Prashant M. Pawar ·
R. Balasubramaniam ·
Babruvahan P. Ronge ·
Santosh B. Salunkhe · Anup S. Vibhute ·
Bhuwaneshwari Melinamath *Editors*

# Techno-Societal 2020

## Proceedings of the 3rd International Conference on Advanced Technologies for Societal Applications—Volume 1

Springer

Prashant M. Pawar · R. Balasubramaniam ·
Babruvahan P. Ronge · Santosh B. Salunkhe ·
Anup S. Vibhute · Bhuwaneshwari Melinamath
Editors

# Techno-Societal 2020

## Proceedings of the 3rd International Conference on Advanced Technologies for Societal Applications—Volume 1

Springer

*Editors*
Prashant M. Pawar
SVERI's College of Engineering
Pandharpur, Maharashtra, India

Babruvahan P. Ronge
SVERI's College of Engineering
Pandharpur, Maharashtra, India

Anup S. Vibhute
SVERI's College of Engineering
Pandharpur, Maharashtra, India

R. Balasubramaniam
Precision Engineering Center
Bhabha Atomic Research Centre
Mumbai, Maharashtra, India

Santosh B. Salunkhe
SVERI's College of Engineering
Pandharpur, Maharashtra, India

Bhuwaneshwari Melinamath
SVERI's College of Engineering
Pandharpur, Maharashtra, India

ISBN 978-3-030-69920-8 ISBN 978-3-030-69921-5 (eBook)
https://doi.org/10.1007/978-3-030-69921-5



This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

# Contents

# Properties of Extended Binary Hamming [8, 4, 4] Code Using MATLAB

**N. S. Darkunde, S. P. Basude, and M. S. Wavare**

**Abstract** The main aim of this paper is to study various properties of extended binary Hamming [8, 4, 4] code, when we know its generator matrix. Using MATLAB, we can study syndrome decoding, weight of a codeword, error correction and error detection of binary Hamming [8, 4, 4] code.

**Keywords** Linear code · Generator matrix · Parity check matrix · Hamming code · Syndrome decoding

## 1 Introduction

In the late 1940s, Claude Shannon has started development of information theory and coding theory as a mathematical model for communication. At the same time, R. Hamming found a necessity for error correction in his work on computers. Already Parity checking was used to detect errors in the calculations of the relay-based computers of the day, and Hamming realized that a more sophisticated pattern of parity checking allowed the correction of single errors along with the detection of double errors. The codes that Hamming put forth, were important for theoretical and practical reasons. In [1, 2], algorithmic approach for error correction has been studied for few codes.

Each binary Hamming code [3] has minimum weight and distance 3 and that of extended binary Hamming code has weight and distance as 4. In this paper, we are going to study some properties of extended binary Hamming [8, 4, 4] code.

N. S. Darkunde (✉) · S. P. Basude
School of Mathematical Sciences, SRTM University, Nanded, Maharashtra, India

M. S. Wavare
Department of Mathematics, Rajarshi Shahu Mahavidyalaya, Latur, Maharashtra, India



P. M. Pawar et al. (eds.), *Techno-Societal 2020*,
https://doi.org/10.1007/978-3-030-69921-5_68

## 2 Preliminaries

### 2.1 Linear Codes [4]

Let $Fq$ denote the finite field with q elements, where $q$ is some power of a prime. A linear $[n, k]_q$—code is a k-dimensional subspace of $F_q^n$. The parameters $n$ and $k$ are referred to as the length and dimension of the corresponding code.

***Example*** The subset C = {000, 001, 010, 011} of vector space $F_2^3$ is $[3, 2]_2$ linear code. Similarly, C = {0000, 1100, 2200, 0001, 0002, 1101, 1102, 2201, 2202} is $[4, 2]_3$ linear code.

### 2.2 Hamming Distance and Hamming Weight [5]

Let $x, y \in F_q^n$, The Hamming distance from $x$ to $y$, denoted by $d(x, y)$, is defined to be the number of places at which $x$ and $y$ differ.

***Example*** Consider $x = 01010$, $y = 01101$, $z = 11101$ in $F_2^5$, Then $d(x, y) = 3$, $d(x, z) = 4$. For any $x \in F_q^n$, the support of $x$, denoted by supp($x$), is defined to be the set of nonzero coordinates in $x = (x_1 x_{2,} x_3, \ldots, x_n)$, that is $supp(x) := \{i : x_i \neq 0\}$.

For a $[n, k]_q$ code C containing at least two words, the nonnegative integer given by $\min\{d(x, y) : x, y \in C, x \neq y\}$ is called minimum distance of C. It is denoted by $d(C)$.

***Example*** For a code C = {0000, 1000, 0100, 1100} in $F_2^4$, we see that $d(C) = 1$.

**Definition 1** [3] Let $u$ be a positive integer. A code $C$ is $u$—error-detecting if, whenever a codeword incurs at least one but at most u errors, the resulting word is not a codeword.

A code is exactly $u$—error-detecting if it is $u$—error-detecting but not $(u + 1)$ error-detecting.

***Example*** Consider C = {000000, 000111, 111222} $\subseteq F_2^6$. This code is 2-error-detecting, because changing any codeword in one or two positions does not result in another codeword. In fact, $C$ is exactly 2-error-detecting, as changing each of the last three positions of 000000 to 1 will result in the codeword 000111(so $C$ is not 3-error-detecting).

**Theorem 1** [3] *A code C is u—error-detecting if and only if* $d(C) \geq u + 1$, *that is, a code with distance d is an exactly* $(d - 1)$*—error-detecting code.*

**Definition 2** [3] Let $v$ be a positive integer. A code $C$ is $v$—error-correcting if minimum distance decoding is able to correct $v$ or fewer errors, assuming that the incomplete decoding rule is used. A code $C$ is exactly $v$—error-correcting if it is $v$—error-correcting but not $(v + 1)$—error-correcting.

***Example*** Consider $C = \{000, 111\}$ in $F_2^3$. It is easy to see that, $C$ is 1-error-correcting.

**Theorem 2** [3] *A code C is v -error-correcting if and only if* $d(C) \geq 2v + 1$, *that is a code with distance d is an exactly* $\lfloor (d-1)/2 \rfloor$*—error correcting code, where* $\lfloor x \rfloor$ *denote the greatest integer less than or equal to* $x$.

**Definition 3** (*Dual Code*) [3] Given a $[n, k]_q$ code C in $F_q^n$, the subspace

$$C^{\perp} = \{x = (x_1, x_2, x_3, \ldots, x_n) \in F_q^n : x.c = 0 \,\text{for all}\, c = (c_1, c_2, \ldots, c_n) \in C\},$$

We have following properties of $C$ and $C^{\perp}$.

1. $|C| = q^{\dim(C)}$, i.e. $\dim(C) = log_q|C|$;
2. $C^{\perp}$ is a linear code and $\dim(C) + \dim(C^{\perp}) = n$;
3. $(C^{\perp})^{\perp} = C$.

## 2.3 Generator Matrix and Parity-Check Matrix

**Definition 4** [5] A matrix $G$ of order $k \times n$ is said to be the generator matrix for a $[n, k]_q$ code $C$, if its rows form basis for $C$.

**Definition 5** [4] A parity-check matrix $H$ for a code $C$ is a generator matrix for $C^{\perp}$.

## 2.4 Extended Binary Hamming [8, 4, 4] Code

Let $G = (I_4|A)$ where $I_4$ is a $4 \times 4$ identity matrix and $A$ is a $4 \times 8$ matrix given by

$$\text{A} = \begin{pmatrix} 10000111 \\ 01001011 \\ 00101101 \\ 00011110 \end{pmatrix}$$

The binary linear code generated by matrix $G$ is called extended binary Hamming [8, 4, 4] code.

### *2.5 Syndrome Decoding [3]*

An efficiency of decoding technique works well, when length $n$ of a given code is small, but it can take a more time when, $n$ is very large, so this time can be saved by using the syndrome to identify the coset from which the word is taken. In [3], the procedure of syndrome decoding has been demonstrated.

Step 1: Let $w$ be received word in the transmission and for this received word $w$, first compute the syndrome of w denoted by Syn(w) which is given by, $syn(w) = wH^T$, where $H$, is parity check matrix of a given code.

Step 2: After constructing Syndrome look up table, we will find the coset leader $u$ next to the syndrome, $syn(w) = syn(u)$.

Step 3: Finally decode the received word $w$ as $v = w - u$.

Now, let us study the properties of extended binary Hamming [8, 4, 4] code using MATLAB [6].

## 3 Properties of Extended Binary Hamming [8, 4, 4] Code Using MATLAB

Communication fails due to the error in the communication channel. In this process, receiver receives the original message with error in it. If the system knows how much error has come with original message then that error can be removed. It is difficult task to correct the errors in the communication; hence we have used MATLAB to accomplish this task.

### *3.1 MATLAB Program for Syndrome Decoding Using Extended Binary Hamming [8, 4, 4] Code*

```
% Hamming Code
n = 8; k =4; %Length and Dimension
 clc % Clearscreen
 I=eye(4); % Identity matrix of order 4
 X=[ 0 1 1 1; 1 0 1 1; 1 1 0 1; 1 1 1 0 ];
G=[I  X] % Generator matrix of extended binary Hamming [8, 4] code
 H=mod([-X' eye(4)],2) % Denote H as a parity Check Matrix for extended
binary Hamming Code
 d = gfweight(G) % Distance of Code
 slt= syndtable(H); % Produce Syndrome look up table.
w = [1 0 0 1 1 1 1 1 ] % Message received
```

```
syndrome = rem(w * H',2);
syndrome_de = bi2de(syndrome,'left-msb'); % Convert to decimal.
 disp(['Syndrome = ',num2str(syndrome_de),
 ' (decimal), ',num2str(syndrome),' (binary)'])
corrvect = slt(1+syndrome_de,:)
% Correction vector % Now compute the corrected codeword.
 correctedcode = rem(corrvect+w,2)
G = 1 0 0 0 0 1 1 1
0 1 0 0 1 0 1 1
0 0 1 0 1 1 0 1
0 0 0 1 1 1 1 0
H =
0 1 1 1 1 0 0 0
1 0 1 1 0 1 0 0
1 1 0 1 0 0 1 0
1 1 1 0 0 0 0 1
d =   4
Single-error patterns loaded in decoding table. 7 rows remaining.
2-error patterns loaded. 0 rows remaining.
w = 1 0 0 1 1 1 1 1
Syndrome = 6 (decimal), 0 1 1 0 (binary)
corrvect =1 0 0 0 0 0 0 1
correctedcode =
0 0 0 1 1 1 1 0
```

## *3.2 MATLAB Program for Encoding Message Using Extended Binary Hamming [8, 4, 4] Code*

```
% Matlab Programme for Encoding of the information message
Clc
 I=eye(4); % Identity matrix of order 4
X=[ 0 1 1 1; 1 0 1 1; 1 1 0 1; 1 1 1 0 ];
G=[I X] % Generator matrix of extended binary Hamming [8, 4] code
H=mod([-X' eye(4)],2) % Parity Check Matrix for extended binary
Hamming Code
u=input('Enter the message bit of length 4 for which you want to enode using
extended binary Hamming Code=' )% input message of length 4
v=mod(u*G ,2) % Encoding of message u
Synv=mod(v*H',2) % Syndrome of enoded messeage
G =
```

```
1 0 0 0 0 1 1 1
0 1 0 0 1 0 1 1
0 0 1 0 1 1 0 1
0 0 0 1 1 1 1 0
H =
0 1 1 1 1 0 0 0
1 0 1 1 0 1 0 0
1 1 0 1 0 0 1 0
1 1 1 0 0 0 0 1

Enter the message bit of length 4 for which you want to enode using
extended binary Hamming Code= [1 0 0 1]
u =1 0 0 1
v =1 0 0 1 1 0 0 1
Synv =0 0 0 0
```

## 4 Conclusion

Once we know the generator matrix of any linear code, using MATLAB we can have many of the things that can be discussed. In this paper, we have studied extended binary Hamming $[8, 4, 4]$ code with the help of MATLAB. The parity check matrix in standard form, for extended binary Hamming $[8, 4, 4]$ code is calculated and using it, syndrome of a particular codeword is calculated. We have encoded the message and decoded it correctly by using MATLAB. Using MATLAB, it has been verified that, the distance of an extended binary Hamming code is 4 and it is 1-error-correcting code.

## References

1. Patil AR, Darkunde NS (2019) Algorithmic approach for error-correcting capability and decoding of linear codes arising from algebraic geometry. In: Information and communication technology for competitive strategies. Springer, Berlin, pp 509–517
2. Patil AR, Darkunde NS (2017) On some error-correcting linear codes. Asian J Math Comput Res 56–62
3. Ling S, Xing C (2004) coding theory-a first course, 1st edn. Cambridge University Press
4. Huffman WC, Pless V (2010) Fundamentals of error-correcting codes. Cambridge University Press
5. MacWilliams FJ, Sloane NJA (1977) The theory of error-correcting codes. Elsevier
6. Pratap R (2017) Getting started with MATLAB: a quick introduction for scientists and engineers, 7e. Oxford University Press.