Encoding Using the Binary Schubert Code [43, 7] Using MATLAB

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

Abstract The main aim of this paper is to study the encoding process of binary Schubert code [43,7] using MATLAB, when one knows its generator matrix. Using MATLAB, we can study syndrome decoding, weight of a codeword, error correction and error detection of the code.

Keywords Linear code · Generator matrix · Parity check matrix · Schubert code · Encoding

1 Introduction

In the late 1940's due to Shannon, Hamming and Golay, the approach to error correction coding was taken by modern digital communications. Error is the most afflicted part in any type of communication. In this 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. So much research work has been done in the field of error correction and detection of incoming messages. In this paper we examine the family of Schubert unions, in particular, the binary Schubert code which were defined in [1] and how they are used in practice with the help of MATLAB. We have designed some representatives of generator matrix and the encoding as well as decoding process have been discussed. Using MATLAB, the error correcting capability of defined codes in paper [2] have been verified. Linear error correcting codes associated to Schubert varieties which is also known as sub-varieties of Grassmannian were introduced by Ghorpade and Lachaud [3]. These Grassmannian codes were studied by Ryan [4, 5] and Nogin [2, 6]. The upper bound for minimum distance of Schubert code was studied in [3].

M. S. Wavare (✉)
Department of Mathematics, Rajarshi Shahu Mahavidyalaya, Latur, MS, India

N. S. Darkunde · S. P. Basude
School of Mathematical Sciences, SRTM University, Nanded, MS 431606, India

P. M. Pawar et al. (eds.), *Techno-Societal 2020*,
https://doi.org/10.1007/978-3-030-69921-5_72

2 Justification and Need for the System

The transmission process, transmitted message(signal) passes through some noisy channel. Due to noise in this channel, some errors are introduced in received information. We have to detect errors and then correct it using some encoding and decoding techniques. There are two types of error control methods:

I. Error Detection with Retransmission
II. Forward acting error correction.

In the first method, retransmission request is done when some error has been occurred in received data, whereas in the second method, the error in the received message is detected and proper decoding technique at receiver end is applied. This forward acting error correction technique is used when a single source transmits signals to number of receivers. In this situation retransmission is impossible.

Error coding techniques play the important role in the digital communication. In the simulation tool of MATLAB, we have many error controls techniques like cyclic code, Constitutional code, linear block code, Reed Muller code and Hamming code but there is no space for Schubert code. So main aim of this paper is to generate a programme in MATLAB format and encode and detect the error in transmitting data with the help of MATLAB.

3 Preliminaries

3.1 Linear Codes [7, 8]

A linear code of length n with dimension k is a linear subspace C with dimension k of the vector space F_q^n, where F_q is the finite field with q elements. Such a code is called a q-ary code. If q = 2 or q = 3, the code is described as a binary code, or a ternary code respectively. The vectors in C are called codewords. The size of a code is the number of codewords in it and it is q^k.

3.1.1 Basic Definitions

Let F_q denotes the finite field having q elements, where $q = p^h$, p a prime and h a natural number. We denote F_q^n as the n-dimensional vector space over F_q. For any $x \in F_q^n$, the support of (x), is the nonzero entries in $x = (x_1, x_2, \ldots, x_n)$. The support weight (or Hamming norm) of x is defined by, $x = |\sup p(x)|$.

More generally, if W is a subspace of F_q^n, the support of W, Supp (W) is the set of positions where not all the vectors in W are zero and the support weight (or Hamming norm) of W is defined by,

$$W = |\text{supp }(W)|$$

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. The minimum distance d = d(C) of C is defined by,

$$d = d(C) = \min\{x : x \in C, x \neq 0\}$$

More generally, given any positive integer r, the rth higher weight dr = dr(C) is defined by

$$d_r = d_r(C) = \min\{D : D \; is \; a \; subspace \; of \; C \; with \; \dim \; D = r\}$$

Note that $d_1(C) = d(C)$. It also follows that $d_i \leq d_j$ when $i \leq j$ and that $d_k = |\sup p(C)|$, where k is dimension of code C. Thus, we have $1 \leq d_1 = d < d_2 < \ldots < d_{k-1} < d_k = n$. The first weight d_1 is equal to the minimum distance and the last weight is equal to the length of the code.

An $[n, k]_q$-code is said to be nondegenerate if it is not contained in a coordinate hyperplane of F_q^n. Two $[n, k]_q$-codes are said to be equivalent if one can be obtained from another by permuting coordinates and multiplying them by nonzero elements of Fq. It is clear that this gives a natural equivalence relation on the set of $[n, k]_q$-codes.

The matrix whose rows forms basis for linear code known as generator matrix where as the generator matrix of dual of any linear code C is known as parity check matrix for linear code C.

3.1.2 U-Error Detecting Code [9]

Let u be any 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 does not belong to a code C. A code is exactly u-error-detecting if it is u-error-detecting but it can not (u+1)-error-detecting code.

Theorem 3.1.1 [9] 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.

3.1.3 V-Error Correcting Code [10]

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.

Theorem 3.1.2 [9] A code C is v-error-correcting if and only if $d(C) \geq 2v + 1$.

3.1.4 Binary Schubert Code [1]

Let binary matrix G is the 5 × 18 matrix given by,

$$G = \begin{bmatrix} 100000000011111111 \\ 010000011111110000 \\ 001000101111001000 \\ 000101010101100101 \\ 000011100101001110 \end{bmatrix}$$

Then the linear code generated by this matrix G is known as binary Schubert code of length 18. This code is a binary [18,5,7] linear code.

Now in the next section let us consider the binary Schubert code of length 43, which will be constructed in the similar way of binary Schubert code of length 18 [1], So for the next section we have used a binary Schubert code of length 43 and dimension 7 to discuss the encoding of the message.

4 MATLAB Programme for Encoding of the Information Message

```
clc
I=eye (7); % Identity matrix of order 7
X= [ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1;
0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
0 0 0 0 0 1 1 1 0 0 1 1 0 1 1 1 1 1 1 0 0 0 0 1 0 0 0 0 1 0 1 0 0 0 0 1;
0 0 0 0 1 0 1 1 0 1 0 1 1 0 1 1 0 0 1 1 0 1 0 1 0 1 0 0 0 0 1 0 0 0 1 0;
0 1 1 1 0 0 0 0 1 0 0 0 1 1 1 1 0 1 0 1 1 0 0 0 1 0 0 1 0 1 0 0 0 1 0 0;
1 0 1 1 0 1 0 1 0 0 0 0 0 1 1 1 0 1 0 0 0 0 0 1 1 0 1 0 1 1 0 0 1 0 0 0;
1 1 0 1 1 0 0 1 0 0 0 0 1 0 1 1 0 0 0 1 0 0 0 1 1 1 1 1 0 0 0 1 0 0 0 0];% Matrix X
G=[I X]; % The generator matrix for binary Schubert Code of length 43
H=mod ([-X' eye(36)],2); % Parity Check Matrix for the defined Code
u=input ('Enter the message bit of length 7 for which you want to encode using the defined v=mod(u*G,2) % Encoding of message u
Synv=mod(v*H',2) % Syndrome of encoded message
```

u=input (‘Enter the message bit of length 7 for which you want to encode using the defined Enter the message bit of length 7 for which you want to encode using the defined Code

u = [1 1 0 0 1 0 0 1]

v =

Columns 1 through 13

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

Columns 14 through 26

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

Columns 27 through 39

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

Columns 40 through 43

1 1 0 1

Synv =

Columns 1 through 13

0 0 0 0 0 0 0 0 0 0 0 0 0

Columns 14 through 26

0 0 0 0 0 0 0 0 0 0 0 0 0

Columns 27 through 36

0 0 0 0 0 0 0 0 0 0

% Use a Binary length 43

n = 18; k = 5; %Length and Dimension

clc % Clearscreen

I=eye(7); % Identity matrix of order 7

X= [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1;

0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0;

0 0 0 0 0 1 1 1 0 0 1 1 0 1 1 1 1 1 0 0 0 0 1 0 0 0 0 1 0 1 0 0 0 0 1;

0 0 0 0 1 0 1 1 0 1 0 1 1 0 1 1 0 0 1 1 0 1 0 1 0 1 0 0 0 0 1 0 0 0 1 0;

0 1 1 1 0 0 0 0 1 0 0 0 1 1 1 1 0 1 0 1 1 0 0 0 1 0 0 1 0 1 0 0 0 1 0 0;

1 0 1 1 0 1 0 1 0 0 0 0 0 1 1 1 0 1 0 0 0 0 0 1 1 0 1 0 1 1 0 0 1 0 0 0;

1 1 0 1 1 0 0 1 0 0 0 0 1 0 1 1 0 0 0 1 0 0 0 1 1 1 1 1 0 0 0 1 0 0 0 0]; % Matrix X
G=[I X]; % The generator matrix for extended binary Schubert Code of length 43

H=mod([-X' eye(36)],2); % Denote H as a parity Check Matrix for defined Code

slt= syndtable(H); % Produce Syndrome look up table.

w = [1 0 0 1 1 1 1 1 1 0 0 1 1 1 1 0 0 1 0 1 1 1 0 0 0 0 0 0 1 1 1 1 1 0 1 0 1 0 1

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).

5 Conclusions

Thus, using the MATLAB, we have encoded the message information of length 43 with the help of generator and parity check matrix of given Schubert code [1] of length 43, But due to limitation of size of matrix one cannot construct syndrome table and can't decode it, which we can observe from the decoding program at the end in above section.

References

1. Waware Mahesh S (2019) Codes associated to Schubert varieties G(2,5) over F_2. New Trends Math Sci 7(1):71–78
2. Yu D (2017) Nogin (1997), Spectrum of codes associated with the grassmannian G(3; 6). Probl Inf Transm 33(2):26–36
3. Ghorpade SR, Lachaud G (2000) Higher weights of grassmann codes, coding theory, cryptography and related areas. In: Buchmann J, Hoeholdt T, Stictenoth H, Tapia-Resillas H (eds) Springer, Heidelberg, Germany, pp 120–131
4. Ryan CT (1987) An application of grassmannian varieties to coding theory. Congr Numer 57:257–271
5. Ryan CT (1987) Projective codes based on grassmann varieties. Congr Numer 57:273–279
6. Yu D (1996) Nogin, codes associated to grassmannians, arithmetic, geometry and coding theory. In: Pellikan R, Perret M (eds) S. G. Vl_adut_, Walter de Gruyter, Berlin, New York, pp 145–154
7. Hu_man WC, Pless V (2010) Fundamentals of error-correcting codes. Cambridge University Press
8. MacWilliams FJ, Sloane NJA (1977) The theory of error-correcting codes. Elsevier
9. Ling S, Xing C (2004) Coding theory-a first course, 1st edn. Cambridge University Press

10. Patil AR, Darkunde NS (2019) Algorithmic approach for error-correcting capability and decoding of linear codes arising from algebraic geometry. Springer, Information and Communication Technology for Competitive Strategies, pp 509–517
11. Pratap R, Getting started with MATLAB: a quick introduction for scientists and engineers, 7e. Oxford University Press