Solutions 7

Exercise 7.1. Let α be a primitive nth root of unity (that lives in $\mathbb{F}_{2^m}$). Such a code is defined all polynomials $m(x) \in \mathbb{F}_2[x]$ such that $\deg m(x) < n$ and $g(\alpha) = 0$. Now if we write F_{2^m} as a $\mathbb{F}_2$-vector space of dimension m, the powers of α written as elements of $\mathbb{F}_2^m$ take all possible values except 0. So a check matrix of the code is exactly the check matrix of a Hamming code.

Exercise 7.2.

1. Here is the table

i	β^i	i	β^i
1	$1+\omega$	5	$-1-\omega$
2	$-\omega$	6	ω
3	$1-\omega$	7	$-1+\omega$
4	-1	8	1

2. The conjugate root of β is β^3. The conjugate root of β^2 is β^6. So we get $g(z) = (z-1)(z-\beta)(z-\beta^2)(z-\beta^3)(z-\beta^6) = z^5 - z^3 + z^2 + z + 1$. The code is $[8,3,5]_3$-code.

3. We can correct up to 2 errors. Suppose $y(z) = c(z) + e(z)$ with $e(z) = az^r + bz^s$, where $a, b \in \mathbb{F}_3$ and $r, s \leq 7$. Set $X = \beta^r$ and $Y = \beta^s$. We have

$$S_0 = y(\beta^0) = e(\beta^0) = 0 = a + b$$

$$S_1 = y(\beta^1) = 1 - \omega = aX + bY$$

$$S_2 = y(\beta^2) = 1 = aX^2 + bY^2$$

So $a = -b$, $S_2/S_1 = X + Y = \frac{1}{1-\omega} = -1 - \omega$.

We can assume without loss of generality that $a = 1$ (if $a = -1$, this will exchange X and Y). So $X - Y = 1 - \omega$. So $X = -\omega = \beta^2$ and $Y = -1 = \beta^4$. So $e(z) = z^2 - z^4$. The sent message was thus $z^7 + z^4 - z^2 + z + 1 = g(x)(z^2 + 1)$

Exercise 7.3.

1. Let ω denote a primitive 31st root of unity in $\mathbb{F}_{32}$. First, we write a complete list of minimal polynomials for various powers of ω. Denote the minimal polynomial of ω^i by g_i. Then g_i is also the minimal polynomial of $\omega^{2i}, \omega^{4i}, \ldots$ (e.g., $g_1 = g_2 = g_4 = g_8 = g_{16}$). According to this, the powers of ω for which each g_i is the minimal polynomial are listed below:

$$\begin{aligned}
g_0 &: \ 0 \\
g_1 &: \ 1, 2, 4, 8, 16 \\
g_3 &: \ 3, 6, 12, 24, 17 \\
g_5 &: \ 5, 10, 20, 9, 18 \\
g_7 &: \ 7, 14, 28, 25, 19 \\
g_{11} &: \ 11, 22, 13, 26, 21 \\
g_{15} &: \ 15, 30, 29, 27, 23
\end{aligned}$$

Thus the degree of g_0 is 1 and the rest of the g_i's have degree 5. Now in order to design the code, we need to take three degree 5 polynomials that are factors of the generating polynomial $g(x)$ for the code (because the dimension of the code must be 16, we need the degree of the generating polynomial to be $31-16 = 15$), and wee need the generator polynomial to contain 6 consecutive powers of ω as its roots (as the distance of the code must be at least 7). We see that a suitable choice is $g(x) = g_1(x)g_3(x)g_5(x)$, which has $\omega^1, \ldots, \omega^6$ as its roots.

2. Let $H(z) = 1 - \sigma_1 z + \sigma_2 z^2 - \sigma_3 z^3$ be the error-locator polynomial. Thus $H'(z) = -\sigma_1 + 2\sigma_2 z - 3\sigma_3 z^2$. As we will be working with the coefficients of these polynomials in characteristic two, we can simplify the polynomials as $H(z) = 1 + \sigma_1 z + \sigma_2 z^2 + \sigma_3 z^3$ and $H'(z) = \sigma_1 + \sigma_3 z^2$. Let $S(z) := S_1 + S_2 z + S_3 z^2 + \cdots$ be a power series defined by the S_i. According to the Newton relations, we must have

$$H(z) \cdot S(z) = -H'(z),$$

thus,

$$(1 + \sigma_1 z + \sigma_2 z^2 + \sigma_3 z^3) \cdot (S_1 + S_2 z + S_3 z^2 + \cdots) = \sigma_1 + \sigma_3 z^2.$$

We already know that $\sigma_1 = S_1$. Now comparing the coefficients of various powers of z on both sides (namely, z^2 and z^3), we obtain

$$S_3 + \sigma_1 S_2 + \sigma_2 S_1 = \sigma_3 \Rightarrow S_3 + S_1 S_2 + \sigma_2 S_1 = \sigma_3, \tag{1}$$

and,

$$S_4 + \sigma_1 S_3 + \sigma_2 S_2 + \sigma_3 S_1 = 0 \Rightarrow S_4 + S_1 S_3 + \sigma_3 S_1 = \sigma_2 S_2. \tag{2}$$

Substituting (1) in (2) gives

$$\sigma_2 S_2 = S_4 + S_1 S_3 + S_3 S_1 + S_1^3 S_2 + \sigma_2 S_1^2,$$

thus,

$$\sigma_2 = (S_4 + S_1^2 S_2)/(S_2 + S_1^2).$$

Using this in (1) finally gives

$$\sigma_3 = S_3 + S_1 S_2 + S_1(S_4 + S_1^2 S_2)/(S_2 + S_1^2).$$

3. As seen in the lecture, a simple decoding algorithm will first compute the syndromes $S_1, \ldots, S_6$ from the received word y as $S_i := y(\omega^i)$, and then uses the identities obtained in the previous parts to compute $\sigma_1, \sigma_2, \sigma_3$, and thus, the error locator polynomial $H(z)$ (if all the S_i are zero, no error has occurred and decoding stops right away). The roots of $H(z)$ include the powers of ω at which the errors have occurred. By erasing y at the obtained positions, we can apply an erasure decoding algorithm (which amounts to solving a system of linear equations) to find the exact set of errors.