Linear Codes Capable of Detecting and Correcting Errors

A Project Report
Submitted By

Ritika Verma

in partial fulfilment of the requirements
for the award of degree of

M.Sc. (Mathematics and Computing)

Under the Supervision of

Dr Ritu Arora
Associate Professor, Department of Mathematics
Janki Devi Memorial College
University of Delhi

Department of Mathematics and Computing
Indian Institute of Technology , Guwahati
Assam-781039

July 26, 2022

Certificate

from institution

Acknowledgement

It is my privilege to express my sincerest regards to my project supervisor Associate Professor Dr.Ritu Arora for her valuable guidance, encouragement, whole-hearted cooperation and constructive criticism throughout the duration of my project. I deeply express my sincere thanks to Principal ma'am, Prof. Swati Pal for encouraging me and allowing me to present the project on the topic "Linear Codes Capable of Detecting and Correcting Errors" under the Department of mathematics, Janki Devi Memorial college, University of Delhi. It helped me doing a lot of research and I came to know about so many new things. I take this opportunity to thank Indian Institute of Technology, Guwahati in allowing me to take this project. I pay my respect and love to my parents and all other family members and friends for their love and encouragement throughout, in completing this project.

Contents

Acknowledgements **ii**

1 Introduction **1**
1.1 Rings . . . 1
1.2 Ideal . . . 3
1.3 Residue Classes . . . 3
1.3.1 Residue class ring: . . . 3
1.4 Polynomial Rings . . . 5
1.5 Galois Field . . . 6

2 The Coding Problem **9**
2.1 The Communication Channel . . . 9
2.2 Observations on Error-detecting and Error-Correcting Codes . . . 10
2.3 Type of Codes . . . 11
2.4 Block Codes . . . 12

3 Linear Block Codes **15**
3.1 Linear Block Codes,Generator Matrices, Parity-Check Martix . . . 15
3.2 The Hamming Metrics . . . 16
3.3 Generator Matrix . . . 17
3.4 Parity Check Matrix . . . 19
3.5 The Standard Array and Syndrome . . . 22
3.6 Decoding of Linear Block Code . . . 26

Bibliography **27**

Chapter 1

Introduction

1.1 Rings

Definition : Let $(R, +, .)$ be an algebraic structure, then R is said to be a ring if :

1. $(R, +)$ is an abelian group.
2. $(R, .)$ is a semi-group.
3. $a.(b + c) = a.b + a.c \quad \forall\ a, b \in R$
   $(a + b).c = a.c + b.c \quad \forall\ a, b \in R$

Properties of Rings :

1. **Zero of Ring:** Let $(R, +, .)$ be a Ring. An element $a \in (R, +)$ is called the zero element of the ring $(R, +, .)$ if $a + b = b + a = b \quad \forall\ b \in (R, +)$. Zero element of Ring is denoted be 0.
2. **Unity of Ring:** Let $(R, +, .)$ be a Ring. An element $1 \in (R, +, .)$ if exists is said to be unity if $a.1 = 1.a \quad \forall a \in (R, +, .)$.
3. **Unit elements of Rings:** Let $(R, +, .)$ be a Ring with **unity**. An element a in $(R, +, .)$ is said to be a unit if $\exists$ b $\in (R, +, .)$ such that $a.b = b.a = 1$.
4. **Commutative Ring:** Let $(R, +, .)$ be a Ring. If $a.b = b.a \quad \forall\ a, b \in R(, +, .)$, then $R(, +, .)$ is called a Commutative Ring.
5. **Group of units of Rings:** Let $(R, +, .)$ be a Ring with **unity** then the collection of all units is a group under multiplication called group of units of $(R, +, .)$ and is denoted by $U(R)$.
6. **Idempotent elements of Ring:** Let $(R, +, .)$ be a Ring. An element $a \in (R, +, .)$ is said to be Idempotent element if $a^2 = a$.
7. **Nilpotent elements of Ring:** Let $(R, +, .)$ be a Ring. An element $a \in (R, +, .)$ is said to be Nilpotent element if $\exists\ k \in \mathbb{N}$ *such that* $a^k = 0$.
8. **Trivial Ring :** $(0, +, .)$ is known as trivial ring.

Examples :

Rings	**Zero**	**Unity**	**Units**	**Commutative**
$(\mathbb{Z}_n, +_n, .n)$	0	1	$\mathbb{Z}^*$	Yes
$(\mathbb{Q}_n, +_n, .n)$	0	1	$\mathbb{Q}^*$	Yes
$(\mathbb{R}_n, +_n, .n)$	0	1	$\mathbb{R}^*$	Yes
$(\mathbb{C}_n, +_n, .n)$	0	1	$\mathbb{C}^*$	Yes
$P(\mathbb{N}, \triangle, \cap)$ i.e Power Set of natural numbers	$\emptyset$	$\mathbb{N}$	$\mathbb{N}$	Yes
$M_2(\mathbb{R})$	$[0]_2$	I_2	$GL(2, \mathbb{R})$	No
$M_2(\mathbb{Z})$	$[0]_2$	I_2	$GL(2, \mathbb{Z})$	No
$\mathbb{Z}[\iota]$=$\{a+b\iota : a, b \in \mathbb{Z}\}$	0	1	$\{\iota, -\iota, 1, -1\}$	Yes
$\mathbb{Z}[\sqrt{5}\iota]$=$\{a + b\iota : a, b \in \mathbb{Z}\}$	0	1	$\{1, -1\}$	Yes

Zero Divisors : Let $(R, +, .)$ be a Ring and $0 \neq a \in (R, +, .)$ is said to be a Zero Divisor if $\exists\ 0 \neq b \in (R, +, .)$ such that $a.b = 0$ *or* $b.a = 0$

Integral Domain : A commutative Ring with unity$(C.R.U)$ without zero divisors is defined as Integral Domain.

Division Ring: Let $(R, +, .)$ be a Ring with unity. If every non-zero element is a unit then the Ring is called Division Ring.

Field : A commutative Division Ring is said to be a Field.

Examples:

Integral Domain	$(\mathbb{Z}_p, +_p, .p)$, p is prime	$(\mathbb{Z}, +, .)$	$(\mathbb{R}, +, .)$
Division Ring	$(\mathbb{R}, +, .)$	$(\mathbb{C}, +, .)$	–
Field	–	–	–

- **Note:** A field is always an integral domain but converse may not be true. e.g : $(\mathbb{Z}, +, .)$
- A finite Integral Domain is a Field.

- A finite Division Ring is a Field.

Characteristic of Ring : Let $(R, +, .)$ be a ring. If $\exists\ n \in \mathbb{N}$ such that $n.a = 0\ \forall a \in (R, +, .)$ then we say $(R, +, .)$ is of finite (non -zero) characteristic n and it is denoted by $charR$ *or* chR.
If no such n exists, then we say characteristic of $(R, +, .)$ is infinite(Zero).

Remark. If $(R, +, .)$ is a ring with unity (1)having characteristic m then for a $\in$ $(R, +, .)$:

$$\begin{aligned} 1.a &= a \\ m.a &= m.1.a \\ m.a &= 0 \qquad (\because\ 1\ \textit{has characteristic}\ m\ ) \end{aligned}$$

examples :
1)$P(\mathbb{N}, \triangle, \cap)$ (Table - 1.1) has characteristic 2.
2)$(\mathbb{Z}_{10}, +_{10}, \cdot_{10})$ has characteristic 10.

1.2 Ideal

Definition: Let $(R, +, .)$ be a Ring and $\emptyset \neq I \subseteq (R, +, .)$, then I is said to be an Ideal of Ring $(R, +, .)$ if :

1. $a - b \in I \qquad \forall a, b \in I$
2. $r.a\ ,\ a.r \in I \qquad \forall a \in I, r \in (R, +, .)$

It is denoted by I $\trianglelefteq (R, +, .)$
For any Ring $(R, +, .)$, $\{0\}$ and $(R, +, .)$ are improper Ideals.

1.3 Residue Classes

Definition : Let $(R, +, .)$ be a Ring and I be an ideal of $(R, +, .)$. The cosets of I by the elements in $(R, +, .)$ are called Residue Classes of I.

Let us consider an ideal I of ring $(R, +, .)$ such that :
$I = \{0\ ,\ i_1\ ,\ i_2\ ,\ i_3,\ \ldots\}$
if $r_1 \in (R, +, .)$ *and* $r_1 \notin I$ then residue class of I by r_1 is :
$\{r_1 + 0, r_1 + i_1, r_1 + i_2, r_1 + i_3,\ \ldots\}$
If $r_1 \in I$ then residue class of I by r_1 is same as that of ideal I.

1.3.1 Residue class ring:

Definition: The residue classes of a ring with respect to an ideal form a ring. This ring is called residue class ring.

Theorem 1. *Every residue class modulo m conatins either 0 or a positive interger less than m. Zero is an element of the ideal, and each positive integer less than m is in a distinct residue class.*

Proof. $(R, +, .)$ is a ring. Suppose that I is an Ideal of $(R, +, .)$.
Let s be any element in a residue class of the Ideal I. Now, by using *Euclidean Division algorithm* we can write

$$s \ = \ mq \ + \ r \ , \qquad\qquad where \ 0 \leq r < m$$

$\Rightarrow s$ is in the residue class of r.
$\Rightarrow \{r\} \ \ = \{s\}$
$\Rightarrow r \ + I \ = \ s \ + I$
$\Rightarrow r \ - \ s \in \ I$
$\Rightarrow r \ - \ s$ is a multiple of m
if $r \neq s$, clearly they could not both be less than m and nonnegative.
$\Rightarrow \{0\}, \{1\}, ..., \{m-1\}$ include each residue class once and only once.

□

Theorem 2. *The residue class ring modulo m is a field if and only if m is a prime number.*

Proof. Firstly, we will show that the residue class ring modulo m is a field then m is a prime number by contradiction.
$(R, +, .)$ is a ring. Suppose that I is an Ideal of $(R, +, .)$.
Suppose that m is not prime.
$\Rightarrow m \ = \ rs$ for some integers r and s that are not multiples of m.
now, $\{m\}$ i.e., residue class of I by m, so $\{m\} = \{0\}$
$\Rightarrow \{rs\} \ = \ \{0\}$
$\Rightarrow \{r\}\{s\} \ = \ \{0\}$ (using cosets properties)
$\Rightarrow \{r\}^{-1}\{r\}\{s\} \ = \ \{r\}^{-1}\{0\}$, if $\{r\}$ has an inverse.
$\Rightarrow \{s\} \ = \ \{0\} \Rightarrow s$ is a multiple of m. But, This contradicts the fact that integer s is not a multiple of m.
Hence, $\{r\}$ has no inverse.
$\Rightarrow$ The residue class ring is not a field ($\because$ in a field every non zero element has an inverse.)

Now, we will show that if m is a prime then residue class ring is a field. So, it is sufficient to show that every non-zero element in the residue class ring has an inverse.
by, using (theorem-1) we know that, every residue class contains an integer s such that $0 \leq s < m$.
Suppose $s > 1$ ($\because$ 1 has an inverse). Clearly, as m is prime number by assumption and $m > s$, so greatest common divisor of m and s is 1.
$\Rightarrow am \ + \ bs \ = \ 1$ (using properties of gcd)
$\Rightarrow \{am\} \ + \ \{bs\} \ = \ \{1\}$ ($\because \{am + bs\} = \{am\} \ + \ \{bs\}$)
$\Rightarrow \{bs\} \ = \ \{1\}$ ($\because \{am\} = \{0\}$)
$\Rightarrow \{b\}\{s\} \ = \ \{1\}$
$\Rightarrow \{s\}$ has $\{b\}$ as it's inverse. Hence, every non-zero element has an inverse.

Remark. The residue class ring modulo p , where p is a prime, is a field known as prime field, or **Galois Field** of p elements, $GF(p)$.

□

1.4 Polynomial Rings

Definition: Let $(R, +, .)$ be a ring, then an expression of the form

$$p(x) = a_n x^n + a_{n-1} x^{n-1} + ... + a_1 x^1 + a_0$$

where $a_0, a_1, ...a_n \in (R, +, .)$ is called a polynomial over $(R, +, .)$. The collection of all polynomial over $(R, +, .)$ with respect to usual addition and multiplication of polynomials called ring of polynomials over $(R, +, .)$.
It is denoted by $R[x]$.

Irreducible Polynomials: A polynomial $p(x)$ of degree n which is not divisible by any polynomial of $0 \leq$ degree $<$ n is called irreducible.

Greatest common divisor : The greatest common divisor of two polynomials is the monic polynomial of greatest degree which divides both of them.

Remark. Two polynomials are said to be relatively prime if their greatest common divisor is 1.

A set of polynomials is an ideal if and only if it consists of all multiples of some polynomial. Thats is, the ring of polynomial is a principal ideal ring. The ideal which consists of all multiples of $f(x)$ is denoted by $(f(x))$. The residue class ring formed from this ideal is called the ring of polynomials modulo f(x).
We can also observe that the residue class modulo a polynomial $f(x)$ of degree n contains either 0 or a polynomial of degree less than n. Zero residue class modulo a polynomial $f(x)$ is same as that of ideal. Every polynomial of degree less than n is in distinct residue class.

Remark. The residue classes of polynomials modulo a polynomial $f(x)$ of degree n form a commutative linear algebra of dimension n over the coefficient field.

We can note that the above vector space can be mathematically represented as:

$$V = \{t(x) |\ t(x)\ is\ r(x)\ modulo\ f(x) for\ some\ polynomial\ r(x)\ of\ deg\ m,\ m\ \in \mathbb{N}\}$$

under poynomial addition and multiplication over a field F. That is,
Any polynomial in V is of the form:
$\{a_{n-1}x^{n-1} + ... + a_1 x^1 + a_0\} = a_{n-1}\{x^{n-1}\} + ... + a_1\{x^1\} + a_0\{1\}$
$\Rightarrow \{1\}, \{x^1\}, ..., \{x^{n-1}\}$ spans the vector space V.

Also the set $\{\{1\}, \{x^1\}, ..., \{x^{n-1}\}\}$ is linearly independent.
($\because$ *let* a_0 , a_1 , ..., a_{n-1} be elements in field, then consider:
$a_{n-1}\{x^{n-1}\} + ... + a_1\{x^1\} + a_0\{1\} = 0$
$\Rightarrow a_0 = a_1 = , ..., a_{n-1} = 0)$

$\Rightarrow V$ has dimension n.

Every Ideal, which is a subspace of the vector space V formed by the residue classes of polynomials modulo a polynomial $f(x)$ of degree n, has a generator polynomial $g(x)$ that divides $f(x)$. Also, every monic polynomial that divides $f(x)$ generates a different ideal of V. Every residue class in the ideal generated by $g(x)$ contains a unique polynomial that is divisible by $g(x)$ and has degree less than n.

1.5 Galois Field

Theorem 3. *Let $p(x)$ be a polynomial with coefficient in a field F, iff $p(x)$ has no factors with coefficient in F, then the algebra of polynomials over F modulo $p(x)$ is a field.*

Proof. Let $f(x)$ belong to $F[x]$ with degree less than that of $p(x)$. Since $p(x)$ is irreducible so this implies that $f(x)$ and $p(x)$ are relatively prime.
So there exist polynomials $q(x)$ and $r(x)$
s.t.
$f(x)q(x) + p(x)r(x) = 1$.
Therefore $f(x)q(x) = 1(modp(x))$.
So $q(x)$ is multiplicative inverse of $f(x)$ mod $p(x)$ but note that $q(x)$ may not be in $F[x]$ mod $p(x)$.
So let $m(x)$ be remainder when $q(x)$ is divided by $p(x)$.
Then $q(x) = p(x)q_1(x) + m(x)$
So $degm(x) < degp(x)$
And $f(x)m(x) = 1$ mod $(p(x))$
So $m(x)$ is multiplicative inverse of $f(x)$ and it belongs to $F[x]$ mod $(p(x))$ Since $f(x)$ is arbitrary so F mod p is a field

□

The vector space formed by the residue classes of polynomials modulo an irreducible polynomial $p(x)$ of degree k in F is a field. Let us denote this field by K. This field is called an extension field of degree k over F($\because$ the field generated is a $F-$ vector space of dimension k.)
It is represented by :
K
| k
F
The residue class containing x is $\{x\} \in K$. The extension field by $\{x\}$ is denoted as $F[\{x\}]$. The original field F is called a ground field. The field $F[\{x\}]$ is the smallest field containing both F and $\{x\}$. As $p(\{x\}) = 0$ ($\because p(\{x\}) = a_k\{x^k\} + a_{k-1}\{x^{k-1}\} + ... + a_1\{x^1\} + a_0\{1\}$) $\Rightarrow p(\{x\}) = \{a_kx^k + a_{k-1}x^{k-1} + ... + a_1x^1 + a_0\}$. As R.H.S is a polynomial of degree of degree k, hence R.H.S is $\{0\}$ i.e $0 \Rightarrow \{x\}$ is a root of of $\{x\}$, and it's is said that the extension field is obtained by adjoining a root of $p(x)$ to the ground field.
We have already seen that residue classes of integrs modulo any prime number p form a field of p elements called Galois field $GF(p)$ (by using remark 1.3.1). The ring of polynomials over any finite field has at least one irreducible polynomials of every degree.
The field formed by taking polynomials over the galois field $GF(p)$ modulo an irreducible polynomial $g(x)$ of degree m is called Galois Field of p^m elements, or $GF(p^m)$. By (theorem-1.4) the new field formed is a vector space of dimension m over the field $GF(p)$ and hence has p^m elements. Every finite field has the same structure as some galois field and differs only in the way the elements are named. The field $GF(p^m)$ has a characteristic p.

Theorem 4. *In a field of characteristic p,* $(a + b)^p = a^p + b^p$.

Proof. Let F be field with characteristic p. Consider $a, b \in F$. Consider :
$(a + b)^p = a^p + \binom{p}{1}a^{p-1}b + \binom{p}{2}a^{p-2}b^2 + \ldots + b^p$

All the binomial coefficient $\binom{p}{i}$ for $0 < i < p$ have p as a factor and as the characteristic of the field is p, so all the binomial coefficent are zero (by using remark 1.1).
$\Rightarrow (a + b)^p = a^p + b^p$. □

Now, consider a ground field F and an extension field of F, and let β be any element of the extension field. The monic polynomial $m(x)$ of smallest degree with coefficients in the ground field F such that $m(\beta) = 0$, then $m(x)$ is called the minimum polynomial or minimum function of β.

Theorem 5. *The minimum function $m(x)$ of any element β is irreducible.*

Proof. We will prove it by contradiction.
Suppose that the minimum function is reducible.
$\Rightarrow m(x) = m_1(x).m_2(x)$ for some non unit polynomials m_1, m_2 where degree of m_1, m_2 are less than the degree of $m(x)$.
$\Rightarrow m(\beta) = m_1(\beta).m_2(\beta)$
$\Rightarrow m_1(\beta).m_2(\beta) = 0 \quad (\because m(x)$ is minimum polynomial of $\beta)$
$\Rightarrow m_1(\beta) = 0 \quad or \quad m_2(\beta) = 0$
This is a contradiction to the fact that the polynomial $m(x)$ is the smallest degree polynomial such that $m(\beta) = 0$.
Hence, The minimum function $m(x)$ of any element β is irreducible. □

Theorem 6. *If $f(x)$ is a polynomial with coefficient in the ground field F and if $f(\beta) = 0$, then $f(x)$ is divisible by $m(x)$, the minimum function of β.*

Proof. $f(x)$ is a polynomial with coefficient in the ground field F and $f(\beta) = 0$. $m(x)$ is the minimum function of β.
By applying Eucledian algorithm on $f(x)$:
$f(x) = m(x)q(x) + r(x)$
where $r(x)$ has degree less than the degree of $m(x)$.
As $f(\beta) = 0 \Rightarrow m(\beta)q(\beta) + r(\beta) = 0$
$\Rightarrow r(\beta) = 0 \quad (\because m(\beta) = 0)$
Since, $m(\beta)$ is the polynomial of least degree such that $m(x) = 0$, so $r(x)$, a polynomial with degree less than $m(x)$ and $r(\beta) = 0$ is a zero polynomial.
$\Rightarrow r(x) = 0$
$\Rightarrow f(x) = m(x)q(x) \Rightarrow f(x)$ is divisible by $m(x)$. □

Remark. The minimum function of β is unique. It also follows that if $p(x)$ is a monic irreducble polynomial and $p(\beta) = 0$, then $p(x)$ is the minimum function of β.

Theorem 7. *Every element of an extension field of degree k over a field F has a minimum function of degree k or less.*

Proof. The extension field is a vector space of dimension k. Therefore, for any element β the $k+1$ elements 1, β ,β^2 ,...,β^k cannot be linaerly independent.
$\Rightarrow \exists\ a_0\ ,\ a_1\ ,...,\ a_k \in F$ not all zero such that :

$$a_0.1\ +\ a_1.\beta\ +\ ...\ +\ a_k\beta^k = 0$$

$\Rightarrow$ There must be some polynomial of degree k or less in β which is equal to zero, and this polynomial can be made monic by dividing it by it's leading coefficient. $\square$

Chapter 2

The Coding Problem

2.1 The Communication Channel

The general data communication or storage system can be used to describe an informatiom storage system, if the storage medium is considered to be a channel. Telephone line is an example of transmission channel, a magnetic-tape unit including a writing and reading heads is an example of storage device.

,

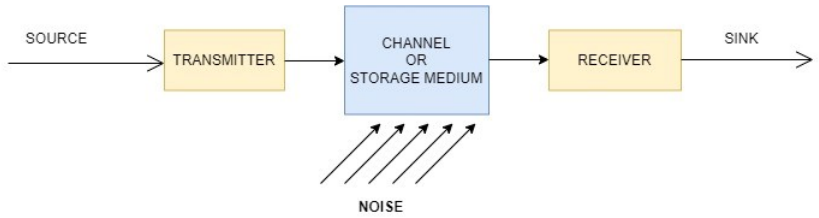

Figure 2.1: Block diagram of a general data communication or storage system.

The information from the **source** is usually in binary(consists of 0 and 1) or decimal digits (consists of digits from 0 to 9) or alphabetic information.
The encoder transforms these messages into electrical signals that are acceptable to the channel. These signals enters into channel and perturbed by noise.
The output enters the decoder and here the electrical signals gets converted into meaningful meassages and delivers to **sink**.
Every channel has an upper limit on the rate at which information can be transmitted reliably through the channel. This limitation of capacity of channel to transmit information is referred as the Channel Capacity.

In Figure-2.2, A source encoder converts source input to a binary stream. This is typically done by sampling an analog input, digitizing the samples and coding them into a binary stream. A channel encoder converts a binary stream to an analog signal that can be transmitted over distances as tones, radio waves and/or light. The channel encoder can take a number of bits from the bitstream at a time to create a multibit symbol. The modulator accepts a single channel symbol and produces at its output the corresponding channel waveform. This handling of a single channel symbol causes a loss in channel capacity.

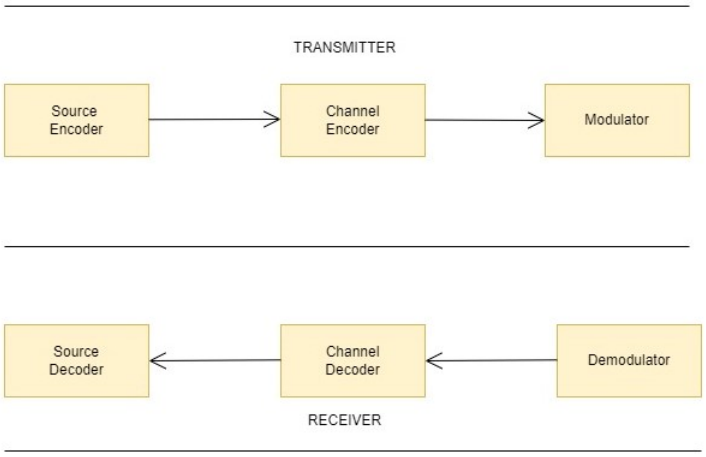

Figure 2.2: Detailed diagram of a Transmitter and receiver.

The demodulator performs the inverse operation of the modulator. It associate a channel symbol with the noise-corrupted received waveform. The independent demodulation of individual waveforms further cause in loss in channel capacity.

Due to channel noise there will be error in transmitted bits at the receiver's end. The use of error -correcting codes is an attempt to circumvent the problem of construction of a set of signals with arbitrarily small errorneous decoding.

2.2 Observations on Error-detecting and Error-Correcting Codes

The signals entered in the channel are perturbed by noise and hence there are occasional errors in the practical system. So, to detect and perhaps correct the errors, codes are used. These codes cannot correct every conceivable pattern of errors but can be designed in such a way that it can correct the most likely pattern. Here the assumption is that each symbol gets affected independently by the noise. So, probability of getting errorneous pattern as output depends only on the number of errors.
The communication channel shown in Figure 2.1 is strictly one-way channel. With a two-way channel, when an error is detected at one terminal, a request for a repeat can be given to effectively correct the error.
Error detection is a much simpler task than error correction. After the error is detected, the process of retransmission of redundant information increases. It can be observed that a combination of correction of most likely error patterns and detection with retransmission for less likely error patterns is more efficient than using only one method i.e., either error correction or detection and retransmission.

2.3 Type of Codes

The channel encoder receives a continuous sequence of information digits and it produces another sequence of output with somewhat more number of digits, which is then transferred to modulator. On the other hand, the decoder accepts a sequence of channel symbols from the demodulator and and transform it in somewhat shorter sequence. Codes are employed in channel encoder and decoder.

There are two different types of codes :

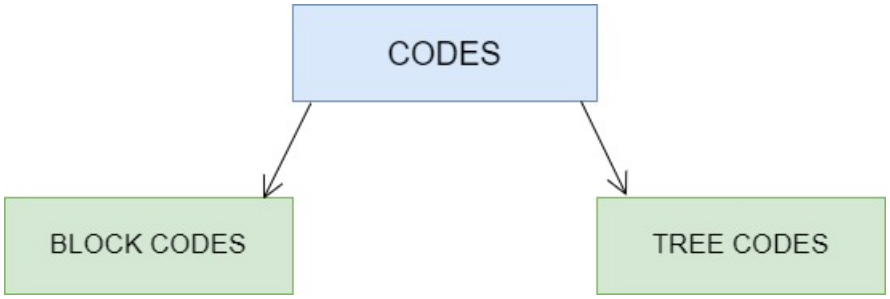

Figure 2.3:

Block Code :

- The encoder breaks the continuous sequence of information digits in k-symbols or blocks. So we get k bits for each block as $[i_1, i_2, ..., i_k]$.
- At channel encoder, block code is added to resolve errors produced due to noise in channel whereas in source encoder redundancy is reduced to improve bandwidth.
- With each possible information block with information bits $[i_1, i_2, ..., i_k]$ and redundant bits $[p_1, p_2, ..., p_r]$, Total bits of code becomes $n = k + r$ hence, $n > k$ and $n = [i_1, i_2, ..., i_k, p_1, p_2, ..., p_r]$.
- The result $n = [i_1, i_2, ..., i_k, p_1, p_2, ..., p_r]$ is called a **Code Word** of length n. A code word whose information bits are together is systematic code word. A code word whose information are not kept together is non systematic code words.

Tree Code :

- The encoder for a tree code operates on the information without breaking it up into independent blocks and process the information continuously and associates each long information bits of sequence with parity bits.

We will focus our study on Block codes only.

2.4 Block Codes

					PARITY		code words					
1	1	1	1		1		1	1	1	1	1	
1	1	1	1		0		1	1	1	1	0	
1	1	1	0		1		1	1	1	0	1	
1	1	1	0		0		1	1	1	0	0	
1	1	0	1		1		1	1	0	1	1	
1	1	0	1		0		1	1	0	1	0	
1	1	0	0		1		1	1	0	0	1	
1	1	0	0		0		1	1	0	0	0	a
1	0	1	1		1		1	0	1	1	1	
1	0	1	1		0		1	0	1	1	0	
1	0	1	0		1		1	0	1	0	1	
1	0	1	0		0		1	0	1	0	0	
1	0	0	1		1		1	0	0	1	1	c
1	0	0	1		0		1	0	0	1	0	
1	0	0	0		1		1	0	0	0	1	
1	0	0	0		0		1	0	0	0	0	
0	1	1	1		1		0	1	1	1	1	
0	1	1	1		0		0	1	1	1	0	
0	1	1	0		1		0	1	1	0	1	d
0	1	1	0		0		0	1	1	0	0	
0	1	0	1		1		0	1	0	1	1	
0	1	0	1		0		0	1	0	1	0	
0	1	0	0		1		0	1	0	0	1	
0	1	0	0		0		0	1	0	0	0	
0	0	1	1		1		0	0	1	1	1	
0	0	1	1		0		0	0	1	1	0	b
0	0	1	0		1		0	0	1	0	1	
0	0	1	0		0		0	0	1	0	0	
0	0	0	1		1		0	0	0	1	1	
0	0	0	1		0		0	0	0	1	0	
0	0	0	0		1		0	0	0	0	1	
0	0	0	0		0		0	0	0	0	0	

Figure 2.4:

The process of decoding the received encoded message is done by **decoding table**. The code word corresponding to a, b, c, d forms the first row of the table. If code words against a, b, c, d respectively are received then it is logical to associate the code words with a, b, c, d respectively. From Figure-2.4, we have collected all the 32 possible code words with length 5.

Now we will make some observations :

1. Firstly, let us make some assumptions. Suppose that the probabilty of receiving the same symbol as transmitted is Q. Suppose that the probability of occurence of one error in a specified position is P. Also assume that $Q > P$, i.e., the received block with no errors is more likely to occur than any other. (With P *and* $Q < 1$)

2. If a particular binary code word of length 'n' is transmitted then the chances that the code word is transmitted without any error at any specified position is $(1 - P)^n$, i.e., Q^n.

3. If a particular binary code of length 'n' is transmitted then the chances of the occurence of one error in a specified position is PQ^{n-1}.

4. Hence, the probability of particular received word that differs from the transmitted word in i positions is P^iQ^{n-i}.

5. Since Q *and* P both are strictly less than 1 ($\because Q$ can't be equal to 1 as we are not considering ideal case in which no error is detected), So for $n,\ i \in \mathbb{N}$ and $n > i > 1$, $Q^{n-i+1} > Q^{n-i}$ and $P^{i-1} > P^i$. This further imply that, the probability of receiving transmitted code word with one error is higher than receiving transmitted code word with two errors. Similarly, The probability of receiving code word with two errors is higher than three errors and so on till n. So, for large values of n, the probabilty of getting more than one error in a specified position is very less.

6. Assuming that all the code words are equally likely to be transmitted, the best decision at the receiver would be always to decode into a code word that differs from the received code word in the fewest positions.

Observing the points mentioned above, under each code word associated with $a,\ b,\ c,\ d$ respectively, we will start listing the remaining code words from Figure-2.4 in the increasing order of the number of errors at the position in the received code word. The Figure-2.5, given below decodes correctly if the received code word has no more than one error. If the code word has more than one error then it might get decoded wrongly.
For example: if 00110 is transmitted and two errors occur, resulting in 10100 as it is listed under the column of 00110, However, if two errors result in 10111, it will be decoded incorrectly into 10011, as 10111 is listed in the column under 10011.

1 1 0 0 0	0 0 1 1 0	1 0 0 1 1	0 1 1 0 1
1 1 0 0 1	0 0 1 1 1	1 0 0 1 0	0 1 1 0 0
1 1 0 1 0	0 0 1 0 0	1 0 0 0 1	0 1 1 1 1
1 1 1 0 0	0 0 0 1 0	1 0 1 1 1	0 1 0 0 1
1 0 0 0 0	0 1 1 1 0	1 1 0 1 1	0 0 1 0 1
0 1 0 0 0	1 0 1 1 0	0 0 0 1 1	1 1 1 0 1
1 1 1 1 0	0 0 0 0 0	0 1 0 1 1	1 0 1 0 1
0 1 0 1 0	1 0 1 0 0	1 1 1 1 1	0 0 0 0 1

Figure 2.5:

Now, The probability of correct decoding can be calculated as following, for the code shown in Figure-2.5. Let us suppose that 11000 is transmitted. It will be decoded correctly if any code word in it's column is received. Code words in it's column are of three types, as code words differ in no position, code words differ in one position and code words differ in 2 position. There is 1 code word of type1, 5 code word of type2 and 2 code word of type3.
So the probability of correct decoding is :
$P(correct\ \ decoding\ \ of\ \ 11000) = 1P^0Q^5 + 5P^1Q^4 + 2P^2Q^3$
similarly, correct decoding for other code words :
$P(correct\ \ decoding\ \ of\ \ 00110) = 1P^0Q^5 + 5P^1Q^4 + 2P^2Q^3$
$P(correct\ \ decoding\ \ of\ \ 10011) = 1P^0Q^5 + 5P^1Q^4 + 2P^2Q^3$
$P(correct\ \ decoding\ \ of\ \ 01101) = 1P^0Q^5 + 5P^1Q^4 + 2P^2Q^3$

The probability that the error is not detected if 11000 is transmitted is same as the probability of receiving other code word when 11000 is transmitted. Since, one

code word differs in four positions and other two in three positions each:
$P(undetected\ \ error\ \ for\ \ 11000) = 1P^4Q + 2P^3Q^2$

The code shown in Figure-2.5, any received code word above red boundary is decoded correctly into the code word at the top of the column, but if the decoder merely signal "error detection" for received words below the red boundary. This would correspond to single-error correction with detection of some combinations of two or more errors.
If the above code is used only for error detection, then the probability of correct decoding is Q^5.

Chapter 3

Linear Block Codes

3.1 Linear Block Codes,Generator Matrices, Parity-Check Martix

Let $\mathbb{F}_q^n$ denote the vector space of all n-tupls over the finite field $\mathbb{F}_q$. An (n, M) code C over $\mathbb{F}_q$ is a subset of $\mathbb{F}_q^n$. The vectors $(a_1, a_2, ..., a_n)$ in $\mathbb{F}_q^n$ are written in the form $a_1a_2, ..., a_n$. If C is a k-dimensional subspace of the vector space $\mathbb{F}_q^n$, then C is called an (n, k) linear code over $\mathbb{F}_q$.The linear block code C has q^k codewords.
Codes over $\mathbb{F}_2$ are called binary codes i.e., codes over the ring $\mathbb{Z}_2$ of intergers modulo 2. Similarly, codes over $\mathbb{F}_3$ and $\mathbb{F}_4$ are called ternary and quaternary codes. We can represent linear block code with either a generator matrix or a parity check marix.

Generator Matrix: Any set of basis vectors for a linear block code C over $\mathbb{F}_q$, can be considered as rows of a matrix $\mathbf{G}$, called a generator matrix of C. The row space of $\mathbf{G}$ is the linear code C, and the vector is a code vector if and only if it is a linear combination of the rows of $\mathbf{G}$. If the dimension of C is k, then the number of rows of $\mathbf{G}$ is k. Each distinct linear combination gives a distinct code vector. There are q^k code vectors in C and this type of code is called (n, k) code.
An (n, k) linear block code can be defined by a $k \times n$ generator matrix.

$$G_{k\times n} = \begin{bmatrix} g_0 \\ g_1 \\ \vdots \\ g_{k-1} \end{bmatrix} = \begin{bmatrix} g_{0,0} & g_{0,1} & g_{0,2} & \cdots & g_{0,n-1} \\ g_{1,0} & g_{1,1} & g_{1,2} & \cdots & g_{1,n-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ g_{k-1,0} & g_{k-1,1} & g_{k-1,2} & \cdots & g_{k-1,n-1} \end{bmatrix}$$

Parity-Check Matrix : A Linear (n, k) block can also be specified by an $(n-k)\times n$ parity check matrix $\mathbf{H}$. If C is a subspace of dimension k, its null space is a vcetor space C' of dimension $n-k$. The matrix H of rank $n-k$ whose row space is C' can be made with a basis for C' as rows. As, a code vector v in C is a linear combination of the rows of $\mathbf{G}$ and C is the null space of C', this implies v is orthogonal to every row of $\mathbf{H}$, that is, if $v = (v_0, v_1, ..., v_{n-1})$ is a binary n-tuple, then v is a codeword

if and only if

$$vH^T \ = \ (0, 0, ..., 0)$$

If the element in the i^{th} row and j^{th} column of $\mathbf{H}$ is denoted by h_{ij}, then above equation can also be represented as :

$$\sum_j v_j h_{ij} \ = \ 0, \qquad\qquad 1 \leq i \leq n-k$$

These equations can be simplified as :

$$\begin{aligned}
&v_0 h_{1,0} + v_1 h_{1,1} + ... + v_{n-1} h_{1,n-1} && = 0 \\
&v_0 h_{2,0} + v_1 h_{2,1} + ... + v_n h_{2,n-1} && = 0 \\
&... \quad ... \quad ... \quad ... \quad ... \quad ... && = 0 \\
&v_0 h_{n-k,0} + v_1 h_{n-k,1} + ... + v_n h_{n-k,n-1} && = 0
\end{aligned}$$

From the set of equations, we can observe that v must satisfy $n-k$ independent equations. Also, any combination of the above equations also give an equation that the components of v must satisfy, as:
for some scalars $a_0, a_1, ..., a_{n-1}$ in the field $\mathbb{F}_q$, consider :
$a_0(v_0 h_{1,0} + v_1 h_{1,1} + ... + v_{n-1} h_{1,n-1}) + a_1(v_0 h_{2,0} + v_1 h_{2,1} + ... + v_n h_{2,n-1}) + ... + a_{n-1}(v_0 h_{n-k,0} + v_1 h_{n-k,1} + ... + v_n h_{n-k,n-1}) = 0$
$\Rightarrow v_0(a_0 h_{1,0} + a_1 h_{2,0} + ... + a_{n-1} h_{n-k,0}) + v_1(a_0 h_{1,1} + a_1 h_{2,1} + ... + a_{n-1} h_{n-k,1}) + ... + v_{n-1}(a_0 h_{1,n-1} + a_1 h_{2,n-1} + ... + a_{n-1} h_{n-k,n-1}) = 0$
$\Rightarrow$ Components of v must satisfy the linear combinations of the $n-k$ equations.
This corresponds to the fact that v is orthogonal to every vector in C'. These equations are called generalized parity check. Since, the above equations holds for every v in C, so, in particular it holds for the k basis vectors of the matrix G. So we get :

$$GH^T = 0$$

Both a vector space C and its null space C' are subspaces of the spaces of all n-tuple, and therefore, both are Linear codes. They are called dual codes. If C is an (n, k) code, C' is an $(n, n-k)$ code.

3.2 The Hamming Metrics

Hamming Weight : The Hamming weight of a vector v is defined as the number of nonzero components. It is denoted by $w(v)$.
As we know the hamming distance between two vectors v_1 and v_2 is the number of positions in which they differ. If v_1 and v_2 in C and C is the subspace of the vector space $\mathbb{F}_q^n$, this implies both the vectors are code words then $v_1 - v_2$ is also a code word. The distance between v_1 and v_2 is $w(v_1 - v_2)$, i.e, the distance between two code words is the weight of another code word.

Weight of coset: The weight of a coset is said to be the weight of the minimum weight element in the coset.
The minimum distance for a linear code equals the minimum weight of its nonzero vectors.

Example1: For a block code with $q = 2$ and $n = 5$, the set of vectors

$(00000), (10011), (01010), (11001), (00101), (00101), (10110), (01111), (11100)$

forms a vector space V_1 and hence a linear block code.
The minimum weight is 2 and hence the minimum distance is 2.

Example2: Suppose there are four strings 010, 011, 101 and 111.

$010 +_2 011 = 001, d(010, 011) = 1.$

$010 +_2 101 = 111, d(010, 101) = 3.$

$010 +_2 111 = 101, d(010, 111) = 2.$

$011 +_2 101 = 110, d(011, 101) = 2.$

$011 +_2 111 = 100, d(011, 111) = 1.$

$101 +_2 111 = 010, d(011, 111) = 1.$

Hence, the Minimum Hamming Distance, dmin = 1 and minimum weight of nonzero vector is also 1.

3.3 Generator Matrix

Equivalent Codes: Two codes that differ only in arrangement of symbols have the same probability of error and such codes are called Equivaelnt codes.

Suppose that $\mathbf{G}$ is the generator matrix. If C is the row space of $\mathbf{G}$ then C' is code equivalent to C if and only if C' is the row space of matrix $\mathbf{G}'$ that is obtained from G by rearranging columns.
Thus, permuting the columns of a generator matrix leads to a generator matrix for an equivalent codes. The permuted matrix is the generator matrix for the same linear code, i.e., permutation of rows doesn't change the row space.

Combinatorially Equivalent: If one matrix can be obtained from another by a combination of row operations and column permutations, the two matrices are called combinatorially equivalent.
$\mathbf{G}$ and $\mathbf{G}'$ both are combinatorially equivalent in echelon canonical form.

$\mathbf{G}'$ can be formed from $\mathbf{G}$ by reducing $\mathbf{G}$ in echelon form. Then by rearranging the columns to form $k \times k$ identity matrix, resulting into another combinatorially

equivalent matrix $\mathbf{G}''$ as:

$$G''_{k\times n} = \begin{bmatrix} 1 & 0 & \dots & 0 & p_{1,1} & \dots & p_{1,n-k} \\ 0 & 1 & \dots & 0 & p_{2,1} & \dots & p_{2,n-k} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & p_{k,1} & \dots & p_{k,n-k} \end{bmatrix} = \begin{bmatrix} I_k P \end{bmatrix}$$

Let $u = (u_1, u_2, ..., u_k)$ be an arbitrary information k-tuple and consider v be the codeword, i.e., the combination of rows of $\mathbf{G}''$ then:

$$v = u\mathbf{G}''$$

$$v = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_k \end{bmatrix} \begin{bmatrix} 1 & 0 & \dots & 0 & p_{1,1} & \dots & p_{1,n-k} \\ 0 & 1 & \dots & 0 & p_{2,1} & \dots & p_{2,n-k} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & p_{k,1} & \dots & p_{k,n-k} \end{bmatrix}$$

$$v = (u_1, u_2, ..., u_k, c_1, c_2, ..., c_{n-k})$$

where

$$c_j = \sum_{i=1}^{k} a_i p_{ij}$$

The first k components are called the information symbols, and the last $n-k$ components are called the redundant or parity check symbols, which are added during encoding and help in detecting errors. A code of this type is called a systematic code.

Example: Let $k = 3$ and $n = 6$. The table gives a (6,3) linear block code.

Message	Codewords
(u_0, u_1, u_2)	$(v_0, v_1, v_2, v_3, v_4, v_5)$
(0 0 0)	(0 0 0 0 0 0)
(1 0 0)	(1 0 0 1 1 0)
(0 1 0)	(0 1 0 1 0 1)
(1 1 0)	(1 1 0 0 1 1)
(0 0 1)	(0 0 1 0 1 1)
(1 0 1)	(1 0 1 1 0 1)
(0 1 1)	(0 1 1 1 1 0)
(1 1 1)	(1 1 1 0 0 0)

We can write the coded bits in terms of information bits:

$$\begin{aligned} v_0 &= u_0 \\ v_1 &= u_1 \\ v_2 &= u_2 \\ v_3 &= u_0 +_2 u_1 \\ v_4 &= u_0 +_2 u_2 \\ v_5 &= u_1 +_2 u_2 \end{aligned}$$

$$\begin{bmatrix} v_0 & v_1 & v_2 & v_3 & v_4 & v_5 \end{bmatrix} = \begin{bmatrix} u_0 & u_1 & u_2 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

Therefore, the generator matrix for this code is :

$$G = \begin{bmatrix} g_0 \\ g_1 \\ g_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

We can observe that v_0, v_1, v_2 are information bits and v_3, v_4, v_5 are parity bits. We can find the codeword for the message $u = (101)$, by using generator matrix:

$$\begin{bmatrix} v \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} g_0 \\ g_1 \\ g_2 \end{bmatrix}$$

So we get :

$$\begin{aligned} [v_0 \quad v_1 \quad v_2 \quad v_3 \quad v_4 \quad v_5] &= 1 \,.\, g_0 + 0 \,.\, g_1 + 1 \,.\, g_2 \\ &= 1.(1\,0\,0\,1\,1\,0) + 0.(0\,1\,0\,1\,0\,1) + 1.(0\,0\,1\,0\,1\,1) \\ &= (1\,0\,0\,1\,1\,0) + (0\,0\,1\,0\,1\,1) \\ &= (1\,0\,1\,1\,0\,1) \end{aligned}$$

In this example, Linear code is in systematic form.

3.4 Parity Check Matrix

Let us consider the example to understand the connection between Generator matrix and Parity Check matrix.
Consider a $(7, 4)$ linear systematic code with a generator matrix:

$$G = \begin{bmatrix} g_0 \\ g_1 \\ g_2 \\ g_3 \end{bmatrix} = \left[\begin{array}{cccc|ccc} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{array}\right]$$

The encoding equation can be written as :

$$\begin{bmatrix} v_0 & v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \end{bmatrix} = \begin{bmatrix} u_0 & u_1 & u_2 & u_3 \end{bmatrix} \left[\begin{array}{cccc|ccc} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{array}\right]$$

We can write the encoded bits as :

$$\begin{aligned} v_0 &= u_0 \\ v_1 &= u_1 \\ v_2 &= u_2 \\ v_3 &= u_3 \\ v_4 &= u_0 +_2 u_2 +_2 u_3 \\ v_5 &= u_0 +_2 u_1 +_2 u_2 \\ v_6 &= u_1 +_2 u_2 +_2 u_3 \end{aligned}$$

We can write v_4, v_5 ,v_6 as the following:

$$\begin{aligned} v_4 +_2 u_0 +_2 u_2 +_2 u_3 &= 0 \\ v_5 +_2 u_0 +_2 u_1 +_2 u_2 &= 0 \\ v_6 +_2 u_1 +_2 u_2 +_2 u_3 &= 0 \end{aligned}$$

Here we will replace u_0 by v_0 , u_1 by v_1, u_2 by v_2 and u_3 by v_3. So, equivalently, we can write the encoding equations as :

$$\begin{aligned} v_4 + v_0 + v_2 + v_3 &= 0 \\ v_5 + v_0 + v_1 + v_2 &= 0 \\ v_6 + v_1 + v_2 + v_3 &= 0 \end{aligned}$$

We can write it in Matrix form:

$$\begin{bmatrix} v_0 & v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Since, for any codeword v, it satisfies the equation $vH^T = (0, 0, ..., 0)$. Hence, Parity Check Matrix is :

$$H = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

We can also observe that as $vH^T = (0, 0, ..., 0)$ holds and $v = u.\mathbf{G}$
$\Rightarrow u.\mathbf{G}H^T = (0, 0, ..., 0)$
In other words, $\mathbf{G}H^T = (0, 0, ..., 0)$, this means that rows of $\mathbf{G}$ and $\mathbf{H}$ are orthogonal to each other. So, This implies that $\mathbf{H}$ lies in the null space of $\mathbf{G}$ as described in the former section.

For a systematic code with generator matrix $\mathbf{G} = [I_k, P]$, the Parity Check matrix can be written as :

$$H = \begin{bmatrix} P^T I_{n-k} \end{bmatrix}$$

Example: Consider a (7,4) linear systematic code with generator matrix

$$G = \left[\begin{array}{cccc|ccc} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{array}\right]$$

Then the parity-check matrix in the systematic form is

$$H = \left[\begin{array}{cccc|ccc} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{array}\right]$$

Parity Check matrix is used to verify whether a vector is valid codeword or not. If we have generator matrix or parity check matrix, then the linear block code can be specified from either of them.

3.5 The Standard Array and Syndrome

Let C be an (n, k) linear code, let v_1 be the all-zero (identity) vector, and let $v_2, v_3, ..., v_{q^k}$ be the other code vectors. Let us suppose that $e_1, e_2, ..., e_l$ are the remaining $n - tuple$.
Form an array of vectors from the vector space, C as follows:

1. Arrange the 2^k codewords as the top row of the array with $v_1 = 0$ as the first element.
2. Suppose $j - 1$ rows of the array have been formed. Choose a vector e_j from V_n which is not in the previous $j - 1$ rows.
3. Form the j^{th} row by adding e_j to each codeword v_i in the top row and placing $e_j + v_i$ under v_i
4. Continue until all the vectors from C appear in the array.

$$\begin{array}{ccccccc} v_1 & v_2 & v_3 & v_4 & v_5 & \dots & v_{q^k} \\ e_2 & e_2 + v_2 & e_2 + v_3 & e_2 + v_4 & e_2 + v_5 & \dots & e_2 + v_{q^k} \\ e_3 & e_3 + v_3 & e_3 + v_3 & e_3 + v_4 & e_3 + v_5 & \dots & e_3 + v_{2^k} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{array}$$

All the n-tuples under first column are called the coset leaders. This table continues untill each possible n-tuple appears somewhere in the array. This array is called the standard array. It is useful in analyzing the block codes.

Error Pattern: If a vector u is transmitted and a vector v is received, then $v - u$ is called the error pattern.

Example: For a $(6, 3)$ linear code generated by the following matrix :

$$G = \left[\begin{array}{c} g_0 \\ g_1 \\ g_2 \end{array}\right] = \left[\begin{array}{cccccc} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{array}\right]$$

Then the standard array is shown below:
Step 1: We will write all the possible codewords v.
We have $v = u.G$, where all possible values of u are listed below.

Message (u_0, u_1, u_2)
(0 0 0)
(1 0 0)
(0 1 0)
(1 1 0)
(0 0 1)
(1 0 1)
(0 1 1)
(1 1 1)

Therefore, for each u listed above we need to calculate code vector v as $u.G$. All the possible values of v are as following:

message	code word
0 0 0	0 0 0 0 0 0
1 0 0	1 0 0 0 1 1
0 1 0	0 1 0 1 0 1
1 1 0	1 1 0 1 1 0
0 0 1	0 0 1 1 1 0
1 0 1	1 0 1 1 0 1
0 1 1	0 1 1 0 1 1
1 1 1	1 1 1 0 0 0

We can construct the standard array as follows:

0 0 0 0 0 0	1 0 0 0 1 1	0 1 0 1 0 1	1 1 0 1 1 0	0 0 1 1 1 0	1 0 1 1 0 1	0 1 1 0 1 1	1 1 1 0 0 0
1 0 0 0 0 0	0 0 0 0 1 1	1 1 0 1 0 1	0 1 0 1 1 0	1 0 1 1 1 0	0 0 1 1 0 1	1 1 1 0 1 1	0 1 1 0 0 0
0 1 0 0 0 0	1 1 0 0 1 1	0 0 0 1 0 1	1 0 0 1 1 0	0 1 1 1 1 0	1 1 1 1 0 1	0 0 1 0 1 1	1 0 1 0 0 0
0 0 1 0 0 0	1 0 1 0 1 1	0 1 1 1 0 1	1 1 1 1 1 0	0 0 0 1 1 0	1 0 0 1 0 1	0 1 0 0 1 1	1 1 0 0 0 0
0 0 0 1 0 0	1 0 0 0 1 1	0 1 0 0 0 1	1 1 0 0 1 0	0 0 1 0 1 0	1 0 1 0 0 1	0 1 1 1 1 1	1 1 1 1 0 0
0 0 0 0 1 0	1 0 0 0 0 1	0 1 0 1 1 1	1 1 0 1 0 0	0 0 1 1 0 0	1 0 1 1 1 1	0 1 1 0 0 1	1 1 1 0 1 0
0 0 0 0 0 1	1 0 0 0 1 0	0 1 0 1 0 0	1 1 0 1 1 1	0 0 1 1 1 1	1 0 1 1 0 0	0 1 1 0 1 0	1 1 1 0 0 1
1 0 0 1 0 0	0 0 0 1 1 1	1 1 0 0 0 1	0 1 0 0 1 0	1 0 1 0 1 0	0 0 1 0 0 1	1 1 1 1 1 1	0 1 1 1 0 0

As the first row of table consists of codewords with hamming weight 3 or large, we will start taking the next row with the vector which is not taken in the first row. So, we will take the 6-tuple with hamming weight 1. In the similar way we can complete the standard array.

Results:
1) Every vector in C appears exactly once in the standard array.

Proof. Proof by contradiction.
Consider any two vectors $e_i + v_{q^{k-i}}$ and $e_j + v_{q^{k-j}}$ are same.
$\Rightarrow e_i + v_{q^{k-i}} = e_j + v_{q^{k-j}}$
$\Rightarrow e_i + v_{q^{k-i}} + v_{q^{k-i}} = e_j + v_{q^{k-j}} + v_{q^{k-i}}$
$\Rightarrow e_i = e_j + v_{q^{k-j}} + v_{q^{k-i}}$, as $v_{q^{k-i}} + v_{q^{k-i}} = 0$
$\Rightarrow e_i = e_j + v_{q^{k-l}}$ for some code vector $v_{q^{k-l}}$, (as sum of two code vectors is code vector).
As $e_j + v_{q^{k-l}}$ lies in the row of $e_j \Rightarrow e_i$ lies in the row of e_j.
This is not possible.
Hence, Every vector in C appears exactly once in the standard array. □

No two vectors in the same row of a standard array are identical. This fact follows from above.

2) Each row is called a coset.

3) There are exactly 2^{n-k} cosets.

4) The first elemnet of each coset is called the coset leader.(Any element in the coset can be used as its coset leader. This does not change the elements of the coset, it changes the order of them.)

Let $v = (v_0, v_1, ..., v_{n-1})$ be a codeword from a binary (n, k) linear block code with generator matrix $\mathbf{G}$ and parity check matrix $\mathbf{H}$.

Assume u is the transmitted over a Binary Signal Channel, then binary received sequence.
$v = (v_0, v_1, ..., v_{n-1}) = u + e$ (modulo 2)
$v = (u_0, u_1, ..., u_{n-1}) + (e_0, e_1, ..., e_{n-1})$
$v = (u_0 + e_0, +e_1 + u_1, ..., u_{n-1} + e_{n-1})$
where $e = (e_0, e_1, ..., e_{n-1})$ is the error pattern.
After receiving u, the decoder must determine if u contains errors (error detection) and locate the errors in u (error correction).
Error detection is possible by using the concept of Syndrome.

Syndrome: Let v be a received vector, the $(n-k)$ component vector $s = (s_0, s_1, .., s_{n-k-1}) = v\mathbf{H}^T$ is called the syndrome.
We can easily observe that v is a code vector if and only if its syndrome is zero. As $\mathbf{H}$ is parity check matrix and v must statisfy all the rows of $\mathbf{H}$, so the all component of s are zero.
When s is non-zero then it indicates that there is an error in the received vector v.

The syndrome s computed from the received vector v actually depends only on the errir pattern e, and not on the transmitted code word u.

$$s = v.\mathbf{H}^T = (u + e).\mathbf{H}^T = u.\mathbf{H}^T + e.\mathbf{H}^T$$

As, u is a n-tuple code vector in the null space of H, hence $u.\mathbf{H}^T = 0$.
$\Rightarrow s = e.\mathbf{H}^T$

Example:Consider a $(7, 4)$ linear code with parity -check matrix:

$$H = \left[\begin{array}{cccc|ccc} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{array}\right]$$

Let $v = (0100001)$. Then the syndrome of v is:
suppose that $s = (s_0, s_1, s_2)$ be the syndrome of v
Then $s = (s_0, s_1, s_2) = v.\mathbf{H}^T$

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \neq 0$$

$\Rightarrow s \neq 0$
Hence, v is not a valid code word.

Consider v_1 and v_2 be two vectors.
$\Rightarrow v_1$ and v_2 are in the same coset if and only if $v_1 - v_2$ are in the code vector space. If the code space is the null space of $\mathbf{H}$, then $v_1 - v_2$ is in the code space if and only if :
$(v_1 - v_2)\mathbf{H}^T = 0$
By using distributive laws for matrices, we get :
$(v_1 - v_2)\mathbf{H}^T = v_1\mathbf{H}^T - v_2\mathbf{H}^T = 0$
$\Rightarrow s_1 - s_2 = 0$, where s_1, s_2 are syndrome of v_1, v_2
$\Rightarrow s_1 = s_2$ We can observe that, if syndrome of two vectors are equal then the vectors are in the same coset. Hence, this result holds both ways. So, two vectors v_1 and v_2 are in the same coste if and only if their syndromes are equal.

Hence, Each of the 2^{n-k} coset leader has a different syndrome. So, there is one to one correspondence between a coset leader and a syndrome.

3.6 Decoding of Linear Block Code

Recalling the structure of standard array.

$$\begin{array}{ccccccc}
v_1 & v_2 & v_3 & v_4 & v_5 & \dots & v_{q^k} \\
e_2 & e_2+v_2 & e_2+v_3 & e_2+v_4 & e_2+v_5 & \dots & e_2+v_{q^k} \\
e_3 & e_3+v_3 & e_3+v_3 & e_3+v_4 & e_3+v_5 & \dots & e_3+v_{2^k} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
e_{2^{n-k}} & e_{2^{n-k}}+v_2 & e_{2^{n-k}}+v_3 & e_{2^{n-k}}+v_4 & e_{2^{n-k}}+v_5 & \dots & e_{2^{n-k}}+v_{2^k}
\end{array}$$

The j^{th} column of a standard array:

$$D_j = \{v_j, e_2+v_j, e_3+v_j, ..., e_{2^{n-k}}+v_j\}$$

contains exactly one codeword.
If the received codeword r belongs to column D_j, then r is decoded into codeword v_j. So, if v_j is the tranmitted codeword, and the error pattern is a coset leader e_i, then $r = v_j = e_i$ is in the column of D_j, which contains v_j, which is correct decoding.

However, if the error pattern is not a coset leader then r is not in column D_j, hence it will result into incorrect decoding.

Let's say the error pattern x caused by the channel is in l^{th} coset and under the code vector $v_i \neq 0$.
$\Rightarrow x = e_l + v_i$
and the received vector is
$r = v_j + x$
$\Rightarrow r = e_l + v_i + v_j$
$\Rightarrow r = e_l + v_s$
$\Rightarrow$ The received vector is in D_s and it will be decoded as v_s, which is not the transmitted code vector v_j.

Hence, decoding is correct if and only if the error pattern is a coset leader, and the 2^{n-k} coset leaders are the only error patterns that can be corrected.

Bibliography

- W.Wesley Peterson , E.J. Weldon, Jr.. **Error-Correcting Codes**.Cambridge, Massachusetts, And London, England: The MIT Press, 1971.

- W.cary Hauffman and Vera Pless. **Fundamental of Error-Correcting Codes.**Cambridge University Press, 2003.