmit. However, the trade-off when using techniques like Huffman encoding is that it requires a significantly higher Eb/N0 to maintain the desired SER. If the focus of transmission is on speed with fewer bits to send, then this approach is still worth considering. Huffman coding also adds more complexity to the system, requiring more advanced strategies to reduce the BER, such as error correction and error detection. Additionally, PSK exaggerates the BER with higher sensitivity to noise, especially when combined with the variable-length nature of Huffman encoding. It is advisable to use very low M PSK if source coding is applied.

References

[1] D. A. Huffman, "A method for the construction of minimum-redundancy codes," *Proc. IRE*, vol. 40,

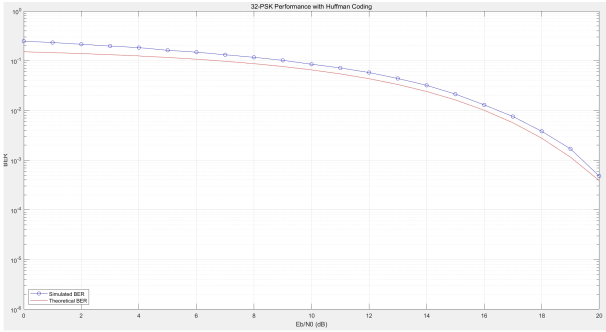

Fig. 9. BER plot of 32PSK without Huffman Decoding

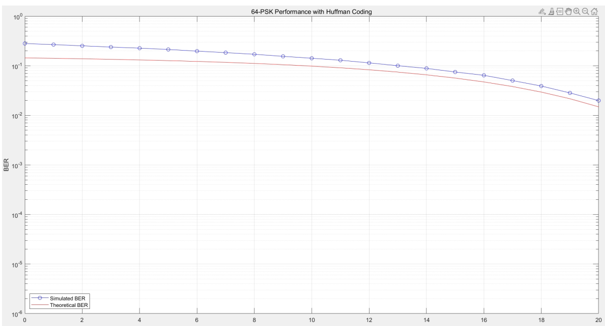

Fig. 10. BER plot of 64PSK without Huffman Decoding

no. 9, pp. 1098-1101, Sep. 1952. doi: [10.1109/JRPROC.1952.273899](https://doi.org/10.1109/JRPROC.1952.273899)

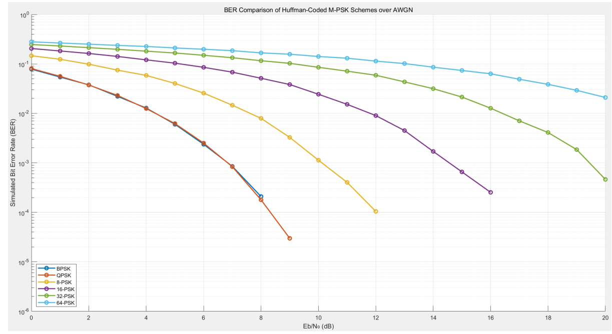

Fig. 11. BER plot of all PSKs without Huffman Decoding

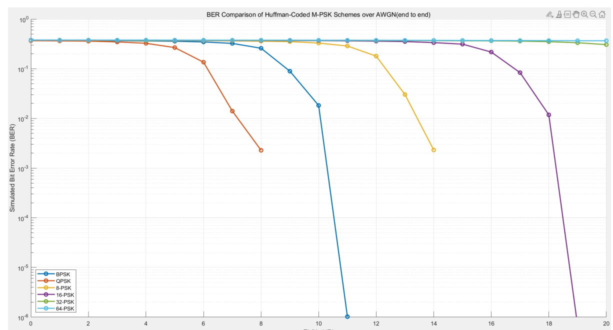

Fig. 12. BER plot of all PSKs with Huffman Decoding

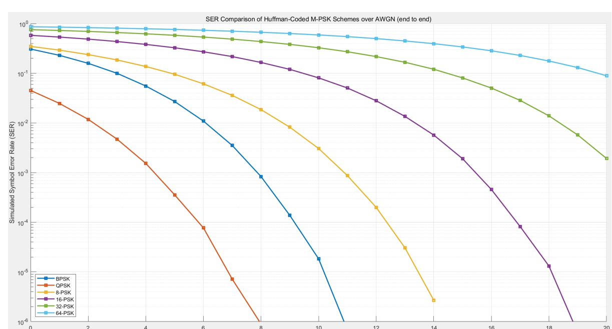

Fig. 13. Huffman SER plot of all PSKs with Huffman Decoding