BCH CODES

Contents

Contents(Cont.)

1. Introduction

- BCH (Bose – Chaudhuri – Hocquenghem) codes form a large class of multiple random error-correcting codes.
- They were first discovered by Hocquenghem in 1959, and independently by Bose and Chaudhuri in 1960.
- They are cyclic codes.
- They are constructed based on Galois fields.

2 . Primitive BCH Codes

- For any integers $m \geq 3$ and $t < 2^{m-1}$, there exists a primitive BCH code with the following parameters:

$$n = 2^m - 1$$

$$n - k \leq mt$$

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

- This code is capable of correcting t or fewer random error over a span of $2^m - 1$ bit positions.
- The parameter t is called the designed **error-correcting capability** and the parameter $2t+1$ is called the **designed minimum distance**.

- For example, for $m = 6$ and $t = 3$, there exists a BCH code with

$$n = 2^6\text{-}1 = 63$$

$$n - k \leq 6 \times 3 = 18$$

$$d_{\min} \geq 2 \times 3 + 1 = 7$$

The code is a triple-error-correcting (63, 45) BCH code.

3. Generator Polynomial

- Let α be a primitive element in Galois field GF(2^m).
- For $1 \le i \le t$, let $\phi_{2i-1}(X)$ be the **minimum polynomial** of the field element α^{2i-1}.
- The degree of $\phi_{2i-1}(X)$ is m or a factor of m.
- The generator polynomial $g(X)$ of a ***t*-error-correcting primitive** BCH code of length $2^m - 1$ is given by

$$g(X) = LCM\{\phi_1(X), \phi_3(X), \cdots \phi_{2t-1}(X)\} \quad (1)$$

- Note that the degree of $g(X)$ is *mt* or less. Hence the number of parity-check bits, $n - k$, of the code is at most *mt*.

  Example 1: Let $m = 4$ and $t = 3$. Let α be a primitive element in GF(2^4) which is constructed based on the primitive polynomial $p(X) = 1 + X + X^4$. The minimum polynomials of α, α^3 and α^5 are:

$$\phi_1(X) = 1 + X + X^4$$

$$\phi_3(X) = 1 + X + X^2 + X^3 + X^4$$

$$\phi_5(X) = 1 + X + X^2$$

Hence the generator polynomial of the triple-error-correcting BCH code of length $n = 2^4 - 1 = 15$ is

$$\begin{aligned} g(X) &= LCM\{\phi_1(X), \phi_3(X), \phi_5(X)\} \\ &= \phi_1(X)\phi_3(X)\phi_5(X) \\ &= 1 + X + X^2 + X^4 + X^5 + X^8 + X^{10} \end{aligned}$$

The code is a (15, 5) cyclic code.

Example 2: Let $m = 6$ and $t = 5$. Let α be a primitive element in GF(2^6) which is constructed based on the primitive polynomial $p(X) = 1 + X + X^6$. The minimum polynomials of α, α^3, α^5, α^7 and α^9 are:

$$\phi_1(X) = 1 + X + X^6$$

$$\phi_3(X) = 1 + X + X^2 + X^4 + X^6$$

$$\phi_5(X) = 1 + X + X^2 + X^5 + X^6$$

$$\phi_7(X) = 1 + X^3 + X^6$$

$$\phi_9(X) = 1 + X^2 + X^3$$

Hence the generator polynomial of the 5-error-correcting BCH code of length $n = 2^6 - 1 = 63$ is

$$g(X) = LCM\{\phi_1(X), \phi_3(X), \phi_5(X), \phi_7(X), \phi_9(X)\}$$

$$= \phi_1(X)\phi_3(X)\phi_5(X)\phi_7(X)\phi_9(X)$$

The degree of $g(X)$ is 27. Consequently, the code is a (63, 36) cyclic code.

Table 1: A list of primitive BCH codes

n	k	t	n	k	t	n	k	t
7	4	1	255	163	12	511	268	29
15	11	1		155	13		259	30
	7	2		147	14		250	31
	5	3		139	15		241	36
31	26	1		131	18		238	37
	21	2		123	19		229	38
	16	3		115	21		220	39
	11	5		107	22		211	41
	6	7		99	23		202	42
63	57	1		91	25		193	43
	51	2		87	26		184	45
	45	3		79	27		175	46
	39	4		71	29		166	47
	36	5		63	30		157	51
	30	6		55	31		148	53
	24	7		47	42		139	54
	18	10		45	43		130	55
	16	11		37	45		121	58
	10	13		29	47		112	59
	7	15		21	55		103	61
127	120	1		13	59		94	62
	113	2		9	63		85	63
	106	3	511	502	1		76	85
	99	4		493	2		67	87
	92	5		484	3		58	91
	85	6		475	4		49	93
	78	7		466	5		40	95
	71	9		457	6		31	109
	64	10		448	7		28	111
	57	11		439	8		19	119
	50	13		430	9		10	121
	43	14		421	10	1023	1013	1
	36	15		412	11		1003	2
	29	21		403	12		993	3
	22	23		394	13		983	4
	15	27		385	14		973	5
	8	31		376	15		963	6
255	247	1		367	16		953	7
	239	2		358	18		943	8
	231	3		349	19		933	9
	223	4		340	20		923	10
	215	5		331	21		913	11
	207	6		322	22		903	12
	199	7		313	23		893	13
	191	8		304	25		883	14
	187	9		295	26		873	15
	179	10		286	27		863	16
	171	11		277	28		858	17

Table 1: A list of primitive BCH codes (Cont.)

n	k	t	n	k	t	n	k	t
1023	848	18	1023	553	52	1023	268	103
	838	19		543	53		258	106
	828	20		533	54		248	107
	818	21		523	55		238	109
	808	22		513	57		228	110
	798	23		503	58		218	111
	788	24		493	59		208	115
	778	25		483	60		203	117
	768	26		473	61		193	118
	758	27		463	62		183	119
	748	28		453	63		173	122
	738	29		443	73		163	123
	728	30		433	74		153	125
	718	31		423	75		143	126
	708	34		413	77		133	127
	698	35		403	78		123	170
	688	36		393	79		121	171
	678	37		383	82		111	173
	668	38		378	83		101	175
	658	39		368	85		91	181
	648	41		358	86		86	183
	638	42		348	87		76	187
	628	43		338	89		66	189
	618	44		328	90		56	191
	608	45		318	91		46	219
	598	46		308	93		36	223
	588	47		298	94		26	239
	578	49		288	95		16	147
	573	50		278	102		11	255
	563	51						

Table 2: The elements of GF(2^4) generated by $p(X) = 1 + X + X^4$

Power representation	Polynomial representation	4-Tuple representation
0	0	(0 0 0 0)
1	1	(1 0 0 0)
α	α	(0 1 0 0)
α^2	α^2	(0 0 1 0)
α^3	α^3	(0 0 0 1)
α^4	$1 + \alpha$	(1 1 0 0)
α^5	$\alpha + \alpha^2$	(0 1 1 0)
α^6	$\alpha^2 + \alpha^3$	(0 0 1 1)
α^7	$1 + \alpha + \alpha^3$	(1 1 0 1)
α^8	$1 + \alpha^2$	(1 0 1 0)
α^9	$\alpha + \alpha^3$	(0 1 0 1)
α^{10}	$1 + \alpha + \alpha^2$	(1 1 1 0)
α^{11}	$\alpha + \alpha^2 + \alpha^3$	(0 1 1 1)
α^{12}	$1 + \alpha + \alpha^2 + \alpha^3$	(1 1 1 1)
α^{13}	$1 + \alpha^2 + \alpha^3$	(1 0 1 1)
α^{14}	$1 + \alpha^3$	(1 0 0 1)

Table 2: Minimal polynomials of the elements in GF(2^4) generated by $p(X) = X^4 + X + 1$ (cont.)

Conjugate roots	Minimal polynomials
0	X
1	$X + 1$
$\alpha,\alpha^2,\alpha^4,\alpha^8$	$X^4 + X + 1$
$\alpha^3,\alpha^6,\alpha^9,\alpha^{12}$	$X^4 + X^3 + X^2 + X + 1$
α^5,α^{10}	$X^2 + X + 1$
$\alpha^7,\alpha^{11},\alpha^{13},\alpha^{14}$	$X^4 + X^3 + 1$

Table 3: Minimal polynomials for GF(2^m)

For example, the minimal polynomial of α^3 is

$\phi_3(X) = 1 + X^2 + X^3$, which is denoted by

3 (0, 2, 3)

The conjugate roots of $\phi_3(X)$ are

α^3 , $\alpha^{3\times2} = \alpha^6$, $\alpha^{6\times2} = \alpha^{12} = \alpha^5$

1. $m = 2$

1	(0, 1, 2)

2. $m = 3$

1	(0, 1, 3)	3	(0, 2, 3)

Table 3: Minimal polynomials for GF(2^m) (cont.)

3. $m = 4$			
1	(0, 1, 4)	3	(0, 1, 2, 3, 4)
5	(0, 1, 2)	7	(0, 3, 4)
4. $m = 5$			
1	(0, 2, 5)	3	(0, 2, 3, 4, 5)
5	(0, 1, 2, 4, 5)	7	(0, 1, 2, 3, 5)
11	(0, 1, 3, 4, 5)	15	(0, 3, 5)
5. $m = 6$			
1	(0, 1, 6)	3	(0, 1, 2, 4, 6)
5	(0, 1, 2, 5, 6)	7	(0, 3, 6)
9	(0, 2, 3)	11	(0, 2, 3, 5, 6)
13	(0, 1, 3, 4, 6)	15	(0, 2, 4, 5, 6)
21	(0, 1, 2)	23	(0, 1, 4, 5, 6)
27	(0, 1, 3)	31	(0, 5, 6)
6. $m = 7$			
1	(0, 3, 7)	3	(0, 1, 2, 3, 7)
5	(0, 2, 3, 4, 7)	7	(0, 1, 2, 4, 5, 6, 7)
9	(0, 1, 2, 3, 4, 5, 7)	11	(0, 2, 4, 6, 7)
13	(0, 1, 7)	15	(0, 1, 2, 3, 5, 6, 7)
19	(0, 1, 2, 6, 7)	21	(0, 2, 5, 6, 7)
23	(0, 6, 7)	27	(0, 1, 4, 6, 7)
29	(0, 1, 3, 5, 7)	31	(0, 4, 5, 6, 7)
43	(0, 1, 2, 5, 7)	47	(0, 3, 4, 5, 7)
55	(0, 2, 3, 4, 5, 6, 7)	63	(0, 4, 7)

Table 3: Minimal polynomials for GF(2^m) (cont.)

7. $m = 8$

1	(0, 2, 3, 4, 8)	3	(0, 1, 2, 4, 5, 6, 8)
5	(0, 1, 4, 5, 6, 7, 8)	7	(0, 3, 5, 6, 8)
9	(0, 2, 3, 4, 5, 7, 8)	11	(0, 1, 2, 5, 6, 7, 8)
13	(0, 1, 3, 5, 8)	15	(0, 1, 2, 4, 6, 7, 8)
17	(0, 1, 4)	19	(0, 2, 5, 6, 8)
21	(0, 1, 3, 7, 8)	23	(0, 1, 5, 6, 8)
25	(0, 1, 3, 4, 8)	27	(0, 1, 2, 3, 4, 5, 8)
29	(0, 2, 3, 7, 8)	31	(0, 2, 3, 5, 8)
37	(0, 1, 2, 3, 4, 6, 8)	39	(0, 3, 4, 5, 6, 7, 8)
43	(0, 1, 6, 7, 8)	45	(0, 3, 4, 5, 8)
47	(0, 3, 5, 7, 8)	51	(0, 1, 2, 3, 4)
53	(0, 1, 2, 7, 8)	55	(0, 4, 5, 7, 8)
59	(0, 2, 3, 6, 8)	61	(0, 1, 2, 3, 6, 7, 8)
63	(0, 2, 3, 4, 6, 7, 8)	85	(0, 1, 2)
87	(0, 1, 5, 7, 8)	91	(0, 2, 4, 5, 6, 7, 8)
95	(0, 1, 2, 3, 4, 7, 8)	111	(0, 1, 3, 4, 5, 6, 8)
119	(0, 3, 4)	127	(0, 4, 5, 6, 8)

8. $m = 9$

1	(0, 4, 9)	3	(0, 3, 4, 6, 9)
5	(0, 4, 5, 8, 9)	7	(0, 3, 4, 7, 9)
9	(0, 1, 4, 8, 9)	11	(0, 2, 3, 5, 9)
13	(0, 1, 2, 4, 5, 6, 9)	15	(0, 5, 6, 8, 9)
17	(0, 1, 3, 4, 6, 7, 9)	19	(0, 2, 7, 8, 9)
21	(0, 1, 2, 4, 9)	23	(0, 3, 5, 6, 7, 8, 9)

Table 3: Minimal polynomials for GF(2^m) (cont.)

25	(0, 1, 5, 6, 7, 8, 9)	27	(0, 1, 2, 3, 7, 8, 9)
29	(0, 1, 3, 5, 6, 8, 9)	31	(0, 1, 3, 4, 9)
35	(0, 8, 9)	37	(0, 1, 2, 3, 5, 6, 9)
39	(0, 2, 3, 6, 7, 8, 9)	41	(0, 1, 4, 5, 6, 8, 9)
43	(0, 1, 3, 6, 7, 8, 9)	45	(0, 2, 3, 4, 5, 6, 9)
47	(0, 1, 3, 4, 6, 8, 9)	51	(0, 2, 4, 6, 7, 8, 9)
53	(0, 2, 4, 7, 9)	55	(0, 2, 3, 4, 5, 7, 9)
57	(0, 2, 4, 5, 6, 7, 9)	59	(0, 1, 2, 3, 6, 7, 9)
61	(0, 1, 2, 3, 4, 6, 9)	63	(0, 2, 5, 6, 9)
73	(0, 1, 3)	75	(0, 1, 3, 4, 5, 6, 7, 8, 9)
77	(0, 3, 6, 8, 9)	79	(0, 1, 2, 6, 7, 8, 9)
83	(0, 2, 4, 8, 9)	85	(0, 1, 2, 4, 6, 7, 9)
87	(0, 2, 5, 7, 9)	91	(0, 1, 3, 6, 9)
93	(0, 3, 4, 5, 6, 7, 9)	95	(0, 3, 4, 5, 7, 8, 9)
103	(0, 1, 2, 3, 5, 7, 9)	107	(0, 1, 5, 7, 9)
109	(0, 1, 2, 3, 4, 5, 6, 8, 9)	111	(0, 1, 2, 3, 4, 8, 9)
117	(0, 1, 2, 3, 6, 8, 9)	119	(0, 1, 9)
123	(0, 1, 2, 7, 9)	125	(0, 4, 6, 7, 9)
127	(0, 3, 5, 6, 9)	171	(0, 2, 4, 5, 7, 8, 9)
175	(0, 5, 7, 8, 9)	183	(0, 1, 3, 5, 8, 9)
187	(0, 3, 4, 6, 7, 8, 9)	191	(0, 1, 4, 5, 9)
219	(0, 2, 3)	223	(0, 1, 5, 8, 9)
239	(0, 2, 3, 5, 6, 8, 9)	255	(0, 5, 9)

Table 3: Minimal polynomials for GF(2^m) (cont.)

9. $m = 10$

1	(0, 3, 10)	3	(0, 1, 2, 3, 10)
5	(0, 2, 3, 8, 10)	7	(0, 3, 4, 5, 6, 7, 8, 9, 10)
9	(0, 1, 2, 3, 5, 7, 10)	11	(0, 2, 4, 5, 10)
13	(0, 1, 2, 3, 5, 6, 10)	15	(0, 1, 3, 5, 7, 8, 10)
17	(0, 2, 3, 6, 8, 9, 10)	19	(0, 1, 3, 4, 5, 6, 7, 8, 10)
21	(0, 1, 3, 5, 6, 7, 8, 9, 10)	23	(0, 1, 3, 4, 10)
25	(0, 1, 5, 8, 10)	27	(0, 1, 3, 4, 5, 6, 8, 9, 10)
29	(0, 4, 5, 8, 10)	31	(0, 1, 5, 9, 10)
33	(0, 2, 3, 4, 5)	35	(0, 1, 4, 9, 10)
37	(0, 1, 5, 6, 8, 9, 10)	39	(0, 1, 2, 6, 10)
41	(0, 2, 5, 6, 7, 8, 10)	43	(0, 3, 4, 8, 10)
45	(0, 4, 5, 9, 10)	47	(0, 1, 2, 3, 4, 5, 6, 9, 10)
49	(0, 2, 4, 6, 8, 9, 10)	51	(0, 1, 2, 5, 6, 8, 10)
53	(0, 1, 2, 3, 7, 8, 10)	55	(0, 1, 3, 5, 8, 9, 10)
57	(0, 4, 6, 9, 10)	59	(0, 3, 4, 5, 8, 9, 10)
61	(0, 1, 4, 5, 6, 7, 8, 9, 10)	63	(0, 2, 3, 5, 7, 9, 10)
69	(0, 6, 7, 8, 10)	71	(0, 1, 4, 6, 7, 9, 10)
73	(0, 1, 2, 6, 8, 9, 10)	75	(0, 1, 2, 3, 4, 8, 10)
77	(0, 1, 3, 8, 10)	79	(0, 1, 2, 5, 6, 7, 10)
83	(0, 1, 4, 7, 8, 9, 10)	85	(0, 1, 2, 6, 7, 8, 10)
87	(0, 3, 6, 7, 10)	89	(0, 1, 2, 4, 6, 7, 10)
91	(0, 2, 4, 5, 7, 9, 10)	93	(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
95	(0, 2, 5, 6, 10)	99	(0, 1, 2, 4, 5)
101	(0, 2, 3, 5, 10)	103	(0, 2, 3, 4, 5, 6, 8, 9, 10)
105	(0, 1, 2, 7, 8, 9, 10)	107	(0, 3, 4, 5, 6, 9, 10)
109	(0, 1, 2, 5, 10)	111	(0, 1, 4, 6, 10)

Table 3: Minimal polynomials for GF(2^m) (cont.)

115	(0, 1, 2, 4, 5, 6, 7, 8, 10)	117	(0, 3, 4, 7, 10)
119	(0, 1, 3, 4, 6, 9, 10)	121	(0, 1, 2, 5, 7, 9, 10)
123	(0, 4, 8, 9, 10)	125	(0, 6, 7, 9, 10)
127	(0, 1, 2, 3, 4, 5, 6, 7, 10)	147	(0, 2, 3, 5, 6, 7, 10)
149	(0, 2, 4, 9, 10)	151	(0, 5, 8, 9, 10)
155	(0, 3, 5, 7, 10)	157	(0, 1, 3, 5, 6, 8, 10)
159	(0, 1, 2, 4, 5, 6, 7, 9, 10)	165	(0, 3, 5)
167	(0, 1, 4, 5, 6, 7, 10)	171	(0, 2, 3, 6, 7, 9, 10)
173	(0, 1, 2, 3, 4, 6, 7, 9, 10)	175	(0, 2, 3, 7, 8, 10)
179	(0, 3, 7, 9, 10)	181	(0, 1, 3, 4, 6, 7, 8, 9, 10)
183	(0, 1, 2, 3, 8, 9, 10)	187	(0, 2, 7, 9, 10)
189	(0, 1, 5, 6, 10)	191	(0, 4, 5, 7, 8, 9, 10)
205	(0, 1, 3, 7, 10)	207	(0, 2, 4, 5, 8, 9, 10)
213	(0, 1, 3, 4, 7, 8, 10)	215	(0, 5, 7, 8, 10)
219	(0, 3, 4, 5, 7, 8, 10)	221	(0, 3, 4, 6, 8, 9, 10)
223	(0, 2, 5, 9, 10)	231	(0, 1, 3, 4, 5)
235	(0, 1, 2, 3, 6, 9, 10)	237	(0, 2, 6, 7, 8, 9, 10)
239	(0, 1, 2, 4, 6, 8, 10)	245	(0, 2, 6, 7, 10)
247	(0, 1, 6, 9, 10)	251	(0, 2, 3, 4, 5, 6, 7, 9, 10)
253	(0, 5, 6, 8, 10)	255	(0, 7, 8, 9, 10)
341	(0, 1, 2)	343	(0, 2, 3, 4, 8, 9, 10)
347	(0, 1, 6, 8, 10)	351	(0, 1, 2, 3, 4, 5, 7, 9, 10)
363	(0, 2, 5)	367	(0, 2, 3, 4, 5, 8, 10)
375	(0, 2, 3, 4, 10)	379	(0, 1, 2, 4, 5, 9, 10)
383	(0, 2, 7, 8, 10)	439	(0, 1, 2, 4, 8, 9, 10)
447	(0, 3, 5, 7, 8, 9, 10)	479	(0, 1, 2, 4, 7, 8, 10)
495	(0, 1, 2, 3, 5)	511	(0, 7, 10)

4 . Properties

- Consider a *t*-error-correcting BCH code of length $n = 2^m - 1$ with generator polynomial $g(X)$
- $g(X)$ has $\alpha, \alpha^2, \alpha^3, \ldots, \alpha^{2t}$ as root, i.e. ,

$$g(\alpha^i) = 0, \text{ for } 1 \leq i \leq 2t.$$

- Since a code polynomial $V(X)$ is a multiple of $g(X)$ $V(X)$ also has $\alpha, \alpha^2, \alpha^3, \ldots, \alpha^{2t}$ as roots, i.e. ,

$$V(\alpha^i) = 0, \text{ for } 1 \leq i \leq 2t.$$

- A polynomial $V(X)$ of degree less than $2^m - 1$ is a code polynomial if and only if it has α, α^2, α^3 , …, α^{2t} as roots.

5. Decoding of BCH Codes

- Consider a *t*-error-correcting BCH code of length $n = 2^m - 1$ with generator polynomial
- Suppose a code polynomial $V(X)$

$$V(X) = v_0 + v_1 X + v_2 X^2 + \ldots + v_{n-1} X^{n-1}$$

  is transmitted .
- Let $r(X) = r_0 + r_1 X + r_2 X^2 + \ldots + r_{n-1} X^{n-1}$ be the received polynomial .

- Then $r(X) = V(X) + e(X)$, where $e(X)$ is the error pattern caused by the channel noise.
- To check whether $r(X)$ is a code polynomial or not, we simply test whether.

$$r(\alpha) = r(\alpha^2) = ... = r(\alpha^{2t}) = 0$$

- If yes, then $r(X)$ is a code polynomial; otherwise $r(X)$ is not a code polynomial and the presence of errors is detected.
- Decoding of a BCH code consists of the same three steps as for the decoding a cyclic code, namely:

  (1) syndrome computation,

  (2) determination of the error pattern, and

  (3) error correction.

6. Syndrome Computation

- The syndrome consists of $2t$ components in GF(2^m),

$$\overline{S} = (S_1, S_2, ..., S_{2t}),$$

where $S_i = r(\alpha^i)$ for $1 \le i \le 2t$.

- Computation: Let $\phi_i(X)$ be the minimum polynomial of α^i. Dividing $r(X)$ by $\phi_i(X)$, we obtain

$$r(X) = a(X) \times \phi_i(X) + b(X)$$

Then

$$S_i = b(\alpha^i)$$

- $S_i = b(\alpha^i)$ can be obtained by feedback shift-register with connections based on $\phi_i(X)$.

- Example 3: Let $m = 4$ and $t = 2$. Consider the double-error correcting BCH code of length $2^4 - 1 = 15$. The generator polynomial has

$$\alpha, \alpha^2, \alpha^3, \alpha^4,$$

as roots where α is a primitive element in GF(2^4) constructed based on $p(X) = 1 + X + X^4$. The code is a (15, 7) code.

- Suppose the vector,

$$\bar{r} = (100000010\,00000)$$

is received. Then

$$r(X) = 1 + X^8$$

- The minimum polynomial of α, α^2, α^3, α^4 is

$$\phi_1(X) = \phi_2(X) = \phi_4(X) = 1 + X + X^4.$$

- The minimum polynomial of α^3 is

$$\phi_3(X) = 1 + X + X^2 + X^3 + X^4.$$

- Dividing $r(X)$ by $\phi_1(X)$ and $\phi_3(X)$, we have

$$b_1(X) = X^2$$

$$b_3(X) = 1 + X^3$$

- Then

$$S_1 = b_1(\alpha) = \alpha^2$$

$$S_2 = b_1(\alpha^2) = \alpha^4$$

$$S_4 = b_1(\alpha^4) = \alpha^8$$

$$S_3 = b_3(\alpha^3) = 1 + \alpha^9 = \alpha^7$$

- Hence the syndrome of $r(X)$ is

$$\begin{aligned} \overline{S} &= (S_1, S_2, S_3, S_4) \\ &= (\alpha^2, \alpha^4, \alpha^7, \alpha^8) \end{aligned}$$

Example 4: this example uses the built functions in MATLAB6.1 to simulate a Chinese poem transmission over the AWGN channel as SNR = 6.0 dB.

```
clear
fid = fopen('杜甫詩.txt','r');
A = fread(fid); % A is an array of integers
S = char(A'); % to chinese character
D = size(A);
SNR = 6 ;   % 6dB = SNR =
            10log(Eb/No)=10log(signal_pw/(code_rate*2*nois
            e_var)), assume signal_pw = 1
noise_var = 1/(2*code_rate*10^(SNR/10));
```

```
KK = 16;  % info length
NN = 31;  % code length
TT = 3;   % error correct capability
for i=1:2:D(1)
    for j=1:8   % to get all bits in two adjacent integers
        b(j) = bitget(A(i),j);
        b(j+8) = bitget(A(i+1),j);
    end
    u = bchenco(b,NN,KK);  % bch(31,16,3) encoder
    v = -2*u+1;    % with BPSK format, 0 <--> 1,  1 <--> -1
    r =  v  + sqrt(noise_var)*randn(size(v));   % AWGN channel
    for j=1:31
          if  (r(j) > 0)  y(j) = 0;     % hard decision output
          else y(j) = 1;
```

```
    end
  end
  y1 = y(16:23);  % without decoded, obtain the first 8-bit.
  y2 = y(24:31);  % without decoded, obtain the second 8-bit.
  B(i)  =  bits2num(y1,8);
  B(i+1) = bits2num(y2,8);
  v_hat = bchdeco(y,KK,TT);
  C(i) = bits2num(v_hat(1:8),8);
  C(i+1) = bits2num(v_hat(9:16),8) ;
end
fid2 = fopen('杜甫詩(雜訊干擾).txt','w');
fid3 = fopen('杜甫詩(BCH decoded).txt','w');
fprintf(fid2,'%c',char(B));
fprintf(fid3,'%c',char(C));
fclose('all');
```

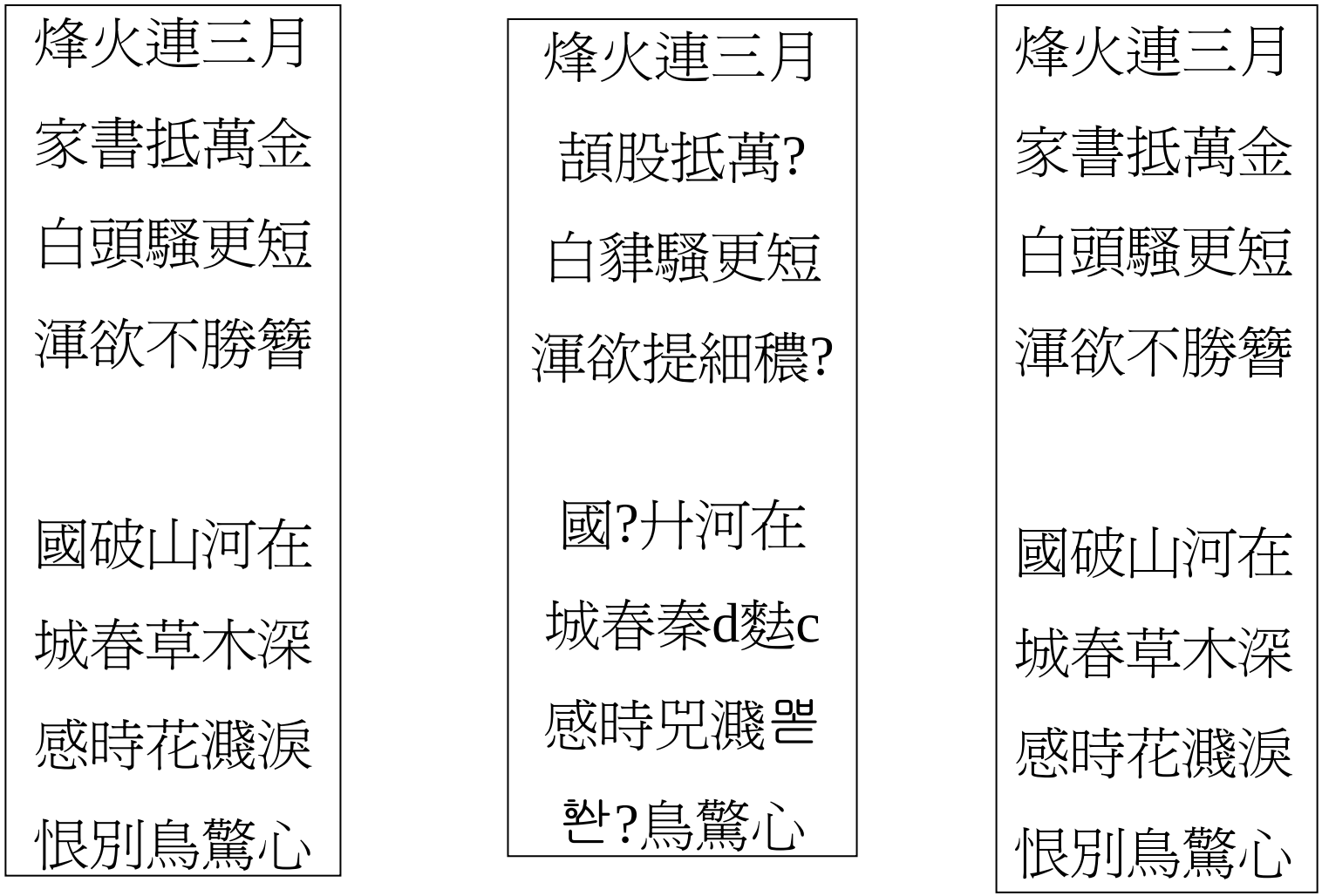


Figure 1: the original Chinese poem (left), degraded by AWGN (middle), recovered with BCH encoder/decoder (right)

HW #9

1. What is the generator polynomial $g(X)$ of the three-error-correcting of BCH code with length 31, which employed in Example 4 ?
2. Referring to Example 4, please input a new Chinese poem, and what is the result of this poem with BPSK signal transmission over the Rayleigh fading channel as SNR = 6.0 ?.

7. Syndrome and Error Pattern

- Since $r(X) = v(X) + e(X)$, then

$$S_i = r(\alpha^i) = v(\alpha^i) + e(\alpha^i) = e(\alpha^i), \qquad (2)$$

for $1 \le i \le 2t$.

- This gives a relationship between the syndrome and the error pattern.
- Suppose $e(X)$ has v errors at the locations $X^{j_1}, X^{j_2}, ..., X^{j_v}$, i.e.,

$$e(X) = X^{j_1} + X^{j_2} + \cdots X^{j_v}, \qquad (3)$$

where $0 \le j_1 < j_2 < ... < j_v < n-1$.

- From (2) and (3), we have the following relation between the syndrome components and error locations :

$$S_1 = e(\alpha) = \alpha^{j_1} + \alpha^{j_2} + \cdots + \alpha^{j_v}$$

$$S_2 = e(\alpha^2) = (\alpha^{j_1})^2 + (\alpha^{j_2})^2 + \cdots + (\alpha^{j_v})^2$$

$$\vdots$$

$$S_{2t} = e(\alpha^{2t}) = (\alpha^{j_1})^{2t} + (\alpha^{j_2})^{2t} + \cdots + (\alpha^{j_v})^{2t}$$

- If we can solve these 2t equations, we can determine $\alpha^{j_1}, \alpha^{j_2}, \cdots, \alpha^{j_v}$.
- Once $\alpha^{j_1}, \alpha^{j_2}, \cdots, \alpha^{j_v}$ are determined, the exponents $j_1, j_2, \ldots, j_v$ tell us the locations of errors.
- Any method for solving these 2t equations is a decoding method.
- Since the elements $\alpha^{j_1}, \alpha^{j_2}, \cdots, \alpha^{j_v}$ give the locations of errors, they are called error-location numbers.
- For simplicity, let

$$\beta_l = \alpha^{j_l}, \quad \text{for } 1 \le l \le v$$

- Then, we have

$$\begin{aligned} S_1 &= \beta_1 + \beta_2 + \ldots + \beta_v \\ S_2 &= \beta_1^2 + \beta_2^2 + \ldots + \beta_v^2 \\ &\vdots \\ S_{2t} &= \beta_1^{2t} + \beta_2^{2t} + \ldots + \beta_v^{2t} \end{aligned} \qquad (4)$$

- These equations are known as **power-sum** symmetric functions .

8 . Error-location Polynomial

- Define

$$\sigma(X) = (1+\beta_1 X)(1+\beta_2 X)...(1+\beta_v X)$$
$$= \sigma_0 + \sigma_1 X + \cdots \sigma_v X^v \qquad (5)$$

  where $\sigma_0 = 1$.

- $\sigma(X)$ has $\beta_1^{-1}, \beta_2^{-1}, \cdots, \beta_v^{-1}$ (the reciprocals of error-location numbers) as roots.
- $\sigma(X)$ is called the **error-location polynomial**.
- If we can determine $\sigma(X)$ from the **syndrome,** then the roots of $\sigma(X)$ give us the error location numbers, and hence the error pattern can be determined.

- From (5), we have the following relationship between the coefficients of $\sigma(X)$ and the error-location numbers:

$$
\begin{aligned}
\sigma_0 &= 1 \\
\sigma_1 &= \beta_1 + \beta_2 + \cdots \beta_v \\
\sigma_2 &= \beta_1\beta_2 + \beta_1\beta_3 + \cdots \beta_{v-1}\beta_v \\
\sigma_3 &= \beta_1\beta_2\beta_3 + \beta_1\beta_2\beta_4 + \cdots \beta_{v-2}\beta_{v-1}\beta_v \\
&\vdots \\
\sigma_v &= \beta_1\beta_2 \cdots \beta_v
\end{aligned}
\qquad (6)
$$

- The above equations are called elementary–symmetric functions.
- From (4) and (6), we have the following relationship between the syndrome and the coefficients of $\sigma(X)$:

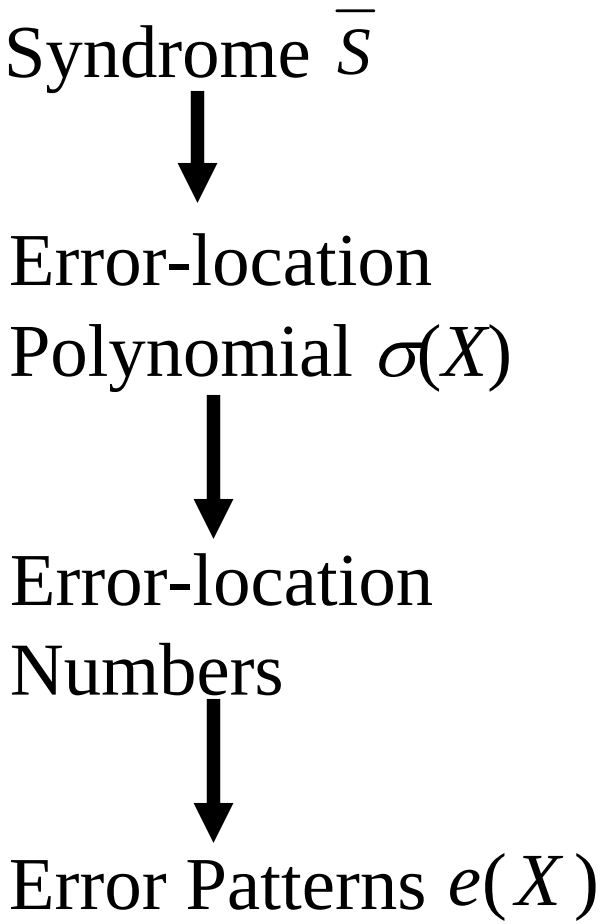
$$
\begin{aligned}
&S_1 + \sigma_1 = 0 \\
&S_2 + \sigma_1 S_1 + 2\sigma_2 = 0 \\
&S_3 + \sigma_1 S_2 + \sigma_2 S_1 + 3\sigma_3 = 0 \\
&\quad\vdots \\
&S_\nu + \sigma_1 S_{\nu-1} + \sigma_2 S_{\nu-2} + \sigma_3 S_{\nu-3} + \cdots \sigma_{\nu-1} S_1 + \nu\sigma_\nu = 0 \\
&S_{\nu+1} + \sigma_1 S_\nu + \sigma_2 S_{\nu-1} + \sigma_3 S_{\nu-2} + \cdots \sigma_\nu S_1 = 0 \\
&\quad\vdots
\end{aligned}
\qquad (7)
$$

- Note that 1 + 1 = 0. Then

$$i\sigma_i = \begin{cases} \sigma_i, & for\ odd\ i; \\ 0, & for\ \text{even}\ i \end{cases}$$

- The equations of (7) are called the **Newton's identities**.
- If we can determine σ_1 , σ_2 , …, σ_v from the Newton's identities, then we can determine the error-location numbers, β_1, β_2, … , β_v by finding the roots of $\sigma(X)$.

A procedure for finding the error pattern



9 . Decoding Procedure for BCH Codes

(1) Compute the syndrome

(2) Find error–location polynomial $\sigma(X)$.

(3) Determine the error-location numbers by finding the roots of $\sigma(X)$.

(4) Correct errors

Example 5: As we mentioned in Example 3, if the all-zero vector is transmitted, i.e.,

$$\overline{V} = (000000000\,00000)$$

$$V(X) = 0$$

and the received vector is,

$$\overline{r} = (100000010\,00000)$$

$$r(X) = 1 + X^8$$

The syndrome is computed as

$$\overline{S} = (S_1, S_2, S_3, S_4)$$

$$= (\alpha^2, \alpha^4, \alpha^7, \alpha^8)$$

From the Newton's identities (7), we obtain the following equations

$$S_1 + \sigma_1 = 0$$

$$S_2 + \sigma_1 S_1 + 2\sigma_2 = 0$$

$$S_3 + \sigma_1 S_2 + \sigma_2 S_1 + 3\sigma_3 = 0$$

$$S_4 + \sigma_1 S_3 + \sigma_2 S_2 + \sigma_3 S_1 + 4\sigma_4 = 0$$

Since it is a 2-error-correcting BCH code, the error-location polynomial is

$$\sigma(X) = \sigma_0 + \sigma_1 X + \sigma_2 X^2$$

$$\sigma_4 = \sigma_3 = 0$$

We substitutes these syndrome values and the Newton's identities become

$$\alpha^2 + \sigma_1 = 0$$

$$\alpha^4 + \sigma_1\alpha^2 = 0$$

$$\alpha^7 + \sigma_1\alpha^4 + \sigma_2\alpha^2 = 0$$

$$\alpha^8 + \sigma_1\alpha^7 + \sigma_2\alpha^4 = 0$$

$$\Rightarrow \sigma_1 = \alpha^2, \sigma_2 = \alpha^8$$

So that, $\sigma(X) = 1 + \alpha^2 X + \alpha^8 X^2$

The error location polynomial is factored as

$$\sigma(X) = 1 + \alpha^2 X + \alpha^8 X^2 = (1 + X)(\alpha^7 + X)$$
$$= \alpha^7 (1 + X)(1 + \alpha^8 X) = (1 + \beta_1 X)(1 + \beta_2 X)$$

The roots are 1 and α^7 . Their multiplication inverse elements is 1 ($1 = \alpha^0 = \beta_1$) and ($\alpha^8 = \beta_2$) . Therefore the error location numbers are 1 and α^8 . The error polynomial is

$$e(X) = X^0 + X^8 = 1 + X^8$$

The decoded polynomial is

$$\hat{V}(X) = r(X) + e(X) = 0 = V(X)$$

10. Berlekamp's Iterative Method for Finding $\sigma(X)$

- $\sigma(X)$ can be computed iteratively .
- The iteration process consists of $2t$ steps .
- At the u-th step, we determine a minimum-degree polynomial

$$\sigma^{(u)}(X) = 1 + \sigma_1^{(u)} X + \sigma_2^{(u)} X^2 + \cdots \sigma_{l_u}^{(u)} X^{l_u}$$

whose coefficients satisfy the first u Newton's identities

- Our next step is to find $\sigma^{(u+1)}(X)$ whose coefficients satisfy the first $u + 1$ Newton's identities.
- First we check whether $\sigma^{(u)}(X)$ also satisfies the $(u + 1)$-th Newton's identity.
- If yes, $\sigma^{(u+1)}(X) = \sigma^{(u)}(X)$ is a minimum degree polynomial whose coefficients satisfy the first $u + 1$ Newton identities.
- If not, a correction term is added to $\sigma^{(u)}(X)$ to form $\sigma^{(u+1)}(X)$ so that its coefficients satisfy the first $u + 1$ Newton's identities.

- To test whether $\sigma^{(u)}(X)$ satisfies the $(u + 1)$-th Newton's identity, we computes

$$d_u = S_{u+1} + \sigma_1^{(u)} S_u + \sigma_2^{(u)} S_{u-1} + ... + \sigma_{l_u}^{(u)} S_{u+1-l_u}$$

  This quantity is called the u-th discrepancy.

- If $d_u = 0$, then the coefficients of $\sigma^{(u)}(X)$ satisfies the $(u + 1)$-th Newton's identity. We set

  $\sigma^{(u+1)}(X) = \sigma^{(u)}(X)$

  $l_{u+1} = l_u$ (actually, l_u is the degree of $\sigma^{(u)}(X)$)

- If $d_u \neq 0$, $\sigma^{(u)}(X)$ needs to be adjusted to satisfy the $(u + 1)$-th Newton's identity.

- Correction: We go back to the steps prior to the u-th step and determine a polynomial $\sigma^{(p)}(X)$ such that $d_p \neq 0$ and $p - l_p$ has the largest value, where l_p is the degree of $\sigma^{(p)}(X)$. Then

$$\sigma^{(u+1)}(X) = \sigma^{(u)}(X) + d_u d_p^{-1} X^{(u-p)} \sigma^{(p)}(X)$$

- $\sigma^{(u+1)}(X)$ is the solution at the $(u+1)$-th step of the iteration process.
- Repeat the testing and correction until we reach the $2t$-th step. Then

$$\sigma(X) = \sigma^{(2t)}(X).$$

- The above iteration method applies to both binary and nonbinary BCH codes.

- For binary BCH codes, it can be reduced to t steps. Every even step can be skipped .

Execution of the Iteration Process

- Note that $\sigma^{(t)}(X) = 1 + S_1 X$ satisfies the first Newton's identity.
- To carry out the iteration, we set up a table as below and fill out the table:

u	$\sigma^{(u)}(X)$	d_u	l_u	$u - l_u$
-1	1	1	0	-1
0	1	S_1	0	0
1	$1+S_1X$			
⋮				
$2t$				

11. Finding the Roots of σ(X)

- The roots of σ(X) in GF(2^m) in σ(X). If $\sigma(\alpha^i) = 0$, then α^i is a root of $\sigma(X)$ and $\sigma^i = \alpha^{2^m - 1 - i}$ is an error-location number.
- Roots (error-location numbers) determination and error correction can be carried out simultaneously.
- To decode the first received digit r_{n-1}, we check whether α is a root of $\sigma(X)$. If $\sigma(\alpha) = 0$, then r_{n-1} is erroneous and must be corrected.
- If $\sigma(\alpha) \neq 0$, then r_{n-1} is error-free.

Iterative formula, $u = 1, \ldots 2t$

$$\sigma^{(u+1)}(X) = \sigma^{(u)}(X) + d_u d_p^{-1} X^{(u-p)} \sigma^{(p)}(X)$$

$$d_{u+1} = S_{u+2} + \sigma_1^{(u+1)} S_{u+1} + \sigma_2^{(u+1)} S_u + \cdots \sigma_{l_u}^{(u+1)} S_{u+2-l_u}$$

$$d_u = S_{u+1} + \sigma_1^{(u)} S_u + \sigma_2^{(u)} S_{u-1} + ... + \sigma_{l_u}^{(u)} S_{u+1-l_u}$$

- To decode r_{n-i} , we test whether $\sigma(\alpha^i) = 0$.
  If $\sigma(\alpha^i) = 0$, r_{n-i} is erroneous and must be corrected, otherwise r_{n-i} is error-free.

  Example 6: Consider the decoding of the (15, 5) triple-error-correcting BCH code given in Example 1. The generator polynomial has

$$\alpha, \alpha^2, \alpha^3, \alpha^4, \alpha^5, \alpha^6$$

  as roots. The roots α, α^2 and α^4 have the same minimum polynomial,

$$\phi_1(X) = \phi_2(X) = \phi_4(X) = 1 + X + X^4$$

The roots α^3 and α^6 have the same minimum polynomial,

$$\phi_3(\mathrm{X}) = \phi_6(\mathrm{X}) = 1 + X + X^2 + X^3 + X^4$$

The minimum polynomial of α^5 is

$$\phi_5(X) = 1 + X + X^2 \ .$$

- Suppose

$$\overline{V} = (000000000\,00000)$$

is transmitted and

$$\overline{r} = (000101000000100)$$

is received .

- Then $r(X) = X^3 + X^5 + X^{12}$
- Clearly the error pattern is

$$e(X) = X^3 + X^5 + X^{12}$$

- Dividing by $\phi_1(X)$, $\phi_3(X)$ and $\phi_5(X)$ respectively, we have the following remainders:

$$b_1(X) = 1$$

$$b_3(X) = 1 + X^2 + X^3$$

$$b_5(X) = X^2$$

- The syndrome components are:

$$S_1 = b_1(\alpha) = 1$$

$$S_2 = b_1(\alpha^2) = 1$$

$$S_4 = b_1(\alpha^4) = 1$$

$$S_3 = b_3(\alpha^3) = 1 + \alpha^6 + \alpha^9 = \alpha^{10}$$

$$S_6 = b_3(\alpha^6) = 1 + \alpha^{12} + \alpha^{18} = \alpha^5$$

$$S_5 = b_5(\alpha^5) = \alpha^{10}$$

Hence

$$\overline{S} = (1,1,\alpha^{10},1,\alpha^{10},\alpha^5)$$

- The 1st step:

$$d_1 = S_2 + \sigma_1^{(1)} S_1 = 1 + 1 \cdot 1 = 0$$

- The 2nd step:

$$\sigma^{(2)}(X) = \sigma^{(1)}(X) = 1 + X$$

$$d_2 = S_3 + \sigma_1^{(2)} S_2 = \alpha^{10} + 1 \cdot 1 = \alpha^5$$

- The 3rd step: since $d_2 \neq 0$ and in comparison of the values of d_u and $u - l_u$ in steps 1 and 0. $p = 0$ is taken (i.e. step 0),

$u = 2$

$$\sigma^{(u+1)}(X) = \sigma^{(u)}(X) + d_u d_p^{-1} X^{(u-p)} \sigma^{(p)}(X)$$

$$d_{u+1} = S_{u+2} + \sigma_1^{(u+1)} S_{u+1} + \sigma_2^{(u+1)} S_u + \cdots \sigma_{l_u}^{(u+1)} S_{u+2-l_u}$$

$$\sigma^{(3)}(X) = \sigma^{(2)}(X) + d_2 d_0^{-1} X^{(2-0)} \sigma^{(0)}(X)$$

$$= 1 + X + \alpha^5 X^2$$

$$d_3 = S_4 + \sigma_1^{(3)} S_3 + \sigma_2^{(3)} S_2$$

$$= 1 + 1 \cdot \alpha^{10} + \alpha^5 \cdot 1$$

$$= 0$$

$$\sigma^{(3)}(X) = \sigma^{(2)}(X) + d_2 d_0^{-1} X^{(2-0)} \sigma^{(0)}(X)$$

$$= 1 + X + \alpha^5 X^2$$

$$d_3 = S_4 + \sigma_1^{(3)} S_3 + \sigma_2^{(3)} S_2$$

$$= 1 + 1 \cdot \alpha^{10} + \alpha^5 \cdot 1$$

$$= 0$$

- The 4th step:

$$\sigma^{(4)}(X) = \sigma^{(3)}(X)$$

$$d_4 = S_5 + \sigma_1^{(4)} S_4 + \sigma_2^{(4)} S_3$$

$$= \alpha^{10} + 1 \cdot 1 + 1$$

$$= \alpha^{10}$$

- The 5th step: consider the values of d_u and $u - l_u$ prior to the 4th step. We take p = 2 (i.e. 2nd step)

$$\sigma^{(5)}(X) = \sigma^{(4)}(X) + \alpha^{10} \cdot \alpha^{10} X^{(4-2)}(1+X)$$

$$= 1 + X + \alpha^5 X^3$$

$$d_5 = S_6 + \sigma_1^{(5)} S_5 + \sigma_2^{(5)} S_4 + \sigma_3^{(5)} S_3$$

$$= \alpha^5 + 1 \cdot \alpha^{10} + \alpha^5 \alpha^{10}$$

$$= 0$$

- The 6th step:

$$\sigma^{(6)}(X) = \sigma^{(5)}(X) = 1 + X + \alpha^5 X^3$$

- Iterative process results in the following table :

u	$\sigma^{(u)}(X)$	d_u	l_u	$u - l_u$
-1	1	1	0	-1
0	1	1	0	0
1	$1+X$	0	1	0
2	$1+X$	α^5	1	1
3	$1+X+\alpha^5X^2$	0	2	1 (take $p = 0$)
4	$1+X+\alpha^5X^2$	α^{10}	2	2
5	$1+X+\alpha^5X^3$	0	3	2 (take $p = 2$)
6	$1+X+\alpha^5X^3$	-	-	-

- Iterative process results in the following table :

$$\sigma(X) = \sigma^{(6)}(X) = 1 + X + \alpha^5 X^3$$

- Note that

$$\sigma(\alpha^3) = \sigma(\alpha^{10}) = \sigma(\alpha^{12}) = 0$$

  Hence α^3, α^{10} and α^{12} are roots of $\sigma(X)$.

- The reciprocals of these 3 roots are $\alpha^{-3} = \alpha^{12}$, $\alpha^{-10} = \alpha^5$ and $\alpha^{-12} = \alpha^3$.
- Hence α^3, α^5 and α^{12} are the error-location numbers.
- Consequently, the error pattern is

$$e(X) = X^3 + X^5 + X^{12}$$

HW #10

1. In Example 6, what are the error pattern $e(X)$ and the output of BCH decoder, if the received vector is

$$\bar{r} = (0001010000\,00000)$$

12. The Step-By-Step Decoding

- In this decoding, we do not find the error-location polynomial. Instead, we use the concept of the error-trapping decoding.
- First we define the syndrome matrix as following:

$$M_v^{(0)} = \begin{bmatrix} S_1 & 1 & 0 & & \cdots & 0 \\ S_3 & S_2 & S_1 & 0 & \cdots & 0 \\ \vdots & & & & \vdots & \\ S_{2v-1} & S_{2v-2} & S_{2v-3} & & \cdots & S_v \end{bmatrix}$$

where $\overline{S} = (S_1, S_2, \cdots, S_{2t})$

- Theorem 1: For any binary BCH (n, k, t) code, and any v such that $1 \le v \le t$, the v by v syndrome matrix is singular if the number of errors is v-1 or less, and is nonsingular if the number of errors is v or v+1.
- The decision vector is defined

$$\overline{m} = (m_1, m_2, \cdots, m_t)$$

where decision bit m_v is calculated as

$$m_v = 0 \qquad \text{if } \det(M_v) = 0$$
$$m_v = 1 \qquad \text{if } \det(M_v) \neq 0$$

- The decision vector of a general *t*-error-correcting binary BCH code can be determined as follows:

(1)if there are no errors, then

$$\overline{m} = (0,0,\cdots,0) = (0^t)$$

(2)if there is one error, then

$$\overline{m} = (1,0,\cdots,0) = (1,0^{t-1})$$

(3)if there are *u* errors, then

$$\overline{m} \in \{(X^{u-2},1,1,0^{t-u})\}$$

where the symbol *X* can be 0 or 1.

(4)if there are no less than *t* errors, the

$$\overline{m} \in \{(X^{u-2},1,1)\}$$

- For example, if $t = 2$, the decision vector could be (0, 0) for no errors, (1, 0) for single error, and (1, 1) for two errors.

- For a received sequence $\bar{r} = (r_0, r_1, \cdots r_{n-1})$, and error pattern with weight L , the step-by-step decoding is

  (1)Set $i = t$, $j = -1$

  (2)Check if $\det(M_i) = 0$, If $\det(M_i) = 0$, $i = i-1$, go to step (4).

  (3)$j = j+1$, Complement r_{n-j-1} to determine the modified syndrome and whether $\det(M_i) = 0$. If $\det(M_i) \neq 0$, set $r_{n-j-1} = r_{n-j-1} + 1$ .Otherwise, $i = i-1$. If $i = 0$, go to (5).

(4)If $j < k$, go to (3). If $j = k$, go to (5).

(5)Read the information digits $r_{n-1}, r_{n-2}, \ldots, r_{n-k}$

- Example 7: in Example 3, for 2-error-correcting (15, 7) BCH code, suppose the all-zero vector is transmitted. and the received sequence is

$$\overline{r} = (0\,0\,0\,1\,0\,0\,0\,0\,1\,0\,0\,0\,0)$$

The syndrome is computed as follows:

$$r(X) = X^3 + X^9$$

$$S_1 = r(\alpha) = \alpha^3 + \alpha^9 = \alpha$$

$$S_2 = S_1^2 = \alpha^2$$

$$S_3 = r(\alpha^3) = (\alpha^3)^3 + (\alpha^3)^9 = \alpha^8$$

$$S_4 = S_2^2 = \alpha^4$$

$$\det(M_2) = \det\left(\begin{bmatrix} S_1 & 1 \\ S_3 & S_2 \end{bmatrix}\right) = S_1^3 + S_3 = 1 \neq 0$$

There are at least two errors at the received sequence. To complement r_{14} , the received sequence polynomial becomes as follows:

$$r(X) = X^3 + X^9 + X^{14}$$

The syndrome is computed as follows:

$$S_1 = r(\alpha) = \alpha^3 + \alpha^9 + \alpha^{14} = \alpha^7$$

$$S_2 = S_1^2 = \alpha^{14}$$

$$S_3 = r(\alpha^3) = (\alpha^3)^3 + (\alpha^3)^9 + (\alpha^3)^{14} = \alpha^{10}$$

$$S_4 = S_2^2 = \alpha^{13}$$

$$\det(M_2) = \det\left(\begin{bmatrix} S_1 & 1 \\ S_3 & S_2 \end{bmatrix}\right) = S_1^3 + S_3 = \alpha^{12} \neq 0$$

From the syndrome matrix, it shows that to complement r_{14} does not reduce the number of errors.

- For $p > 0$, $\bar{r}^{(p)}$ is obtained by cyclically shifting p places to the right.
- The syndrome matrix for $\bar{r}^{(p)} + 1$ is defined as $\bar{r}$ follows:

$$M_v^{(p)} = \begin{bmatrix} S_1^{(p)}+1 & 1 & 0 & \cdots & 0 \\ S_3^{(p)}+1 & S_2^{(p)}+1 & S_1^{(p)}+1 & \cdots & 0 \\ \vdots & & & & \\ S_{2v-1}^{(p)}+1 & S_{2v-2}^{(p)}+1 & S_{2v-3}^{(p)}+1 & \cdots & S_v^{(p)}+1 \end{bmatrix}$$

- The modified step-by-step decoding is described as follows:

(1) Set $i = t$, $p = 1$.

(2) Check $\det(M_i^{(0)}) = 0$, if $\det(M_i^{(0)}) = 0$, $i = i$-1, go to step (4).

(3) Complement $r_0^{(p)}$ to determine the shift syndrome and whether $\det(M_i^{(p)}) = 0$.
If $\det(M_i^{(p)}) \neq 0$, set $r_0^{(p)} = r_0^{(p)} + 1$.
Otherwise, $r_{n-p} = r_{n-p} + 1$, and $i = i$-1.
If $i = 0$, go to (5).

(4) $p = p$+1, if $p < k$, go to (3).
If $p = k$, stop.

(5) Read the information digits $r_{n\text{-}1}, r_{n\text{-}2}, \ldots, r_{n\text{-}k}$

Example 8: Consider 2-error-correcting (15, 7) BCH code over GF(2^4) with the primitive polynomial $1+X+X^4$. The generator polynomial is

$$g(X) = \phi_1(X)\phi_3(X) = 1 + X^4 + X^6 + X^7 + X^8$$

Suppose the all-zero vector is transmitted. And the received sequence is

$$\bar{r} = (0\,0\,0\,1\,0\,0\,0\,0\,1\,0\,0\,0\,0)$$

The whole procedure of the step-by-step decoding is shown in the following page. After 7 steps, the information vector is obtained.

$$r(X) = X^3 + X^9$$

$$S_1 = r(\alpha) = \alpha^3 + \alpha^9 = \alpha$$

$$S_2 = S_1^2 = \alpha^2$$

$$S_3 = r(\alpha^3) = (\alpha^3)^3 + (\alpha^3)^9 = \alpha^8$$

$$S_4 = S_2^2 = \alpha^4$$

$$\det(M_1^{(0)}) = S_1 = \alpha$$

$$\det(M_2^{(0)}) = \det(\begin{bmatrix} S_1 & 1 \\ S_3 & S_2 \end{bmatrix}) = S_1^3 + S_3 = 1$$

The corresponding decision vector is

$$\overline{m} = (1, 1)$$

As one time cycle shift in $r(X)$, the corresponding syndrome is

$$S_1^{(1)} = r^{(1)}(\alpha) = \alpha^4 + \alpha^{10} = \alpha^2$$

$$S_2^{(1)} = (S_1^{(1)})^2 = \alpha^4$$

$$S_3^{(1)} = r^{(1)}(\alpha^3) = (\alpha^3)^4 + (\alpha^3)^{10} = \alpha^{11}$$

$$S_4^{(1)} = (S_2^{(1)})^2 = \alpha^8$$

$$\det(M_1^{(1)}) = S_1^{(1)} + 1 = \alpha^8$$

$$\det(M_2^{(1)}) = \det\left(\begin{bmatrix} S_1^{(1)} + 1 & 1 \\ S_3^{(1)} + 1 & S_2^{(1)} + 1 \end{bmatrix}\right)$$

$$= \det\left(\begin{bmatrix} \alpha^8 & 1 \\ \alpha^{12} & \alpha \end{bmatrix}\right)$$

$$= \alpha^8$$

The corresponding decision vector is

$$\overline{m} = (1, 1)$$

The whole decoding process is shown as follows.

$\bar{r}$	$\bar{S}$	$\bar{m}$
0 0 0 1 0 0 0 0 0 1 0 0 0 0 0	$\alpha,\alpha^2,\alpha^8,\alpha^4$	1, 1

	$\bar{r}^{(p)}+1$	$\bar{S}^{(p)}+1$	$\bar{m}$	Info. $\bar{r}$
p=1	1 0 0 0 1 0 0 0 0 0 1 0 0 0 0	$\alpha^8,\alpha^1,\alpha^{12},\alpha^2$	1, 1	0 1 0 0 0 0 0
p=2	1 0 0 0 0 1 0 0 0 0 0 1 0 0 0	$\alpha^{14},\alpha^{13},\alpha^3,\alpha^{11}$	1, 1	0 1 0 0 0 0 0
p=3	1 0 0 0 0 0 1 0 0 0 0 0 1 0 0	$\alpha^1,\alpha^2,\alpha^8,\alpha^4$	1, 1	0 1 0 0 0 0 0
p=4	1 0 0 0 0 0 0 1 0 0 0 0 0 1 0	$\alpha^{10},\alpha^5,\alpha^{10},\alpha^{10}$	1, 1	0 1 0 0 0 0 0
p=5	1 0 0 0 0 0 0 0 1 0 0 0 0 0 1	$\alpha^{13},\alpha^{11},\alpha^2,\alpha^7$	1, 1	0 1 0 0 0 0 0
p=6	0 0 0 0 0 0 0 0 0 1 0 0 0 0 0	$\alpha^9,\alpha^3,\alpha^{12},\alpha^6$	1, 0	0 0 0 0 0 0 0
p=7	1 0 0 0 0 0 0 0 0 0 1 0 0 0 0	$\alpha^5,1,\alpha^9,\alpha^{10}$	1, 1	0 0 0 0 0 0 0

Example 8: This example uses the built functions in MATLAB 6.1 to simulate a photo transmission over the AWGN channel as SNR = 6.0 dB. All statements in this program are almost as Example 4. The original photo, deteriorated photo and recovered photo are shown in the following page.

Example 9: this example uses the built functions in MATLAB 6.1 to simulate a sound record transmission over the AWGN channel as SNR = 6.0 dB.

These three program files are attached on website "老胡小舖"

Figure 2: the original photo

Figure 3: the deteriorated photo by AWGN channel.

Figure 4: the recovered photo with BCH encoder/decoder

HW #10-1

1. Referring to Example 6, what is the result of this photo with BPSK signal transmission over the Rayleigh fading channel as SNR = 6.0 ? Compare with the results of Example 6.
2. Referring to Example 7, what is the result of this sound record with BPSK signal transmission over the Rayleigh fading channel as SNR = 6.0 ? Compare with the results of Example 7.
3. Referring to Example 3, find the BCH decoded sequence.