Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Error Control Coding for Digital Communication Systems Using BCH and Hamming Techniques

Dawit Teklu Weldeslasie

dawit.tekulu@gmail.com

Aksum University

Zenawi Haileslassie Berhe

Aksum University

Mengsteab Dagnew Welemariam

Aksum University

Article

Keywords: Error control coding, Digital communication systems, Hamming code, BCH code, Error detection and correction, Reliable data transmission

Posted Date: August 12th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10203401/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Error Control Coding for Digital Communication Systems Using BCH and Hamming Techniques

Dawit Teklu Weldeslasie[1]* https://orcid.org/0009-0000-5388-5197, Zenawi Haileslassie Berhe[2], Mengsteab Dagnew Welemariam[3]

[1]Aksum University, Aksum, Ethiopia, Department of Information Technology
[2]Aksum University, Aksum, Ethiopia, Department of Computer Engineering
[3]Aksum University, Aksum, Ethiopia, Department of Electronic Communication Engineering

*Corresponding author(s). E-mail(s): dawit.tekulu@gmail.com
Contributing authors: zenawilife@gmail.com, mengsteab2dagnew@gmail.com

Abstract

Reliable transmission of digital information is fundamental to modern communication systems, where channel noise and interference inevitably introduce transmission errors that degrade data integrity. Error control coding plays a critical role in detecting and correcting these errors, thereby improving communication reliability. This study investigates the performance of two widely used error control coding techniques, namely Hamming and Bose–Chaudhuri–Hocquenghem (BCH) codes, for reliable digital communication. An error control framework comprising encoder design, channel simulation, and decoder implementation was developed to evaluate the effectiveness of each coding technique under noisy channel conditions. Hamming codes were employed to detect double-bit errors and correct single-bit errors, whereas BCH codes were implemented to detect and correct multiple-bit errors. The performance of both coding schemes was compared based on their error detection and correction capabilities under identical transmission scenarios. The results demonstrate that BCH codes provide superior error correction performance in noisy communication channels, while Hamming codes offer a computationally efficient solution for applications requiring single-error correction. The findings highlight the trade-off between coding complexity and error correction capability and provide guidance for selecting appropriate error control coding techniques for reliable digital communication systems.

Keywords: Error control coding, Digital communication systems, Hamming code, BCH code, Error detection and correction, Reliable data transmission

1. Background and context

The rapid growth of modern digital communication, data storage, wireless networking, cloud computing, and Internet of Things (IoT) applications has substantially increased the demand for reliable and high-speed information transmission [1]. As communication systems continue to support larger data volumes under increasingly complex operating environments, ensuring the integrity of transmitted information has become a fundamental challenge. Information theory and error-control coding play a pivotal role in addressing this challenge by enabling reliable communication over noisy channels while maintaining high spectral efficiency [2].

During data transmission, communication channels are inevitably affected by noise and other impairments that may alter the transmitted symbols, resulting in transmission errors. To mitigate these errors, channel coding introduces carefully designed redundancy into the transmitted data, allowing the receiver to detect and correct errors without requiring retransmission [3][4]. Consequently, error-control coding has become an indispensable component of modern digital communication systems, data storage technologies, satellite communications, wireless networks, and deep-space communication systems [5].

A typical digital communication system consists of a source encoder, a channel encoder, a communication channel, a channel decoder, and a source decoder. The source encoder converts the original information into a binary bit stream, whereas the channel encoder maps the information bits into structured codewords by introducing controlled

redundancy. After transmission through the channel, the received signal is processed by the channel decoder, which exploits the coding structure and channel characteristics to recover the original information as accurately as possible. Finally, the source decoder reconstructs the transmitted message for delivery to the destination [6].

Among the various channel models, the Additive White Gaussian Noise (AWGN) channel is the most widely adopted benchmark for evaluating the performance of communication systems. In the AWGN model, the transmitted signal is impaired only by additive Gaussian noise with a constant power spectral density. Although this model does not account for channel impairments such as fading, interference, or nonlinear distortion, it provides a mathematically tractable framework for analyzing and comparing coding and decoding algorithms. Consequently, AWGN-based performance evaluation serves as a standard reference for the development and assessment of error-control coding techniques [7].

The design of efficient channel coding schemes requires an optimal balance among transmission reliability, decoding complexity, computational efficiency, and implementation cost. Modern communication systems demand coding techniques capable of approaching channel capacity while maintaining practical encoding and decoding complexity for real-time applications. Accordingly, the development of robust channel encoder–decoder architectures and advanced decoding algorithms remains an active area of research aimed at improving bit error rate (BER) performance, reducing latency, and enhancing the overall reliability of digital communication systems operating under noisy channel conditions [8].

The primary objective of error-control coding is therefore to enable reliable transmission of digital information over noisy channels by maximizing error-correction capability while minimizing bandwidth overhead, computational complexity, and implementation cost. Achieving this balance is essential for next-generation communication systems that require high data rates, low latency, and dependable information delivery.

2. Statement of the problem

The exponential growth of digital communication systems has intensified the demand for reliable, high-speed, and bandwidth-efficient data transmission across wireless networks, satellite communications, data storage systems, and Internet-based applications [9]. However, communication channels are inherently susceptible to noise and other transmission impairments that introduce bit errors, resulting in degraded system performance and reduced communication reliability. As transmission rates continue to increase, designing efficient error-control coding techniques capable of maintaining high reliability with acceptable computational complexity has become a significant research challenge.

Error-control coding addresses this challenge by introducing structured redundancy into the transmitted data, enabling the receiver to detect and correct transmission errors without requiring retransmission [5]. Although numerous coding techniques, including Hamming, and Bose–Chaudhuri–Hocquenghem (BCH) codes, have been developed, each exhibits distinct trade-offs in terms of error-correction capability, decoding complexity, computational cost, and bandwidth efficiency. Consequently, selecting an appropriate coding technique for a given communication environment remains a critical issue in modern digital communication system design.

Among these coding schemes, BCH and Hamming codes have attracted considerable attention because of their strong error-correction performance under noisy channel conditions [10][11]. Nevertheless, a systematic experimental comparison of their encoding and decoding performance under identical simulation settings is essential to better understand their relative strengths, decoding behavior, and communication reliability. Such comparative evaluations provide valuable insights into the suitability of each coding technique for practical communication systems operating over Additive White Gaussian Noise (AWGN) channels [12].

To address this problem, this study develops and evaluates an experimental MATLAB-based simulation framework for implementing and comparing BCH and Hamming coding techniques. The framework performs encoding, transmission through an AWGN channel, and decoding while measuring system performance under different noise conditions. In addition, a graphical user interface (GUI) is developed to facilitate visualization of the encoding and decoding processes and to illustrate the impact of channel noise on decoding performance.

The primary objective of this study is to evaluate the effectiveness of BCH and Hamming error-control coding techniques in improving transmission reliability over noisy communication channels. Specifically, the research investigates their error-correction capability, decoding performance, and overall communication reliability under comparable simulation conditions. The findings are expected to provide useful insights into the selection of suitable channel coding techniques for reliable digital communication systems while balancing error performance, computational complexity, and bandwidth efficiency.

3. Conceptual framework

The primary challenge in designing a digital communication system is to provide a cost-effective means of transmitting information at a rate, reliability level, and quality that satisfy user requirements. This study focuses on the detection and correction of errors in digital information. Such errors commonly arise during the transmission and processing of digital, particularly binary, data due to noise and interference within communication channels. By protecting digital information with an appropriate error-control code, it becomes possible to efficiently detect and correct errors that occur during transmission. As a result, error-control coding has become an essential component of modern information and communication systems and is widely employed across a broad range of digital communication applications.

3.1 Forward error correction (FEC)

Forward Error Correction (FEC) is an error control strategy used in one-way transmission systems. In this approach, error-correcting codes are employed to automatically detect and correct errors at the receiver without requiring retransmission of the transmitted data [13]. Communication systems that utilize FEC cannot request the sender to retransmit corrupted information. Consequently, the entire error control capability of the coding scheme is dedicated to error detection and correction. Since retransmission is not possible in one-way communication systems, error control must be achieved through Forward Error Correction (FEC). A common application of this technique is in deep-space communication systems, where retransmission is impractical because of the significant communication delay. The operation of a Forward Error Correction system is illustrated diagrammatically in Fig 1.

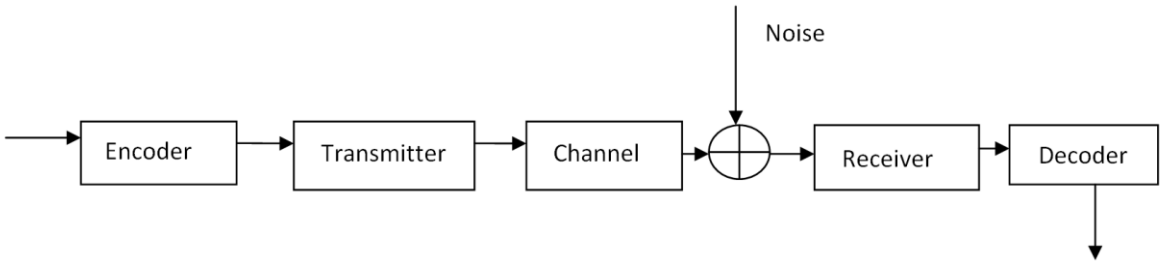


Fig.1 Block diagram of a FEC system

3.2 Automatic repeat request (ARQ)

The primary advantage of Automatic Repeat Request (ARQ) (see Fig 2) over Forward Error Correction (FEC) is that error detection requires considerably simpler decoding equipment than error correction. In addition, ARQ is adaptive because retransmission is requested only when transmission errors are detected [14].

However, when the channel experiences a high error rate, retransmissions become more frequent, reducing system throughput, which is the rate at which newly generated messages are successfully received. Therefore, combining FEC to correct the most common error patterns with ARQ to detect and retransmit less frequent errors provides a more efficient error control strategy than relying on ARQ alone [15].

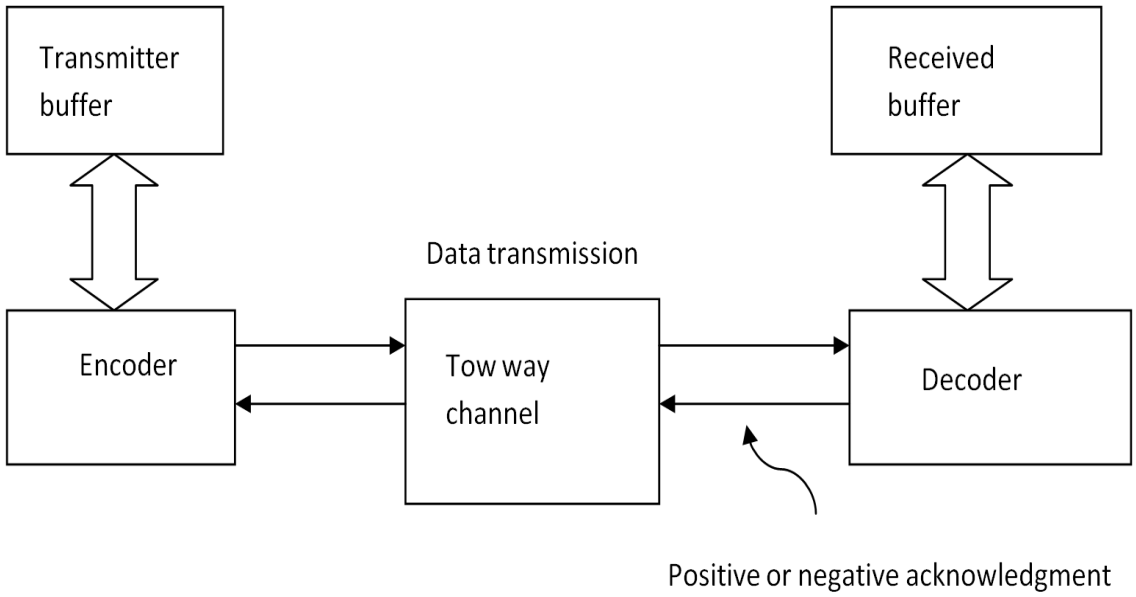


Fig.2 block diagram of an ARQ system

All error-correcting codes are founded on the same fundamental principle of introducing redundancy into the original information to enable the detection and correction of errors that may occur during data storage or transmission. In a practical coding scheme, redundant symbols are appended to the information symbols to form a coded sequence, known as a codeword.

As illustrated in Fig 3, the first k positions of the codeword contain the original information symbols, while the remaining (n − k) positions consist of redundant symbols generated as functions of the information symbols. These redundant symbols provide the additional information required for effective error detection and correction at the receiver.

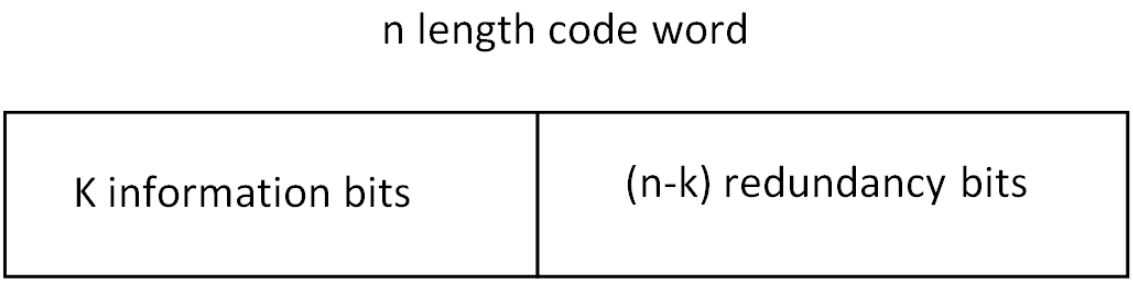


Fig.3 a code word

3.3 How error control codes work

Error control coding improves the reliability of digital communication by intentionally appending redundant bits to the original information before transmission. These additional bits allow the receiver to identify transmission errors and, when the coding scheme permits, recover the original data without requesting retransmission, even when the communication channel is affected by noise [16].

As illustrated in Fig. 4, a block encoder converts an input message containing k information bits into an n-bit codeword, where n > k. For binary block codes, there are 2^k valid codewords among the total 2^k possible binary sequences of length n. During transmission, channel noise may alter one or more bits, producing a received vector that may not correspond to a valid codeword. The decoder therefore estimates the transmitted message by selecting the valid codeword with the smallest distance from the received vector.

The decoding concept can be visualized using the decoding spheres shown in Fig. 4. Each valid codeword occupies the center of a sphere with radius t, representing the maximum number of errors that can be corrected reliably. Adjacent codewords, such as A and B, are separated by the minimum Hamming distance, d_{min}, which is defined as the smallest number of bit positions in which any two valid codewords differ. Increasing the minimum distance enhances the ability of a code to detect and correct transmission errors because it provides greater separation between neighboring codewords.

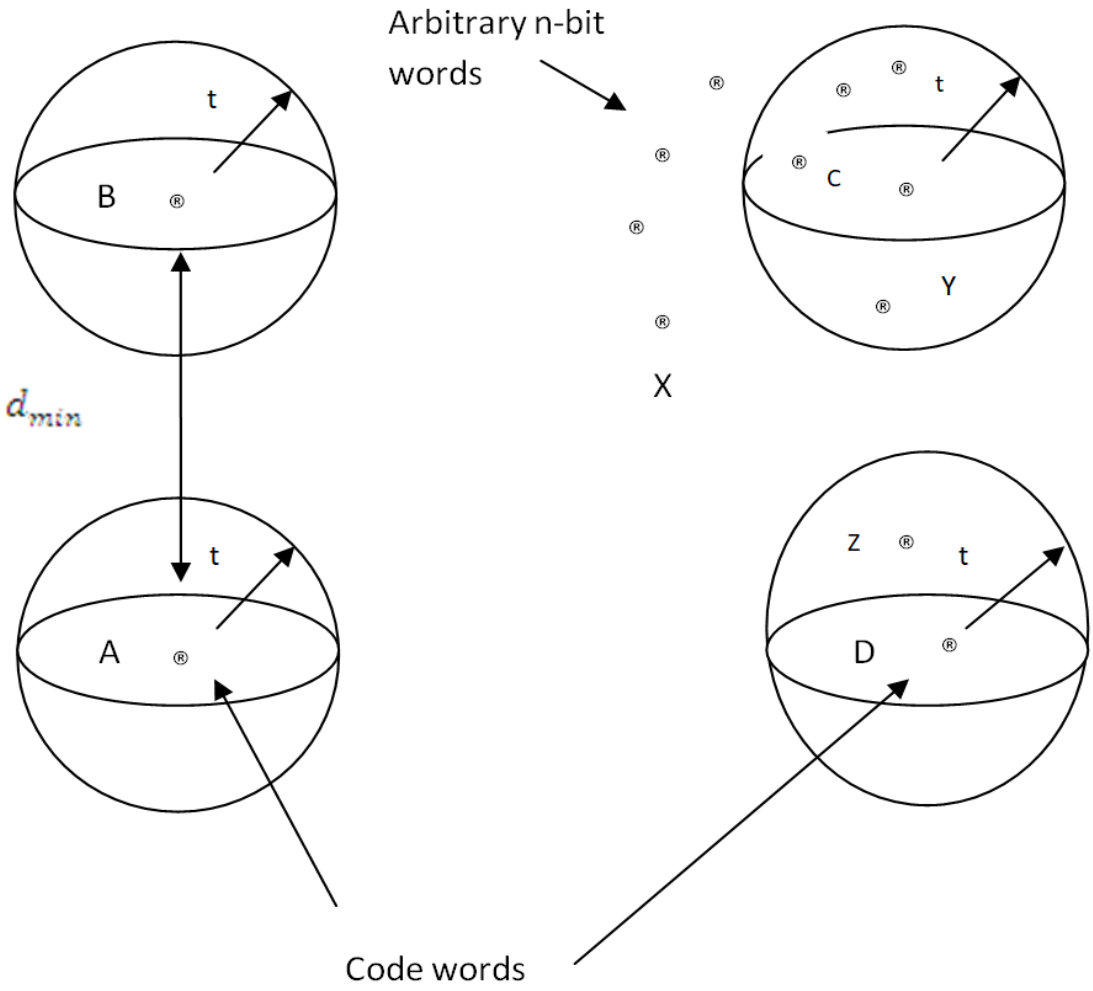


Fig.4 Decoding spheres of radius t

3.4 Error detection

The minimum Hamming distance (d min) determines the error-detection capability of a block code [17], as illustrated in Fig 4. A code can guarantee the detection of every error pattern containing up to u bit errors provided that

$$d_{min} = u + 1.$$

Although some error patterns involving more than u errors may also be detected, only errors up to this limit are guaranteed. For explanation purposes, consider a decoder that stores all 2^k valid codewords in a lookup table. When an n-bit sequence is received, the decoder compares it with the stored codewords. If an exact match exists, the corresponding information bits are accepted as valid; otherwise, the received sequence is identified as containing transmission errors. The error-detection process can be described by the following situations.

Case 1: Error-free transmission.

Suppose codeword C is transmitted and reaches the receiver without alteration. Because the received sequence exactly matches one of the stored valid codewords, the decoder accepts it and correctly recovers the original information.

Case 2: Detectable error.

Assume that codeword C is transmitted, but channel noise changes one or more bits, resulting in the received sequence Y. Since Y does not correspond to any valid codeword, the decoder recognizes that an error has occurred and rejects the received sequence. Although it does not determine the locations or number of erroneous bits, successful error detection has been achieved.

Case 3: Undetectable error.

In rare circumstances, channel errors may transform transmitted codeword C into another valid codeword, D. Because D is a legitimate entry in the codebook, the decoder incorrectly accepts it as the transmitted codeword and delivers the associated information bits. Consequently, the transmission error remains undetected.

3.5 Error correction

The error-correction capability of a block code depends on the minimum distance between valid codewords. As shown in Fig. 4, decoding regions must remain disjoint to avoid ambiguity. This requirement is satisfied when

$$d_{min} = 2t + 1,$$

where t denotes the maximum number of correctable bit errors [4]. Consequently, a code with $d_{min} = 3$ can correct one error, whereas a code with $d_{min} = 5$ can correct two errors. More generally, a code is capable of correcting t errors while simultaneously detecting v additional errors when

$$d_{min} \geq 2t + v + 1.$$

The decoding outcomes of an error-correcting decoder can be classified into three categories.

Case 1: Successful correction.

If codeword C is transmitted and the received vector Y falls inside the decoding sphere centered on C, the decoder identifies C as the most likely transmitted codeword and corrects the altered bits successfully.

Case 2: Incorrect correction.

When transmission errors are sufficiently severe, the received vector Z may lie inside the decoding sphere of another valid codeword D instead of C. The decoder therefore selects D, resulting in an incorrect decoding decision.

Case 3: Error detection without correction.

If the received vector X lies outside the decoding sphere of every valid codeword, the decoder cannot confidently identify the transmitted codeword. Instead of attempting an unreliable correction, it declares that an error has occurred and reports the detection of an uncorrectable transmission error.

4. Mathematical modeling

4.1 Hamming encoder

The Hamming single-bit error correction code is implemented by appending check bits to the output message. Hamming codes constitute the first class of linear codes developed for error correction. These codes, along with their variations, have been widely applied in error control for digital communication and data storage systems [18]. For any positive integer ($m \geq 3$), a Hamming code exists with the following parameters:

Code length: $n= 2^m-1$
Number of information symbols: $k= 2^m-m-1$
Number of parity –check symbols: $n-k=m$
Error-correcting capability : $t=1(d_{min}=3)$

The parity-check matrix H of this code is formed by taking all non-zero m-tuples as its columns. In systematic form, the columns of H are arranged in the following structure:

$$H=[I_m\ Q]$$

Where I_m is an $m \times m$ identity matrix and the submatrix Q consists of 2^m-m-1 columns which are the m-tuples of weight 2 or more . In systematic form, the generator matrix G of the code is given as:

$$G=[Q^T\ I_{2^m-m-1}]$$

Where, QT is the transpose of Q and I_{2^m-m-1} is an $(2^m-m-1) \times (2^m-m-1)$ identity matrix.

The codeword V to be transmitted over the communication channel can be obtained by multiplying the message vector M by the corresponding generator matrix G. Mathematically, this relationship is expressed as V=MG. During transmission, various interferences such as noise may be encountered. As a result, the transmitted information can become corrupted due to the presence of noise in the communication channel.

4. 2 Hamming decoder

The basic task of Hamming decoder is to detect and then correct errors occurring in the transmitted information bits [19]. This is achieved by first computing the syndrome (S) for each received codeword and then matching it with a corresponding lookup table to identify the error location. Once the error positions are determined, a modulo-2 addition is performed between the error location polynomials and the corrupted received information bits (R). This process can be mathematically represented as follows:

$S=RH^T$, where S=syndrome, R=received vectors, and H=parity check matrix

Next, a corresponding error-location lookup table is constructed for all possible syndrome values. After computing the syndrome, the corresponding error-location polynomial is determined by referring to this lookup table $\hat{C}= e \oplus R$, where $\hat{C}$ =correctly received information bit, and e=error location polynomial from the lookup table. Hamming codes can correct all single-error patterns while simultaneously detecting all double-error patterns. When a single error occurs during the transmission of a code vector, the resulting syndrome is non-zero and contains an odd number of ones. In contrast, when double errors occur, the syndrome is also non-zero but contains an even number of ones. Based on these properties, decoding is performed as follows; if the syndrome S is zero, it is assumed that no error has occurred.

If S is non-zero and contains an odd number of ones, a single error is assumed; the corresponding single-error pattern represented by S is then added to the received vector to correct the error. If S is non-zero and contains an even number of ones, an uncorrectable error pattern is declared.

4.3 BCH encoding and BCH decoding

BCH codes are a generalization of Hamming codes, and they can be designed to correct any error pattern [20] of size t. In this sense, the extension of Hamming codes expands the code design from t = 1 (Hamming codes) to codes that support any desired higher value of t (BCH codes). The design approach is based on taking the least common multiple (LCM) of appropriate minimal polynomials. For any positive integer $m \geq 3$ & $t < 2m-1$, there exists a binary BCH code CBCH (n, k) with the following properties:

Code length	$n=2^m - 1$
Number of parity bits	$n-k \leq mt$
Minimum Hamming distance	$d_{min} \geq 2t+ 1$
Error-correction capability	t errors in a received vector

The BCH decoding process has the following properties [21]; If the code word $C(x) = c_0 + c_1x + \ldots\ldots + c_{n-1}\, x^{n-1}$ is transmitted and the transmission errors result in the following received vector $r(x) = r_0 + r_1\, x + \ldots\ldots + r_{n-1}\, x^{n-1}$.

5. The methodology

This study employs a simulation-based experimental approach to evaluate and compare the performance of error-control coding techniques in a digital communication system. The system is modeled using a communication chain consisting of a source, channel encoder, AWGN channel, channel decoder, and destination, and is implemented in MATLAB. Binary information sequences generated at the source are encoded using two error-control coding schemes, BCH and Hamming codes. The encoded bit streams are transmitted through an AWGN channel, where noise is introduced to simulate real-world channel impairments. At the receiver, decoding algorithms are applied to recover the original information.

System performance is evaluated by comparing the transmitted and received sequences and computing the bit error rate (BER) under varying signal-to-noise ratio (SNR) conditions. A comparative analysis of BCH and Hamming codes is conducted to assess their relative error-correction capability and transmission reliability. The complete simulation framework is implemented in MATLAB, and results are presented using graphical visualization to illustrate the effect of noise and the effectiveness of the coding schemes. Fig 5 shows the general system model used in the study.

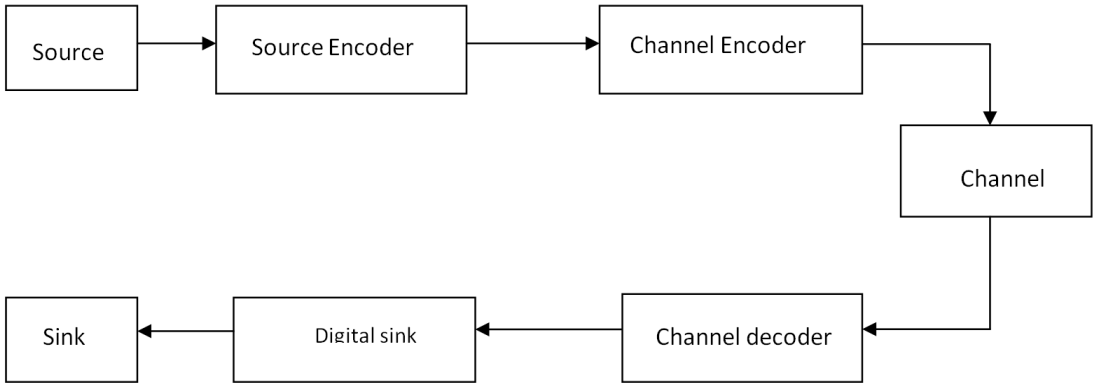


Fig.5 the communication system for source and channel encoder

In conducting this research, the following basic steps will be followed; design of the encoder and decoder, implementation of the system, and performance comparison of the coding techniques used.

5.1 Design of Hamming encoder

The Hamming single-bit error correction code is implemented by appending check bits to the output message according to a specific pattern [22]. Message bits are numbered from left to right starting at 1. Positions that correspond to powers of two (1, 2, 4, 8, …) are assigned as check bits, while the remaining positions (3, 5, 6, 7, 9, …) are used for data bits in sequence.

Each check bit enforces even parity over itself and a specific group of data bits. A data bit belongs to a particular check bit group if the binary representation of its position number contains a 1 in the corresponding check bit position [23]. For example, the data bits associated with check bit 2 include all bit positions whose binary representation has a 1 in the 2's place, such as bits 2, 3, 6, 7, and so on. The fundamental task in this process is the generation of a codeword by combining the source message bits with the redundant parity (check) bits. This represents the first step in implementing error-control coding strategies.

The following Fig 6, based on an 8-bit data message, illustrates this study process. It shows the assignment of data and check bits within a 12-bit output message. The lines beneath the output message indicate the grouping of bits for each parity check, and each group includes a parity computation that determines the corresponding check bit inserted into the final codeword.

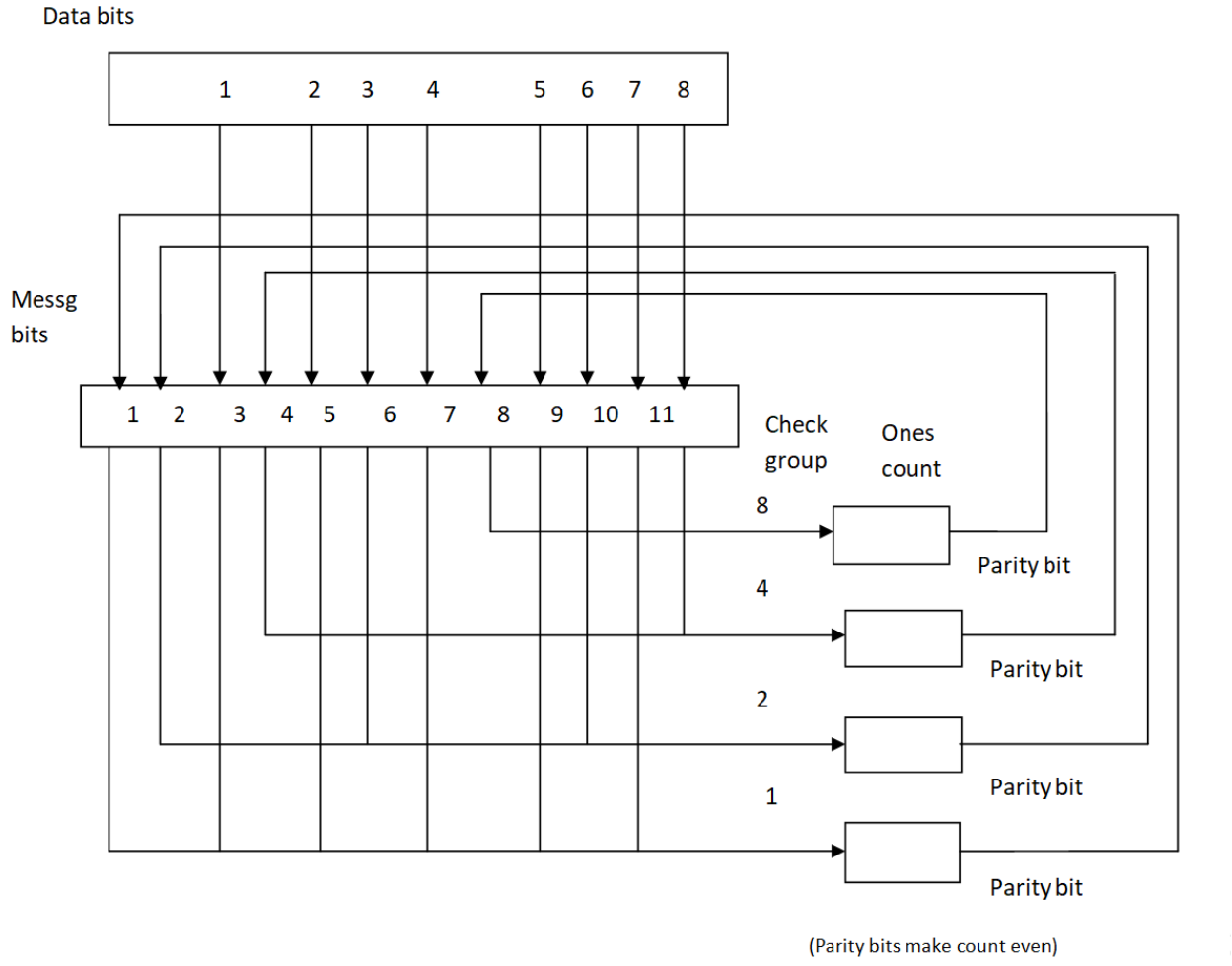


Fig.6 the Hamming code

5.2 Design of BCH encoder

As shown in Figure 7, BCH codes belong to a family of cyclic codes, which provide them with an algebraic structure that simplifies both encoding and decoding procedures. Cyclic codes are a class of error-correcting codes that can be efficiently encoded and decoded using simple shift registers and combinational logic, based on their polynomial representations [24].

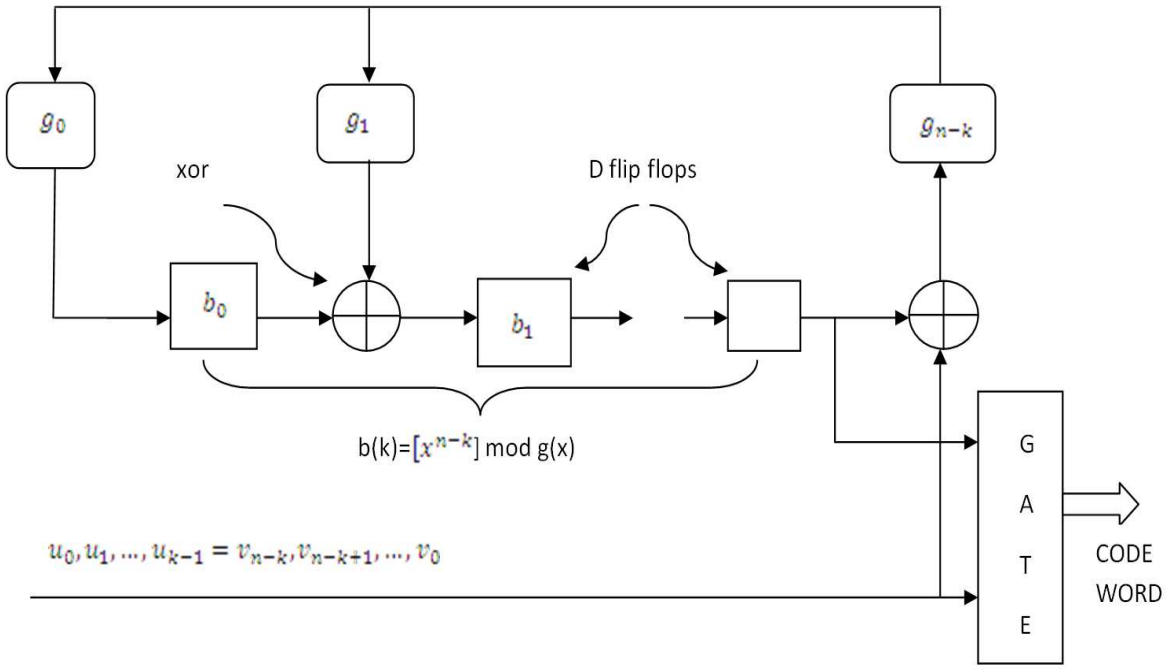


Fig.7 BCH encoder

Where U_r , r = 0, 1. . . k - 1, are the message bits to be encoded. V_r , r = k, k+1... n-1 are the redundant bits. g_r (X) = r=0, 1… n-k are the generator polynomials. The encoding of an (n, k) cyclic code in systematic form consists of three steps. In step one with the gate turned on, the k information digits U_0, U_1,…,U_{n-k-1} shifted into the circuit and simultaneously into the communication channel. Shifting the message U(x) into the circuit from the front end is equivalent to pre multiplying U(x) by X (n-k).

As soon as the complete message has been entered into the circuit, the (n − k) digits stored in the register form the remainder, and thus represent the parity-check digits. In step two it break the feedback connection by turning off the gate. In step three it shift the parity check digits out and send them into the channel.

These n-k parity check digits b_0, b_1, … , b_{n-k-1} together with the information digits form a complete code word. These form the code word b(x) + X^{n-k} U(x). All the three steps can be accomplished with a division circuit, which is a linear (n-k) stage shift register with feedback connection based on the generator polynomial g(x) =1+g_1 x+ g_2 X^2 +. . .+ $g_2X^{(n-k-1)}$ + $X^{(n-k)}$.

5.3 Design of BCH decoder

The design of decoders is more challenging compared to encoder design because it involves both error detection and error correction [25]. The main task is to distinguish the source information signal from the parity bits by detecting and correcting errors in the codeword [26]. Therefore, the decoder is designed in such a way that it can quickly identify and correct erroneous information, enabling the receiver to obtain the correct transmitted codeword. This

represents a fundamental stage in error-control systems. As shown in Fig 8, the decoding procedure consists of three main components: a syndrome register, an error pattern detector, and buffer registers for storing the received vector. The received polynomial is shifted into the syndrome register from the left end. To eliminate the effect of an error digit on the syndrome, the error digit is fed into the shift register from the left end through an exclusive-OR gate. The decoding process proceeds as follows.

In step 1, the syndrome is generated by shifting the entire received vector into the syndrome register, while simultaneously storing the received vector in the buffer register.

In step two the syndrome is read into the detector and is tested for the corresponding error pattern. The detector is a combinational logic circuit which is designed in such a way that its output is 1 if and only if the syndrome in the syndrome register corresponds to a correctable error pattern with an error at the highest order positionx^{n-1}. That is if a '1" appears at the output of the detector, the received symbol in the right most stage of the buffer register is assumed to be erroneous and must be corrected; if the '0" appears at the output of the detector, the received symbol in the right most stage of the buffer register is assumed to be correct and no correction is necessary.

At step three the first received symbol is read out of the buffer. At the same time, the syndrome register is shifted once. If the received symbol is detected to an erroneous symbol, it is then corrected by the output of the decoder. The output of the detector is also feedback to the syndrome register to modify the syndrome. These results in a new syndrome, which corresponds to the altered received vector, shifted one place to the right.

At step four the new syndrome formed in step 3 is used to detect whether or not the second received symbol is an erroneous symbol. The decoder repeats step 2 and 3. The second received symbol is corrected in exactly the same manner as the first received symbol was corrected.

At step five the decoder decodes the received vector symbol by symbol in the manner outlined above until the entire received vector is read out of the buffer register.

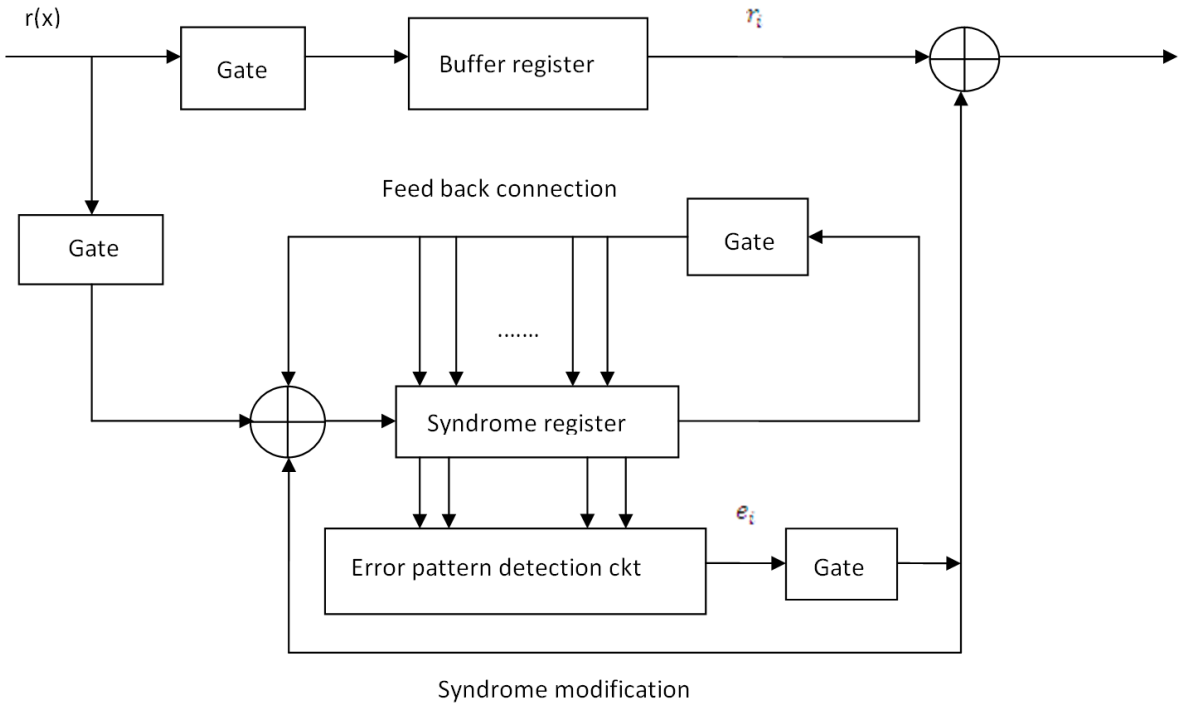


Fig.8 BCH decoder

5.4 The implementation

The implementation is carried out in two stages, without a noisy channel and with a noisy channel. In the case without a noisy channel, the received information is expected to be identical to the transmitted data; that is, the original message is correctly extracted from the codeword without any errors. In the noisy channel case, additive white Gaussian noise (AWGN) is introduced, which may lead to corrupted information at the receiver. In this situation, the transmitted information can be affected, and the decoder is expected to detect and correct errors caused by the presence of AWGN. The performance of the communication system under AWGN is evaluated by comparing results using a bit error rate (BER) versus signal-to-noise ratio (SNR) curve, generated in MATLAB.

The research is implemented using MATLAB software by carrying out each step sequentially until the complete error-control coding system is developed. A graphical user interface (GUI) is also used to demonstrate the performance comparison and to clearly present the operation of the error-control coding system. The main objective at this stage is to develop a better understanding of the algorithms used in error-control coding.

6. Result and discussion

6.1 Hamming code

The results presented below show the performance of Hamming error-control coding in terms of bit error rate (probability of error) versus signal-to-noise ratio under the influence of an AWGN channel. Two cases are considered in error-control coding, the coded condition and the uncoded condition. The coded condition refers to the case in which error-control coding techniques are applied to achieve reliable transmission. The uncoded condition refers to the case in which no error-control coding techniques are used for transmission reliability. From the graphical representation of Hamming code performance, the following observations can be made.

6.1.1 Coded condition

As observed in Fig 9, moving from left to right along the x-axis (signal-to-noise ratio), the probability of error decreases significantly compared to the uncoded case. This indicates that the error detection and correction capability of the Hamming error-control coding technique improves as the signal-to-noise ratio increases. In other words, when the signal-to-noise ratio is high, the signal becomes much stronger than the noise, making it easier to recover the transmitted information.

The Hamming code performs effectively for single-bit errors, as it can detect and correct all such erroneous codewords, which is also demonstrated in this research. However, when two-bit errors are introduced, the code can typically detect them, but its correction capability becomes limited. Furthermore, when more than two-bit errors occur, the code can still detect the presence of errors, but it cannot correct them.

From these observations, it can be concluded that the Hamming error-control coding technique works effectively for single-bit error scenarios, where it can fully detect and correct errors. In this case, it can be considered highly efficient for one-bit error correction and detection. However, for multiple-bit errors, its error-correction performance is not as effective as in the single-bit case. Hamming code works perfect for one bit error and can detect and correct all the erroneous code words, this also demonstrated in our research. But when we introduce two bit errors it can detect but the correction capability is rare. In addition to this when we introduce more than two bit errors we will get the proper detection and but not correction for the introduced errors.

As a result of this we can deduce that the hamming error control coding technique works perfectly that means for one bit error it can detect and correct the entire erroneous bit completely. In this case we can confidentially say that hamming error control coding technique is 100% efficient for one bit error detection and correction. Moreover, more than one bit we can’t get the perfection of error control capability of the hamming code like the one bit error capability.

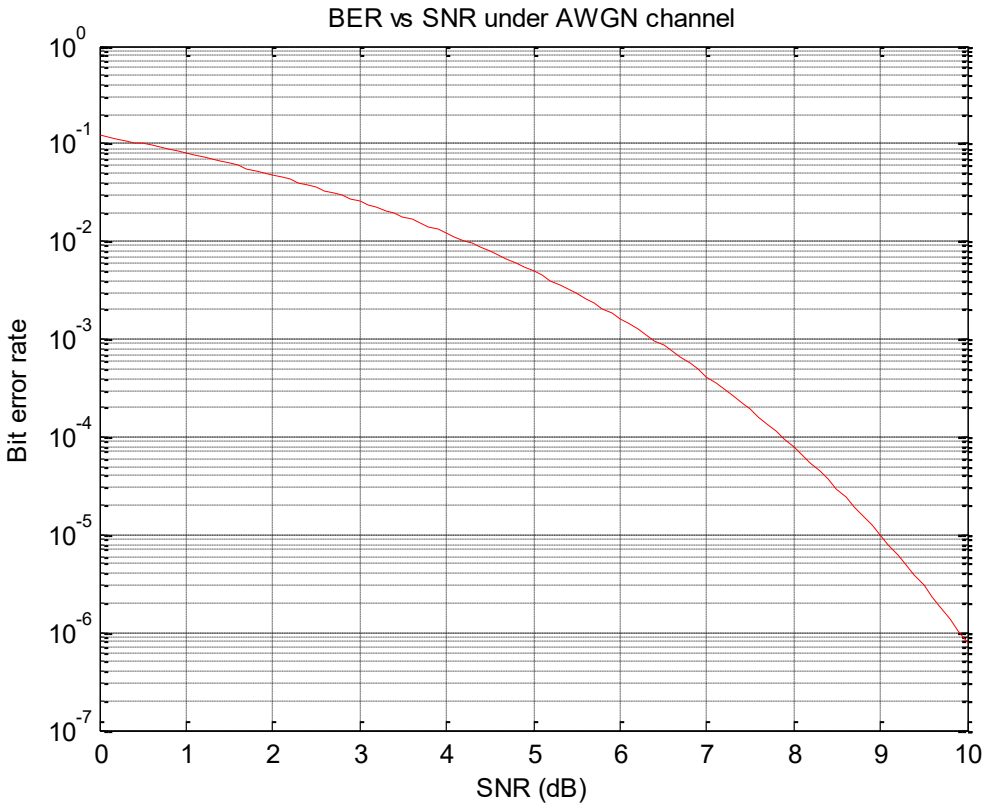


Fig.9 coded condition for hamming error control coding techniques

6.1.2 Uncoded condition

In Fig 10, in the uncoded case the probability of error remains relatively high even at large signal-to-noise ratios. The system performs well only in the absence of significant noise, that is, when the signal-to-noise ratio is very low.

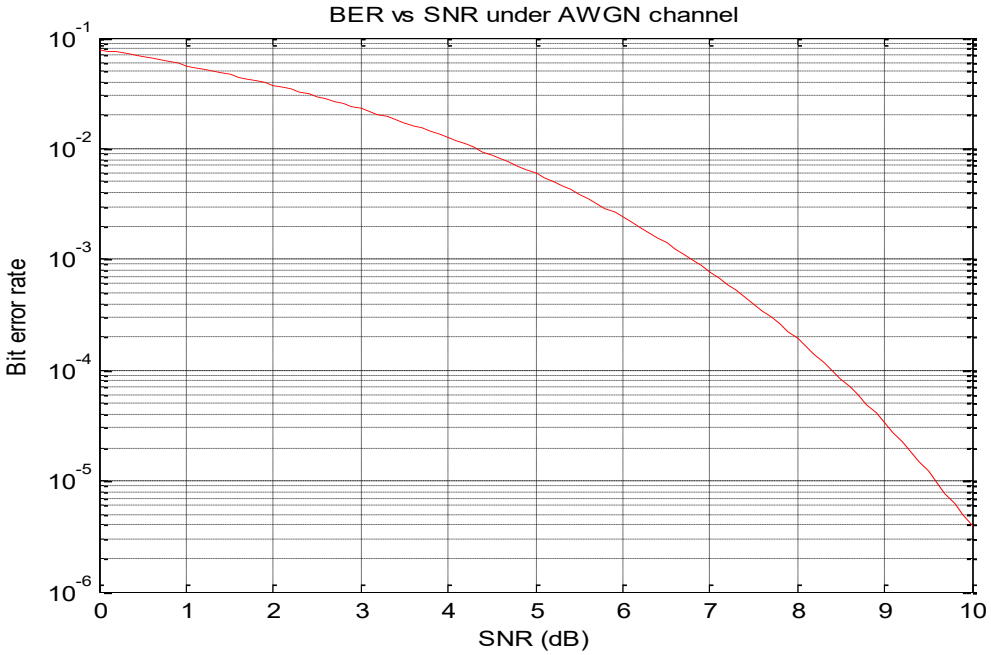
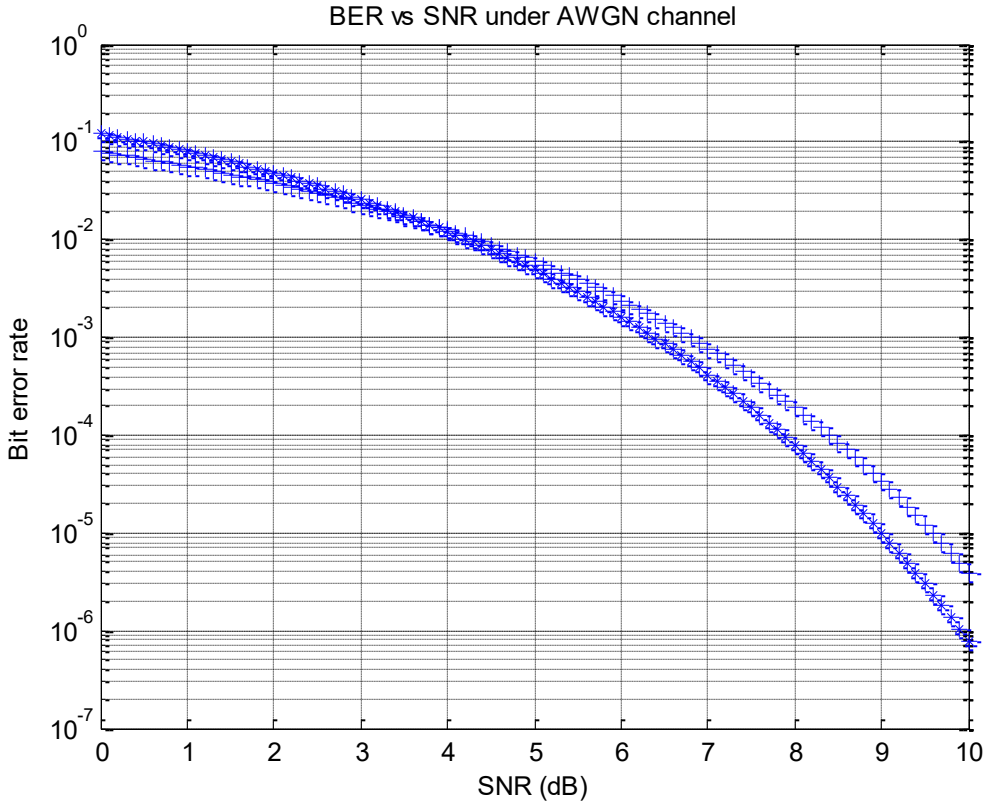


Fig.10 uncoded conditions for hamming error control coding technique

6.1.3 Performance comparison between coded and uncoded condition

As illustrated by the graphical results in Fig 11, the coded condition clearly outperforms the uncoded condition when the signal-to-noise ratio (SNR) is very high (i.e., when the signal strength significantly exceeds the noise). At low SNR levels, the uncoded condition exhibits a lower bit error rate than the coded one. However, the two curves intersect at a specific threshold, beyond which the coded condition achieves a lower probability of error. Consequently, for high-SNR environments, implementing error control coding is highly recommended to correct bit errors and ensure error-free signal reception.



Index: ++++++ Uncoded graph, ******* Coded graph

Fig.11 performance comparison between coded and uncoded conditions for Hamming

6.2 BCH code

For the BCH error control coding techniques we also show the coded and uncoded condition as we saw in the case of hamming error control coding techniques above. The results here under shows the BCH error control coding bit error rate (probability of error) versus signal to noise ratio under the influence of AWGN channel. There are two cases in error control coding; these are Coded condition, and Uncoded condition.

6.2.1 Coded condition

As illustrated by the graphical results in Fig 12, an increasing Signal-to-Noise Ratio (SNR) indicates that signal strength is rising relative to noise power, which subsequently drives down the probability of error. This demonstrates that the detection and correction capabilities of the BCH error control coding technique improve as the SNR increases. Consequently, at higher SNRs, the signal significantly outperforms the noise, allowing for much easier reproduction.

We have developed a BCH code capable of detecting and correcting up to three-bit errors within a transmitted binary signal. The error control performance of this BCH technique improves as the number of error bits per codeword decreases. This capability can be categorized into three distinct performance scenarios:

At simple case (one bit error), we introduced a one bit error in all the transmitted information code words and the BCH error control coding technique which can detect and correct above half of the erroneous transmitted code words.

In moderate case (two bits error), we introduced two bits error in all the transmitted information code words and BCH technique which can detect and correct almost half of the erroneous transmitted code words. Hence, we can surely say that the BCH technique is better for one bit error detection and correction than the two bit error.

At the worst case (tree bit error), we introduced three-bit errors, the system reaches the maximum error-correcting limit of our design. In this scenario, the technique detects and corrects fewer than 50% of the corrupted codewords, typically averaging around 40%.

Generally, these observations demonstrate that as the number of error bits per codeword increases, the total percentage of successfully detected and corrected codewords decreases.

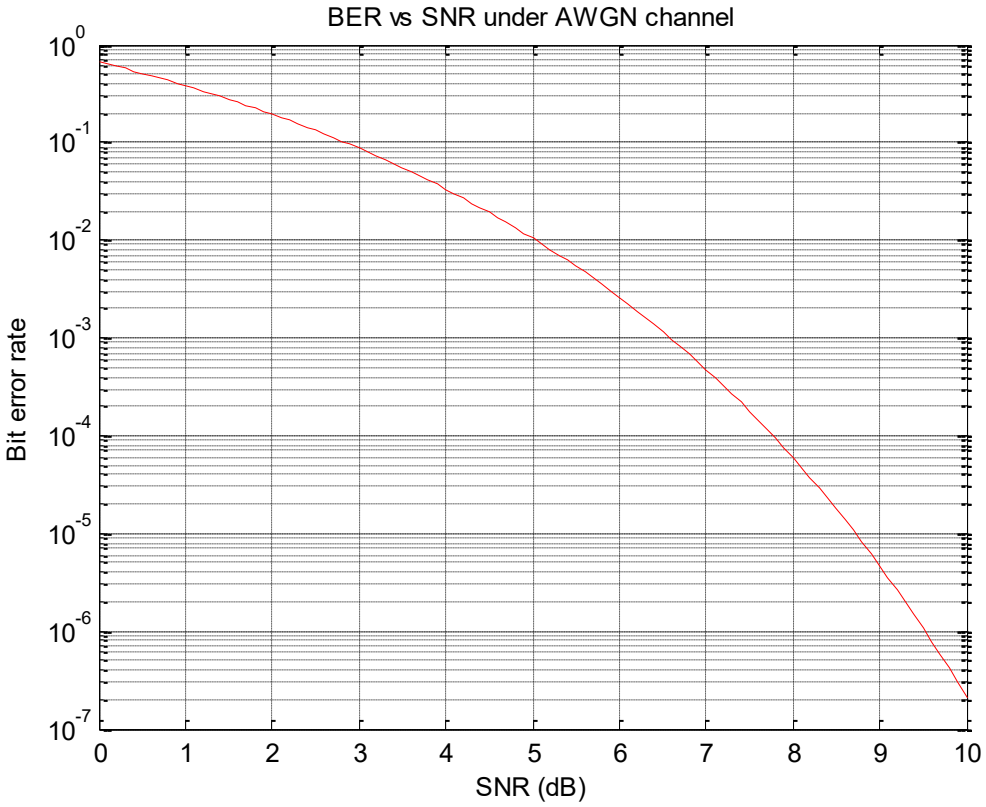


Fig.12 coded condition for BCH error control coding technique

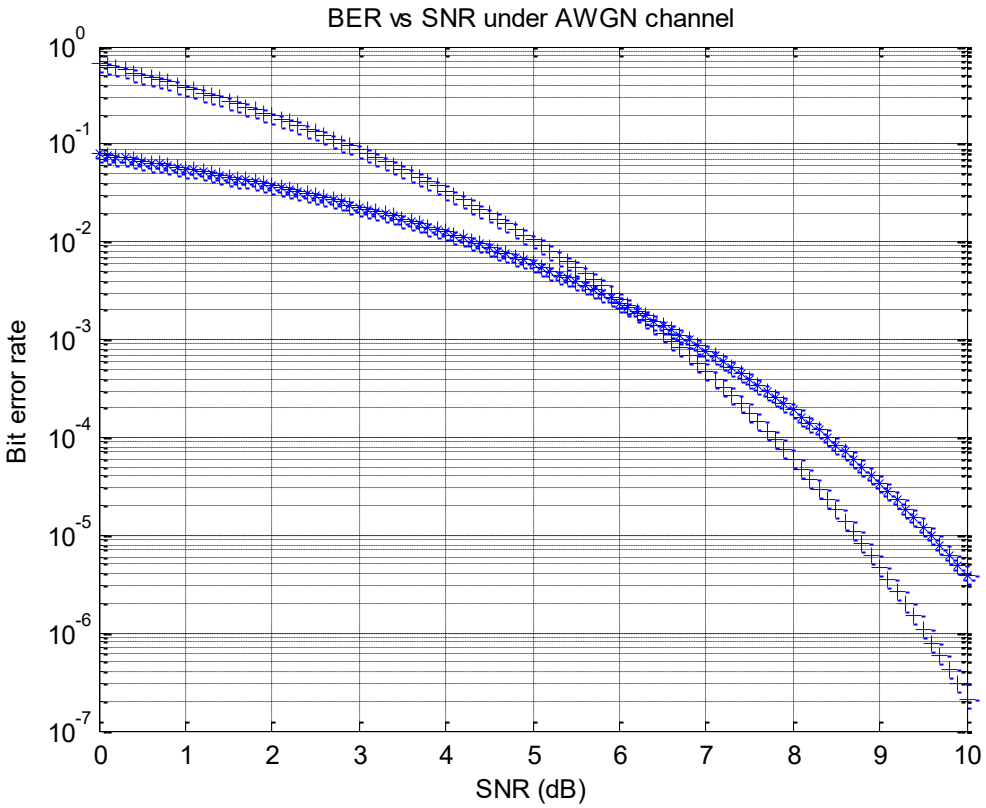
In the uncoded condition, because no error control coding technique is applied, the performance follows the same pattern as the uncoded condition graph for the Hamming case shown in Fig 10.

6.2.2 Performance comparison between coded and uncoded conditions

We can now compare the error detection and correction capabilities of the BCH code against an uncoded system as the signal-to-noise ratio increases. The graphical results in Fig 13 clearly demonstrate that the coded system outperforms the uncoded one under high signal-to-noise ratio conditions, where the signal strength significantly exceeds the noise.

When we compare the coded and uncoded at SNR=0dB, we see that the uncoded condition has small probability of error than the coded. The probability of error variation between the coded and uncoded condition in BCH is larger than that of the hamming code. However, starting at SNR=6dB, the two curves (coded and uncoded) intersect. This crossover point is higher than that of the Hamming code case due to the large initial variation in error probability between the coded and uncoded conditions, as described above.

Beyond SNR=6dB, the coded system achieves a lower probability of error than the uncoded one. Consequently, we can deduce that at high SNR, it is advisable to employ error control coding to correct erroneous bits and receive the transmitted signal without corruption.



Index: ++++++ Uncoded graph, ******* Coded graph

Fig.13 performance comparison between coded and uncoded condition for BCH

6.3 Performance comparison between Hamming and BCH error control coding

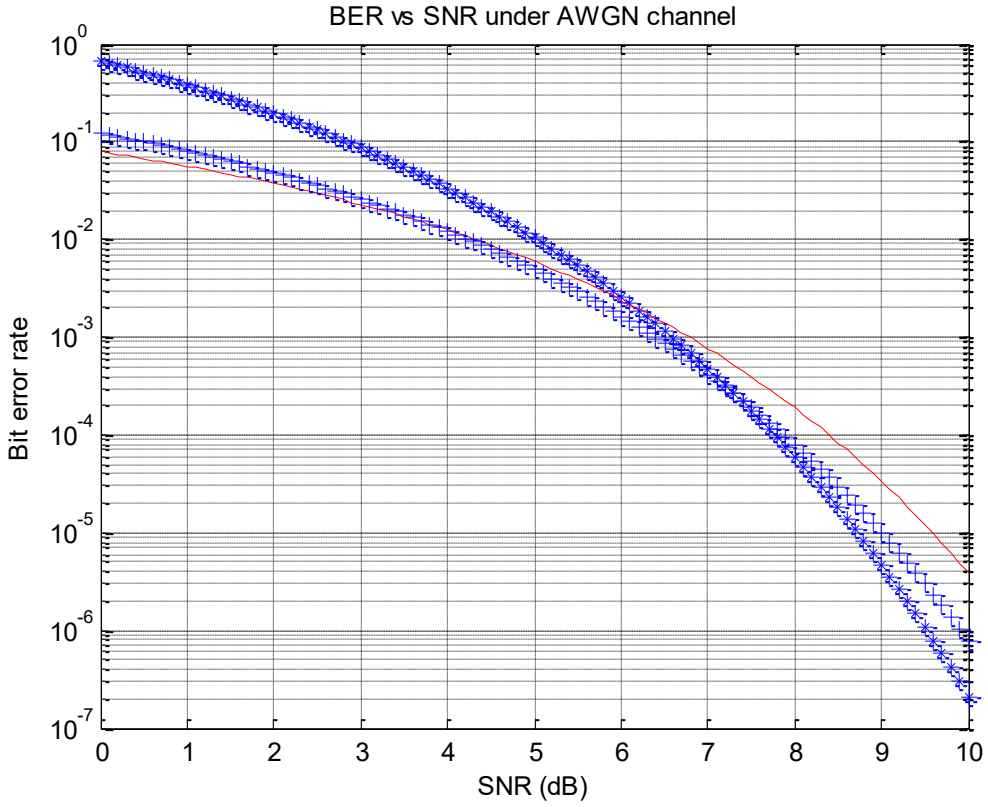
As illustrated in Fig 14, a comparison between the Hamming and BCH error control coding techniques reveals distinct trade-offs in performance. For single-bit errors, the Hamming code demonstrates perfect detection and correction capabilities, completely fixing all corrupted received codewords. In contrast, the designed three-bit error-correcting BCH code successfully detects and corrects just over 50% of these single-bit erroneous codewords.

When handling two-bit errors, the Hamming code can detect the errors but fails to correct them properly; meanwhile, the BCH code successfully detects and corrects two-bit errors, maintaining a capability of around 40%. Consequently, the BCH technique is preferable to the Hamming code under two-bit error conditions.

For errors exceeding two bits, the Hamming code remains capable of detection but cannot properly correct the data. On the other hand, the BCH code continues to detect and correct these higher-order bit errors, though its efficiency drops below the levels achieved during one- and two-bit error conditions.

This indicates that as the number of error bits in a received codeword increases, the corrective effectiveness of the BCH technique decreases, whereas the Hamming code fails to function as required under these advanced error conditions.

Ultimately, while a Hamming code is strictly limited to a single-bit error detection and correction design, the BCH technique offers the flexibility to be engineered for various error-correcting capacities, such as the three-bit capability designed for this case, allowing it to outperform the Hamming code in multi-bit error scenarios. It is possible to design the BCH error control coding technique with different error detecting and correcting capability, for our case it is three error detecting and correcting capability and works better for $SNR \geq 7$ than that of hamming, but for hamming it is only possible to design with one error detecting and correcting capability.



*Index: ****** indicates BCH graph, ++++ indicates Hamming graph, ------- indicates uncoded graph*

Fig.14 performance comparison between BCH and Hamming error control coding techniques

6.4 The graphical user interface (GUI)

Developed for demonstration purposes, the graphical user interface (GUI) in Fig 15 and Fig 16 allows users to input the number of message bits to generate and the number of errors to introduce. It then displays the generated, received, and corrected codewords. Additionally, the interface tracks and displays critical error-correction statistics, including the number of codewords corrected, detected but not corrected, and miscorrected, alongside the total messages sent, messages corrected, and an overall BER versus SNR graph.

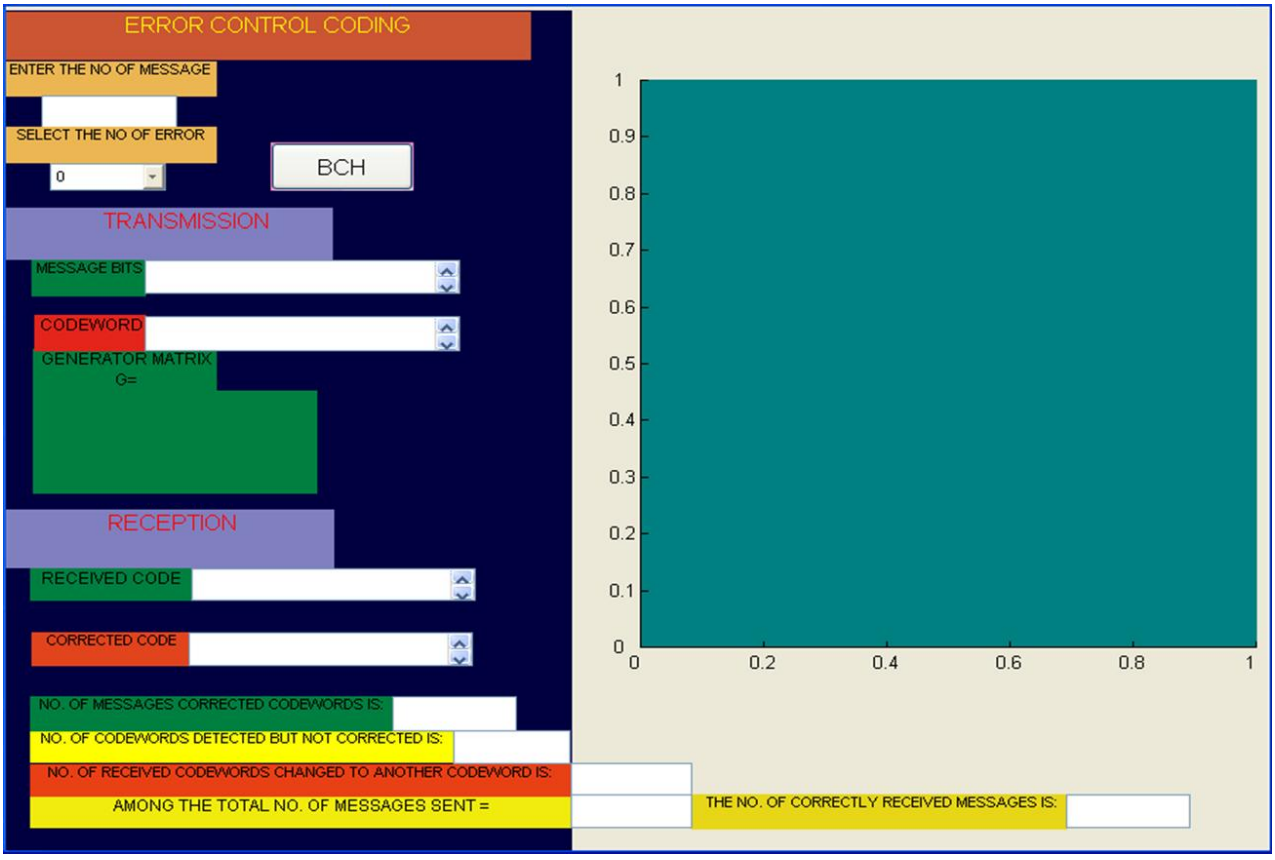


Fig.15 GUI for BCH ECC technique

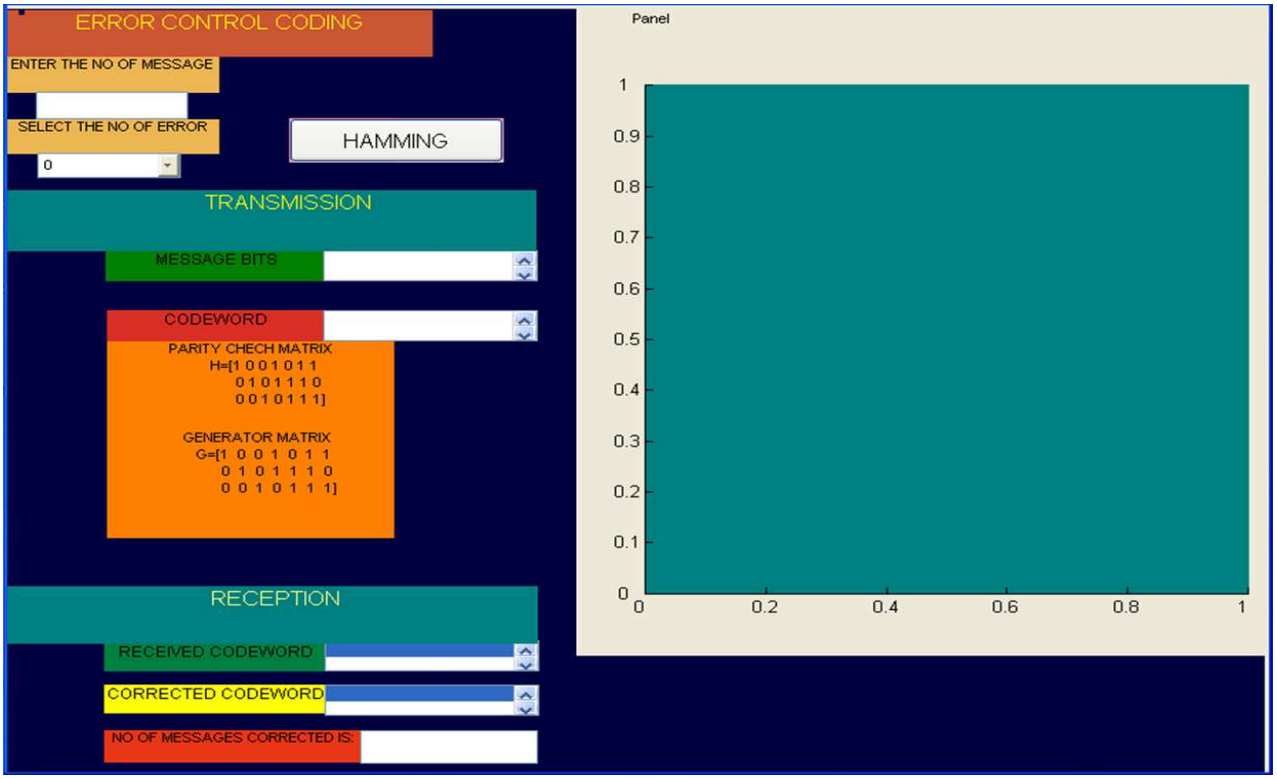


Fig.16 GUI for Hamming ECC technique

As illustrated in Fig 16, the developed GUI for the Hamming code system guides the user through a comprehensive simulation process. It prompts the user to specify the number of message bits to be generated and the number of errors to be introduced, displaying both inputs accordingly. Additionally, the GUI presents the generated message bits, the corresponding generated codeword, and the received codeword. Finally, it displays the corrected codeword along with the total count of successfully corrected messages.

7. Conclusions

This study underscores the critical role of error control coding (ECC) techniques in ensuring reliable information transmission and storage. By mitigating the disruptive effects of channel noise, the strategic design and selection of ECC schemes are paramount to maintaining data integrity. While various coding techniques offer distinct error detection and correction capabilities governed by the volume of bit errors introduced during transmission, our comparative analysis highlights a significant performance divergence between Hamming and Bose-Chaudhuri-Hocquenghem (BCH) codes. Although Hamming codes provide mathematically optimal error detection and correction for single-bit errors, their efficacy degrades in multi-bit error environments. In contrast, the implemented BCH codes demonstrate superior versatility, successfully detecting and correcting multiple random bit errors tailored to specific design parameters. Crucially, our empirical findings indicate that while the BCH scheme cannot rectify all corrupted transmissions, it achieves a robust error correction throughput, successfully recovering 35% to 50% of all corrupted codewords. These results demonstrate that BCH codes offer a highly viable and scalable solution for high-reliability communication systems operating over noisy channels. To mitigate the disruptive effects of channel noise on transmitted signals, it is highly recommended that communication frameworks, particularly long-distance links, integrate optimized ECC techniques. The expanding footprint of digital networks in strategic sectors, such as military operations, satellite systems, and radio transmissions, underscores the necessity for robust error-correction capabilities. Implementing advanced ECC methodologies is critical to overcoming inherent channel impairments, thereby guaranteeing reliable, secure, and resilient global communication.

9. Accordance

All procedures were conducted in accordance with the research ethics of Aksum University 2018 standards.

10. Competing interests

We declare that all the authors have no competing interests.

11. Declarations

11.1 Ethical approval

Ethical approval for this study was obtained from the research ethics committee of Aksum University, Ethiopia (Reference No. AKU-AIT:01/229/2026). The study involved MATLAB simulation and GUI application, and did not involve human participants, animals, or personal data. Therefore, informed consent was not required. The research was conducted in accordance with the ethical standards of Aksum University 2018.

11.2 Consent to participate: Not Applicable

11.3 Consent to publication: All authors have read, approved and consented to its publication

12. Data availability

All data generated and analyzed during the research are included within this paper and Science Data Bank with a full name of the database “Error Control Coding for Digital Communication Systems Using BCH and Hamming Techniques”, and DOI https://doi.org/10.57760/sciencedb.41396 or link https://cstr.cn/31253.11.sciencedb.41396.

13. Funding: Not Applicable

14. Author contribution

Dawit Teklu Weldeslasie and Mengsteab Dagnew Welemariam drafted the design, developed the simulation, and selected the analytical tools and methodology. All authors conducted the experiments and gathered the data. Zenawi Haileslassie Berhe and Mengsteab Dagnew Welemariam managed and organized the research database. Dawit Teklu Weldeslasie developed the GUI application and drafted the initial manuscript. All authors critically reviewed, edited, and provided final approval for the published version.

References

[1]. Huang, M., Lu, X., An, C. *et al.* Computational wiretap coding: framework and practical construction. *Cybersecurity* 9, 42 (2026). https://doi.org/10.1186/s42400-025-00419-9
[2]. Djordjevic, I.B. (2025). Information Theory and Coding Fundamentals. In: Physical-Layer Security, Quantum Key Distribution, and Post-Quantum Cryptography. Springer, Cham. https://doi.org/10.1007/978-3-031-88372-9_2
[3]. Chen, W., Qin, Z. (2026). Joint Source and Channel Coding. In: Semantic Communications. Textbooks in Telecommunication Engineering. Springer, Cham. https://doi.org/10.1007/978-3-032-11005-3_4
[4]. Pande, M., Kulkarni, A.J., Shastri, A.S. (2025). Error Correction Coding. In: Pande, M., Kulkarni, A.J., Shastri, A.S. (eds) Optimization Methods in Mobile Communication Systems. Springer, Singapore. https://doi.org/10.1007/978-981-95-1810-4_5
[5]. Stuber, G. (2026). Error Control Methods. In: Introductory Digital Wireless Communications. Springer, Cham. https://doi.org/10.1007/978-3-032-01559-4_8
[6]. Chen, W., Qin, Z. (2026). Channel Adaptive Semantic Communication. In: Semantic Communications. Textbooks in Telecommunication Engineering. Springer, Cham. https://doi.org/10.1007/978-3-032-11005-3_6
[7]. Rajabi, M.E., Khaleghi Bizaki, H. & Shafiee, M. Adaptive Wireless Covert Communication with Full/Half-Duplex Transceiver using Artificial Noise. *Arab J Sci Eng* **50**, 11333–11346 (2025). https://doi.org/10.1007/s13369-024-09800-1
[8]. Henkel, W. (2026). Channels, Channel Models, and Metrics. In: Channel Coding. Signals and Communication Technology. Springer, Cham. https://doi.org/10.1007/978-3-032-08236-7_1
[9]. Pajovic, M. (2026). Introduction to Digital Communication Systems. In: Fundamentals of EMI Phenomena in Radio Communication Systems. Springer, Cham. https://doi.org/10.1007/978-3-031-88748-2_4
[10]. Wang, X., Zhou, J. BCH codes with explicit minimum distance and applications in quantum synchronizable codes. *Cryptogr. Commun.* 17, 1427–1443 (2025). https://doi.org/10.1007/s12095-025-00815-5
[11]. Efanov, D. Special Aspects of Using Hamming Codes in the Synthesis of Self-Checking Digital Devices Based on the Data Inversion Method. *Russ Microelectron* 54, 819–826 (2025). https://doi.org/10.1134/S1063739725700210
[12]. Al-Hajm, S.Y.S., Develi, I. & Kockaya, K. Hybrid deep learning for robust spectrum sensing over AWGN and fading channels. *EURASIP J. Adv. Signal Process.* 2026, 5 (2026). https://doi.org/10.1186/s13634-025-01276-x
[13]. Abbas, K. (2023). Forward Error Correction. In: From Algorithms to Hardware Architectures. Springer, Cham. https://doi.org/10.1007/978-3-031-08693-9_6
[14]. Faruque, S. (2021). Automatic Repeat Request (ARQ). In: Free Space Laser Communication with Ambient Light Compensation. Springer, Cham. https://doi.org/10.1007/978-3-030-57484-0_2
[15]. Bulo, Y., Sonalika, A. & Krishna, S.M. Adaptive Automatic Repeat Request (AdARQ) Protocol to Improve the Throughput Characteristic of the Time-Varying Wireless Channel. *J. Inst. Eng. India Ser. B* 105, 53–61 (2024). https://doi.org/10.1007/s40031-023-00937-5
[16]. Krishna, Hari, et al. *Computational Number Theory and Digital Signal Processing: Fast Algorithms and Error Control Techniques*. CRC press, 2026.
[17]. Masanti, C., Witschel, HF., Riesen, K. (2025). Automated Error Detection Through Specialized Task Implementation. In: Wallraven, C., Liu, CL., Ross, A. (eds) Pattern Recognition and Artificial Intelligence. ICPRAI 2024. Lecture Notes in Computer Science, vol 14893. Springer, Singapore. https://doi.org/10.1007/978-981-97-8705-0_12
[18]. San-José, R. About the generalized Hamming weights of matrix-product codes. *Comp. Appl. Math.* 44, 186 (2025). https://doi.org/10.1007/s40314-025-03141-x
[19]. Lavanya G., Sharma, A. Hamming weight enumerators and hulls of linear codes with new parameters associated with simplicial complexes. *Cryptogr. Commun.* (2025). https://doi.org/10.1007/s12095-025-00850-2
[20]. Yao, Y., Li, Y. (2024). An Efficient and Fast Hardware Implementation of Parallel BCH Decoder. In: S. Shmaliy, Y. (eds) 8th International Conference on Computing, Control and Industrial Engineering (CCIE2024). CCIE 2024. Lecture Notes in Electrical Engineering, vol 1252. Springer, Singapore. https://doi.org/10.1007/978-981-97-6934-6_65
[21]. Yu, H., Liu, M. & Pang, B. A class of BCH codes of length . *Comp. Appl. Math.* 45, 331 (2026). https://doi.org/10.1007/s40314-026-03732-2
[22]. Nagarathna, H., Ramesh Kumar, K. Implementation of improved hamming code for reliable CAN protocol on FPGA. *Discov Electron* 3, 57 (2026). https://doi.org/10.1007/s44291-026-00206-x
[23]. Marani, V., Reddy, A.B.C., Kavitha, S., Reniwal, B.S. (2025). Approach Toward In-Memory Computing-Based Hamming Code Implementation For Error Correction and Detection. In: Gupta, A., Pandey, J.G., Chaturvedi, N., Dwivedi, D. (eds) VLSI for Embedded Intelligence. VDAT 2023. Lecture Notes in Electrical Engineering, vol 1210. Springer. https://doi.org/10.1007/978-981-97-3756-7_27
[24]. Du, KL., Swamy, M.N.S. (2026). Channel Coding. In: Wireless Communication Systems. Signals and Communication Technology. Springer, Cham. https://doi.org/10.1007/978-3-032-13094-5_13
[25]. Yao, Y., Li, Y. (2024). An Efficient and Fast Hardware Implementation of Parallel BCH Decoder. In: S. Shmaliy, Y. (eds) 8th International Conference on Computing, Control and Industrial Engineering (CCIE2024). CCIE 2024. Lecture Notes in Electrical Engineering, vol 1252. Springer, Singapore. https://doi.org/10.1007/978-981-97-6934-6_65
[26]. Liu, Y., Li, J., Wang, H. *et al.* A BCH error correction scheme applied to FPGA with embedded memory. *Front Inform Technol Electron Eng* 22, 1127–1139 (2021). https://doi.org/10.1631/FITEE.2000323