論文93-30A-11-3

제산방법에 의한 Reed-Solomon부호의 개선된 복호알고리듬

(Improved Decoding Algorithm on Reed-Solomon Codes using Division Method)

鄭 濟 洪*, 朴 鎭 秀**

(Je Hong Jeong and Jin Soo Park)

要 約

비순환 RS부호의 복호알고리듬은 오증(syndrome)을 계산하여, 오류위치다항식(error-location polynomial)을 구한 후, 오류위치를 판단하여, 오류치를 구하는 4단계로 이루어진다. 유크리트 알고리듬을 이용한 기존의 복호알고리듬 중에는 오류위치다항식과 오류추정다항식이 생략될 수 있는 방법이 있는데, 그 단점은 수신된 벡터로부터 오류정정을 위한 다항식을 구하는 과정에서 생략된 두 다항식을 구하는 것과 동일한 양의 계산이 필요하다는 것이다.

본 논문은 GF(2^m)상에서 다항식의 제산방법에 대하여 체계적으로 고찰하고, RS부호의 복호과정에서 수신된 벡터 V'로부터 오류위치다항식과 오류추정다항식을 생략하여 간단한 수식으로 오류정정을 위한 다항식을 구하는 방법을 제안한다. 특히 다항식 M(x)를 x, (x-1),…, (x- α^{n-2})로 각각 나누어 다항식 F' (x)를 구하는 과정에서 생략된 두 다항식을 구하는 것과 동일한 계산량이 필요없이 간단한 식으로 다항식 F' (x)를 나타낼 수 있는 방법을 제안한다. 또한 제안된 식을 RS부호의 복호과정에 적용함으로서 효율적으로 복호가 수행될 수 있는 새로운 복호알고리듬을 제안하고, 제시된 방법의 타당성을 검증하기 위하여 기존의 복호방법과 비교분석하며, 구체적인 예를 들어 컴퓨터 시뮬레이션을 수행함으로서 효율적으로 오류가 정정됨을 확인한다.

Abstract

Decoding algorithm of noncyclic Reed-Solomon codes consists of four steps which are to compute syndromes, to find error-location polynomial, to decide error-location, and to solve error-values. There is a decoding method by which the computation of both error-location polynomial and error-evaluator polynimial can be avoided in conventional decoding methods using Euclid algorithm. The disadvantage of this method is that the same amount of computation is needed that is equivalent to solve the avoided polynomial.

This paper considers the division method on polynomial on GF(2^m) systematically. And proposes a novel method to find error correcting polynomial by simple mathematical expression without the same amount of computation to find the two avoided polynomial. Especially, proposes the method which the amount of computation to find F' (x) from the division M(x) by x, (x-1),…, (x- α^{n-2}) respectively can be avoided. By applying the simple expression to decoding procedure on RS codes, proposes a new decoding algorithm, and to show the validity of presented method, computer simulation is performed.

*正會員, 建陽大學校 電算學科
(Dept. of Computer Science, Keonyang Univ.)

**正會員, 淸州大學校 電子工學科
(Dept. of Elec. Eng., Chongju Univ.)

接受日字 : 1992年 7月 29日

Ⅰ. 서 론

C. E. Shannon은 1948년에 발표한 통신로 부호화(channel coding)에 관한 정리에서 정보를 부호화하여 전송하면 정보전송율이 채널용량을 넘지 않을 때 발생한 오류를 적절한 수준까지 줄일 수 있다고 하였다. 그후 보다 효율적이고 신뢰성 있는 정보를 전달하기 위한 연구가 진행되었고, 오류제어를 위한 부호의 사용과 강력한 오류정정능력을 갖는 부호에 대한 연구가 계속되었다.

I. S. Reed와 G. Solomon은 비이원(nonbinary)부호인 Reed-Solomon(이하 RS)부호를 제안하였는데, 이는 산발오류(random error)뿐만 아니라 연집오류(burst error)도 정정할 수 있고, 효율적인 복호알고리듬과 탁월한 오류정정능력으로 각종 통신시스템과 컴퓨터의 기억장치등에 오류를 정정할 목적으로 널리 이용되고 있다. 그 대표적인 사례로는 미연방 전략정보망(JTIDS)의 (31, 15) RS부호, 미공군위성통신망(AFSTATCOM)의 (7, 2) RS부호, 컴팩디스크의 (32, 28)과 (28, 24) 단축RS부호, DAT(digital audio tape recoder)의 (32, 28)과 (32, 26) CIRC (cross interleaved Reed-Solomon code), IBM컴퓨터 기억장치에 사용되는 (61, 50) 단축 RS부호 등이 있다. [1]

일반적으로 RS부호의 복호알고리듬은 오증(syndrome)을 계산하고, 오류위치다항식을 구하며, 오류위치를 판단하여, 오류치를 계산하는 4단계로 이루어진다. 오증으로부터 오류위치다항식을 구하는 과정은 Peterson에 의하여 처음 제안된 Berlekamp/Massey 복호 알고리듬, 유크리트 알고리듬을 이용한 복호, 변환영역을 이용한 Blahut연구등이 있다. 특히 오류위치다항식을 구하는 과정은 반복계산을 수행해야 하기 때문에 복호기로 장치화하기에는 복잡한 단점이 있어 Chien은 이 단계를 생략한 직접복호법을 제안하였다. [4,8]

I. S. Reed와 T. K. Truong은 RS부호가 유한체 GF(Fn)상에서 FFT알고리듬을 이용하여 복호될 수 있음을 보였는데, 이방법은 일반적인 복호방법보다 더 간단하고, 소프트웨어적으로도 빨리 수행될 수 있는 장점이 있다. [6]

A. Shiozaki는 RS부호의 복호방법중 오류위치다항식과 오류추정다항식이 생략될 수 있는 복호방법을 제안하였다. [7] 그러나 그 단점은 수신된 벡터 V'로부터 오류정정을 위한 다항식 F'(x)를 구하는 과정에서 생략된 두 다항식을 구하는 것과 동일한 양의 계산이 필요하다는 점이다.

본 논문에서는 RS부호의 복호과정에서 제산방법을 이용하여 오류위치다항식과 오류추정다항식을 생략할 수 있는 방법을 제시하였다. 제시된 방법의 장점은 생략된 두 다항식을 구하는 것과 같은 양의 계산과정이 필요없이 간단한 식으로 오류정정을 위한 다항식 F'(x)를 구할 수 있다는 점이다.

또한 본 논문은 GF(2^m)상에서 다항식의 제산방법을 체계적으로 고찰하고, 특히 다항식 M(x)를 x, (x-1), ···, ($x-\alpha^{n-2}$)로 각각 나누어 오류정정을 위한 다항식 F'(x)를 구하는 과정에서 필요하였던 계산량을 생략하고 간단한 수식으로 다항식 F'(x)를 나타낼 수 있는 방법을 제안하였다. 그리고 그 식을 직접 RS부호의 복호과정에 적용함으로서 효율적으로 복호가 수행될 수 있는 새로운 복호알고리듬을 제안하고, 제시된 방법의 타당성을 검증하기 위하여 임의의 산발오류(random error)와 연집오류(burst error)의 경우 구체적인 예를 들어 컴퓨터 시뮬레이션을 수행하고, 기존의 방법과 비교 분석하였다.

Ⅱ. 다항식에 대한 제산방법

GF(2^m)상에서 deg [s(x)] ≥deg [r(x)] ≥0 인 두 다항식 r(x)와 s(x)에 대하여 최대공약수(GCD)를 구하는 과정은 다음과 같다.

$$s(x) = Q_1(x)r(x) + r_1(x) \quad (1)$$
$$r(x) = Q_2(x)r_1(x) + r_2(x) \quad (2)$$
$$r_1(x) = Q_3(x)r_2(x) + r_3(x) \quad (3)$$
$$\cdots\cdots$$
$$r_{n-2}(x) = Q_n(x)r_{n-1}(x) + r_n(x) \quad (4)$$
$$r_{n-1}(x) = Q_{n+1}(x)r_n(x) \quad (5)$$

식(5)의 $r_n(x)$는 r(x)와 s(x)의 최대공약수로 다음과 같이 나타낸다.

$$GCD\ [r(x), s(x)] = a(x)r(x) + b(x)s(x) \quad (6)$$

[정리 1] 두개의 다항식 r(x)와 s(x)가 주어졌을 때, 최대공약수 G(x)는 반드시 존재하고, G(x) = a(x)r(x) + b(x)s(x)를 만족하는 a(x)와 b(x)가 존재한다. [3]

식(6)은 다음과 같은 형태로 나타낼 수 있다.

$$a(x)s(x) + b(x)t(x) = d(x) \quad (7)$$

윗식을 반복적으로 적용하기 위한 초기조건은 다음

과 같다.

$$s_{-1}(x) = 1,\ t_{-1}(x) = 0,\ r_{-1}(x) = a(x)$$
$$s_0(x) = 0,\ t_0(x) = 1,\ r_0(x) = b(x) \quad (8)$$

$i \geq 1$에 대하여, $Q_i(x)$와 $r_i(x)$는 각각 몫과 나머지를 나타내는데 $r_{i-2}(x)$를 $r_{i-1}(x)$로 나눌때 다음 식이 성립한다.

$$r_{i-2}(x) = q_i(x) r_{i-1}(x) + r_i(x)$$
단, $\deg [r_i(x)] < \deg [r_{i-1}(x)]$

식(7)과 식(8)에 의하여 다항식 $s_i(x)$와 $t_i(x)$는 다음과 같이 정해진다.

$$s_{i-2}(x) = s_i(x) + Q_i(x) S_{i-1}(x)$$
$$t_{i-2}(x) = t_i(x) + Q_i(x) t_{i-1}(x) \quad (9)$$

이러한 과정을 반복하면, 식(9)를 다음과 같이 나타낼 수 있다.

$$s_n(x) a(x) + t_n(x) b(x) = r_n(x) \quad (10)$$

식(1)에서 식(10)에 나타낸 다항식 사이의 몇가지 중요한 관계는 표1과 같다.

표 1. 다항식의 성질
Table 1. Properties of polynomials.

(1) $t_0(x)r_1(x) - t_1(x)r_0(x) = (-1)^{i+1}a(x)$,	$0 \leq i \leq n+1$
(2) $s_0(x)r_1(x) - s_1(x)r_0(x) = (-1)^i b(x)$,	$0 \leq i \leq n+1$
(3) $s_0(x)t_1(x) - s_1(x)t_0(x) = (-1)^i$,	$0 \leq i \leq n+1$
(4) $s_i(x)a(x) + t_i(x)b(x) = r_i(x)$,	$-1 \leq i \leq n+1$

Ⅲ. Reed - Solomon부호의 복호방법

갈로아체 $GF(q^m)$에서 부호의 길이가 $n = q^m - 1$인 벡터

$C = (c_0, c_1, \cdots, c_{n-1})$

는 다음 식을 만족할때 RS(n, t)부호이다.

$$\sum_{i=0}^{n-1} c_i \alpha^{ij} = 0, \quad j = 1, 2, \ldots 2t \quad (11)$$

단, α는 $F_q m$에서 원시원이다.

식(11)은 생성함수(generating function)

$C(\alpha) = c_0 + c_1 x + \cdots + c_{n-1} x^{n-1}$

에 대한 조건으로 다음과 같이 변형할 수 있다.

$$C(\alpha^j) = 0, \quad j = 1, 2, \cdots, 2t \quad (12)$$

[정리 2] 식(11)에 의하여 정의된 RS부호는 $C = (c_0, c_1, \cdots, c_{n-1}) \in V_n(F_q n)$인 모든 벡터로 구성되는데, $C(x) = \sum_{i=0}^{n-1} c^i x^i$는 $I(x)$가 $F_q m$상에서 $n-2t-1$이하의 차수 인 $C(x) = (x-\alpha)(x-\alpha^2)\cdots(x-\alpha^{2t}) \cdot I(x)$ 형태를 가지며, 매개변수는 다음과 같다. [9]

$$n = q^m - 1$$
$$k = n - 2t \quad (13)$$
$$d_{min} = 2t + 1$$

수신된 벡터를 R, 부호어를 C라 할때, 식(11)로 정의된 RS부호의 복호알고리듬은 다음과 같다.

(1) 오증(syndrome)을 계산한다.

$$S_j = \sum_{i=0}^{n-1} R_i \alpha^{ij}, \quad j = 1, 2, \ldots, 2t$$

(2) x^{2t}와 $s(x) = s_1 + s_2 x + \cdots + s_{2t} x^{2t-1}$에 제산방법을 수행하는데, $\deg [rj(x)] < t$ 일때 수행과정을 끝낸 후, $\sigma(x) = t_j(x)$, $\omega(x) = r_j(x)$ 라 놓는다.

(3) 오류위치의 집합, 즉

$B = \{\beta \in F_q m,\ \sigma(\beta) = 0\}$ 인 B를 구한다.

(4) 각 $\beta \in B$에 대하여, $E\beta = \omega(\beta)/\sigma'(\beta)$라 놓는다.

(5) 각 $i = 0, 1, \cdots n-1$에 대하여

$E_i = 0$, $\alpha^i \notin B$인 경우.

$= E\beta$, $\alpha^i = \beta \in B$인 경우.

(6) $C = (R_0 + E_0,\ R_1 + E_1, \cdots, R_{n-1} + E_{n-1})$을 출력한다.

위 알고리듬에서 (3)은 오류위치집합이고, (4)는 오류치의 집합이다. A. Shiozaki는 (3)과 (4)가 생략될 수 있는 새로운 방법을 제안하였는데, 그 단점은 $F'(x)$를 구하는 부분에서 오류위치다항식과 오류치다항식을 구하는 것과 동일한 계산량이 필요하다는 점이다. [7]

본 논문에서는 다항식 $F'(x)$를 구하는 과정에서 필요했던 계산량이 생략되고, 간단한 수식을 이용하여 $F'(x)$를 구할 수 있는 방법을 제안하고, 그 결과에 제산방법을 적용하여 효율적으로 복호과정이 수행될 수 있음을 확인하였다.

Ⅳ. 제산방법을 이용한 RS부호의 개선된 복호방법

확대체 $GF(2^m)$상에서 k개의 정보기호를

$U = (u_0, u_1, \cdots, u_{k-1})$, $u_j \in GF(2^m)$ (14)

이라하면 , 식(14)를 다항식으로 표현하여

$$F(x) = u_0 + u_1x + \cdots + u_{k-1}x^{k-1} \quad (15)$$

로 나타낼 수 있으며, α 가 원시원 (primitive element)일때, 부호벡터 V는

$$V=(v_0,v_1,v_2,\ldots,v_{n-1})$$
$$=(F(0),F(1),F(\alpha),\ldots,F(\alpha^{n-2})),n=2^m$$
$$M(x)=\prod_{\beta\in GF(2^m)}(x-\beta)=(x^2)^m+x \text{ 라 할때} \quad (16)$$

모든 β 에 대하여

$$F(x)=\sum_{\beta\in GF(2^m)}F(\beta)M(x)/(x-\beta) \quad (17)$$

로 나타낼 수 있으며, 수신벡터 v'는 v' = v + e 가 된다.

B를 발생한 오류의 집합이라 하면, 오류위치다항식은

$$\sigma(x)+\prod_{\beta\in B}(x-\beta) \quad (18)$$

이고, 오류치다항식은

$$\omega(x)=\sum_{\beta\in B}E\beta\prod_{\gamma\in B,\gamma\neq B}(x-\gamma) \quad (19)$$

가 된다. F(x)와 v'에 대응하는 F'(x)의 관계식은 식(16)-식(19)로부터

$$F'(x)-F(x)=\{\omega(x)M(x)/\sigma(x)\} \quad (20)$$
$$\sigma(x)F'(x)=\sigma(x)F(x)+\omega(x)M(x)$$
즉, $$\sigma(x)F'(x)\equiv\sigma(x)F(x)+\omega(x)M(x) \quad (21)$$

M(x)와 식(20)의 F'(x)에 a(x) = M(x), b(x) = F'(x)라 놓고, 제산방법을 반복적으로 적용한다. deg [$r_j(x)$] < (k+t)일때 제산과정을 종결한 후, F(x) = $r_j(x)/t_j(x)$를 구한다.

식(10)은 a(x), b(x)의 최대공약수를 구하기 위한 연속적인 단계로서 그로부터 다음 보조정리(1)이 성립한다.

[보조정리 1] 만일 $\nu \geq$deg [GCD(a(x), b(x)], $\mu+\nu$ = deg [a(x)] -1이면 다음과 같은 조건을 만족하는 첨자 j가 존재한다.

$$\deg[t_j'(x)]\leq\mu,\deg[r_j'(x)]\leq\nu \quad (22)$$

[정리 3] t(x)≠0, r(x)≠0인 다항식으로 {t(x) b(x) - r(x)}/a(x)의 나머지가 0이고, deg [t(x)] + deg [r(x)] < deg [a(x)] 이면, 보조정리(1)의 첨자 j가 존재하여,

$$t(x) = \lambda(x)t_j(x)$$
$$r(x) = \lambda(x)r_j(x) \quad (23)$$

의 관계가 성립한다. [7]

보조정리(1)에서 a(x) = M(x), b(x) = F'(x)라 하면, ν = k+t+1, μ = t이다. 식(20)을 이용하여 t(x) = σ(x), r(x) = σ(x)F(x)을 정리(3)에 적용하면, 식(24)가 성립한다.

$$F(x)=\{\sigma(x)F(x)\}/\sigma(x)=r(x)/t(x)=r_j(x)/t_j(x) \quad (24)$$

여기서, j는 deg [$r_j(x)$] < k+t인 첫번째 첨자이다.

deg [$r_j(x)$] - deg [$t_j(x)$] ≥k 또는 $r_j(x)$가 $t_j(x)$의 곱이 아니면 t보다 많은 오류가 발생하여 복호가 능하지 않은 오류가 검출된다.

식(24)의 결과로 오류위치다항식과 오류추정다항식을 구함이 없이 복호과정을 수행할 수 있다. 그러나 F'(x)를 구하는 과정에서 두 다항식을 구하는 것과 같은 계산량이 필요하다.

본 논문에서는 간단한 식으로 F'(x)를 구할 수 있는 새로운 방법을 제안하고, 구체적인 예를 적용하여 그 타당성을 보였다.

$GF(2^3)$상에서 n = 8, k = 4, t = 2인 RS부호에 대하여 F'(x)를 구해보자. 정보심볼을 U = (u_0, u_1, u_2, u_3) 이라 하면 $F(x) = u_0 + u_1x + u_2x^2 + u_3x^3$ 이 되며, 부호벡터는

$$V=(F(0),F(1),F(\alpha),F(\alpha^2),F(\alpha^3),F(\alpha^4),F(\alpha^5),F(\alpha^6))$$
$$=(v_0,v_1,v_2,v_3,v_4,v_5,v_6,v_7)$$

이 된다. V' = V + e의 수신된 벡터 V'를

$$V'=(v_0',v_1',v_2',v_3',v_4',v_5',v_6',v_7')$$

라 하면, v'로 부터

$$F'(x) = v_0'M(x)/x + v_1'M(x)/(x-1) + v_2'M(x)/(x-\alpha) + \cdots + v_7'M(x)/(x-\alpha^6)$$

윗식에서 M(x)를 x, x-1, $x-\alpha$, ⋯, $x-\alpha^6$로 각각 나누는 과정에서 계산량이 많이 필요하다. 한편,

$$M(x)/x = x^7 + 1$$
$$M(x)/(x+1)=x^7+x^6+x^5+x^4+x^3+x^2+x$$
$$M(x)/(x+\alpha)=x^7+x^6+\alpha^2x^5+\alpha^3x^4+\alpha^4x^3+\alpha^5x^2+\alpha^6x$$
$$\vdots \qquad \vdots$$
$$M(x)/(x+\alpha^6)=x^7+\alpha^6x^6+\alpha^5x^5+\alpha^4x^4+\alpha^3x^3+\alpha^2x^2+\alpha x$$

으로 이 계산과정은 많은 계산양을 필요로 하는데 그 결과를 나타내면 표2와 같다.

단, j의 각 행은 $GF(2^m)$상에서 $M(x)/(x+\alpha^i)$의 계산 결과 다항식의 계수값이다.

표 2. $GF(2^3)$상에서 $M(x)/(x+\alpha^i)$의 결과
Table 2. Results of $M(x)/(x+\alpha^i)$ over $GF(2^3)$.

x^i i :	7	6	5	4	3	2	1	0
j : 0	1							1
1	1	1	1	1	1	1	1	
2	1	α	α^2	α^3	α^4	α^5	α^6	
3	1	α^2	α^4	α^6	α	α^3	α^5	
4	1	α^3	α^6	α^2	α^5	α	α^4	
5	1	α^4	α	α^5	α^2	α^6	α^3	
6	1	α^5	α^3	α	α^6	α^4	α^2	
7	1	α^6	α^5	α^4	α^3	α^2	α	

표2에서 제 1행은 x^7-1, 제 2행은 $(\alpha^0)^i \bmod 7$, 제 3행은 $(\alpha^1)^i \bmod 7$,..
제 8행은 $(\alpha^7)^i \bmod 7$ 이 되므로 이를 식으로 나타내면 다음식과 같다.

$$P(x)=(x^7-1)+\sum_{j=1}^{7}\left\{x^7+\sum_{i=0}^{6}(\alpha^{(j-1)i} \bmod \alpha^7)x^{7-(i+1)}\right\}$$

P(x)를 수신 벡터 V'에 적용하면 다음식(25)가 성립한다.

$$F'(x)=v_0''x^7-1)+\sum_{j=1}^{7}v_j'\left\{x^7+\sum_{i=0}^{6}{}'\alpha^{(j-1)i} \bmod \alpha^7)x^{7-(i+1)}\right\} \quad (25)$$

n=16인 경우 F'(x)를 구하기 위하여 $M(x)/x$, $M(x)/(x+1),\cdots,M(x)/(x+\alpha^{14})$를 계산한 결과 그 각각의 계수는 표3과 같다.

표3에서 제 1행은 $x^{15}-1$, 제 2행은 $(\alpha^0)^i \bmod 15$, 제 3행은 $(\alpha^1)^i \bmod 15$,..
제 16행은 $(\alpha^{15})^i \bmod 15$가 되므로 이를 식으로 나타내면 다음과 같다.

$$P'(x)={}'x^{15}-1)+\sum_{j=1}^{15}\left\{x^{15}+\sum_{i=0}^{14}{}'\alpha^{(j-1)i} \bmod \alpha^{15})x^{15-(i+1)}\right\}.$$

표 3. $GF(2^4)$상에서 $M(x)/(x+\alpha^i)$의 결과
Table 3. Results of $M(x)/(x+\alpha^i)$ over $GF(2^4)$.

x^i i:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
j:0	1															1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
2	1	α^1	α^2	α^3	α^4	α^5	α^6	α^7	α^8	α^9	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}	
3	1	α^2	α^4	α^6	α^8	α^{10}	α^{12}	α^{14}	α	α^3	α^5	α^7	α^9	α^{11}	α^{13}	
4	1	α^3	α^6	α^9	α^{12}	1	α^3	α^6	α^9	α^{12}	1	α^3	α^6	α^9	α^{12}	
5	1	α^4	α^8	α^{12}	α	α^5	α^9	α^{13}	α^2	α^6	α^{10}	α^{14}	α^3	α^7	α^{11}	
6	1	α^5	α^{10}	1	α^5	α^{10}	1	α^5	α^{10}	1	α^5	α^{10}	1	α^5	α^{10}	
7	1	α^6	α^{12}	α^3	α^9	1	α^6	α^{12}	α^3	α^9	1	α^6	α^{12}	α^3	α^9	
8	1	α^7	α^{14}	α^6	α^{13}	α^5	α^{12}	α^4	α^{11}	α^3	α^{10}	α^2	α^9	α	α^8	
9	1	α^8	α	α^9	α^2	α^{10}	α^3	α^{11}	α^4	α^{12}	α^5	α^{13}	α^6	α^{14}	α^7	
10	1	α^9	α^3	α^{12}	α^6	1	α^9	α^3	α^{12}	α^6	1	α^9	α^3	α^{12}	α^6	
11	1	α^{10}	α^5	1	α^{10}	α^5	1	α^{10}	α^5	1	α^{10}	α^5	1	α^{10}	α^5	
12	1	α^{11}	α^7	α^3	α^{14}	α^{10}	α^6	α^2	α^{13}	α^9	α^5	α	α^{12}	α^8	α^4	
13	1	α^{12}	α^9	α^6	α^3	1	α^{12}	α^9	α^6	α^3	1	α^{12}	α^9	α^6	α^3	
14	1	α^{13}	α^{11}	α^9	α^7	α^5	α^3	α	α^{14}	α^{12}	α^{10}	α^8	α^6	α^4	α^2	
15	1	α^{14}	α^{13}	α^{12}	α^{11}	α^{10}	α^9	α^8	α^7	α^6	α^5	α^4	α^3	α^2	α	

P(x)를 수신 벡터 V'에 적용하면, 다음과 같이 오류정정을 위한 다항식 F'(x)를 구할 수 있다.

$$F'(x)=v_0''x^{15}-1)+\sum_{j=1}^{15}v_j'\left\{x^{15}+\sum_{i=0}^{14}{}'\alpha^{(j-1)i} \bmod \alpha^{15})x^{15-(i+1)}\right\}. \quad (26)$$

n=8일때 표2를 적용하여 F'(x)를 구한 결과 식(25)와 같고, N=16일때 표3을 적용하여 오류정정을 위한 다항식 F'(x)를 구한 결과는 식(26)과 같았다.

일반적으로 F'(x)는 식(27)과 같이 간단한 수식으로 표현할 수 있다.

$$F'(x)=v_0'(x^{n-1}-1)+\sum_{j=1}^{n-1}v_j'\left\{x^{n-1}+\sum_{i=0}^{n-2}(\alpha^{(j-1)i} \bmod \alpha^{15})x^{n-(i+1)}\right\} \quad (27)$$

식(27)을 이용하여 RS부호의 개선된 복호절차를 나타낸 결과는 그림1과 같다.

그림1에서 n, k, t, u는 각각 부호어, 정보비트, 정정가능한 오류의 수, 정보기호를 나타낸다. M(x)와 F'(x)의 최대공약수를 구하는 과정에서 $r_j(x)$와 $t_j(x)$가 구해진다. 만일 $\deg[r_j(x)] \geq (k+t)$이면 j를 1씩 증가시키면서 $\deg[r_j(x)] < (k+t)$일때까지 과정을 반복한다. $\deg[r_j(x)] < (k+t)$일때 $r_j(x)/t_j(x)$의 결과로 정보심볼이 구해진다.

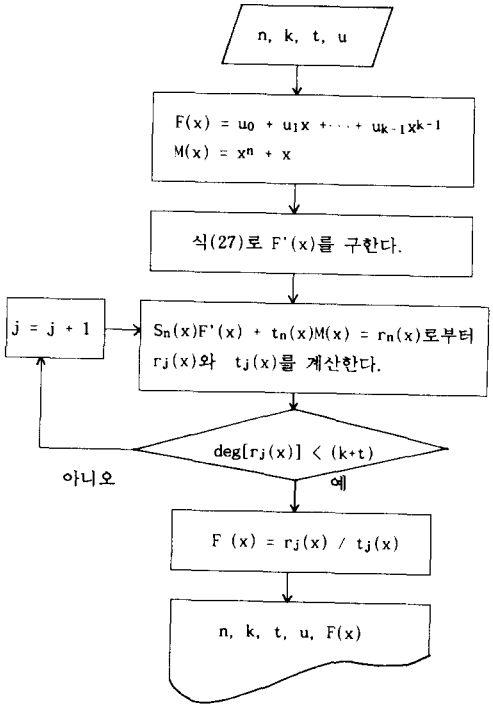

그림 1. 제산방법을 이용한 개선된 RS부호의 복호 알고리듬

Fig. 1. Improved decoding algorithm on RS codes using division method.

V. 분석 및 고찰

GF(2^3)상에서 k=4, t=2인 RS부호에서 네개의 정보심볼을 U = ($\alpha^4, \alpha^5, \alpha^2, \alpha$)이라하면, F(x) = α4+ α^5x+ α^2x^2+ αx^{3}이고, 부호벡터는 V = ($\alpha^4, \alpha^5, \alpha^3, \alpha^3, \alpha^6, \alpha^2, \alpha^4, \alpha^4$)이다. 수신된 벡터를 V' = ($\alpha^4, \alpha^3, \alpha^3, \alpha^3, \alpha^6, \alpha^4, \alpha^4, \alpha^4$)와 같이 두개의 산발오류가 발생했다면, M(x) = x^8 + x 이고,

$$F'(x) = \alpha^4(x^8+x)/x + \alpha^3(x^8+x)/(x-1) + \alpha^3(x^8+x)/(x-\alpha) + \alpha^3(x^8+x)/(x-\alpha^2) + \alpha^6(x^8+x)/(x-\alpha^3) + \alpha^4(x^8+x)/(x-\alpha^4) + \alpha^4(x^8+x)/(x-\alpha^5) + \alpha^4(x^8+x)/(x-\alpha^6)$$

으로 구해진다. 식(27)을 이용하여 F'(x)를 구하면 다음 식이 성립한다.

$$F'(x) = \alpha^4x^7 + \alpha^3x^6 + x^4 + \alpha^6x^3 + x^2 + \alpha^2x + \alpha^4.$$

윗 식을 구하는 과정에서 알 수 있드시 A. Ahiozaki의 방법에서 필요하였던 많은 계산을 수행하지 않고 간단히 F'(x)를 구할 수 있었다.

계산된 F' (x)와 M(x)에 대하여 제산방법을 반복적으로 적용하여 부호벡터를 구한다. 즉, M(x) / F' (x)에서

$Q_1(x) = \alpha^3x + \alpha^2$,

$r_1(x) = \alpha^5x^6 + \alpha^3x^5 + \alpha x + \alpha^6$

이 된다. 여기서 deg [r1(x)] 는 k+t보다 작지 않으므로 제산과정을 계속한다. F' (x)/r1(x)에서

$Q_2(x) = \alpha^6x + 1$,

$r_2(x) = \alpha^3x^5 + \alpha^2x^4 + \alpha^2x^3 + \alpha^3$

이고, deg [r2(x)] < 6 이므로 제산과정을 끝낸다.

한편, 식(1)에서 s(x) = M(x), r(x) = F'(x)라 놓으면,

$r_1(x) = M(x) - F'(x)Q_1(x)$

$r_2(x) = F'(x) - r_1(x)Q_2(x)$

$= F'(x)\{1+ Q_1(x)Q_2(x)\} - Q_2(x)M(x)$이므로

$\{1 + Q_1(x)Q_2(x)\}F'(x) - Q_2(x)M(x) = r_2(x)$

가 되며, $t_2(x) = 1+ Q_1(x)Q_2(x)$, $s_2(x) = Q_2(x)$이므로

$F(x) = r_2(x)/t_2(x) = \alpha^4 + \alpha^5x + \alpha^2x^2 + \alpha x^3$가 된다.

따라서, 두개의 오류가 정정된 U = ($\alpha^4, \alpha^5, \alpha^2, \alpha$)을 얻을 수 있다. 이 경우 컴퓨터 시뮬레이션을 수행하기 위하여 입력값은 GF(2^4)상에서 각 정보기호를 벡터로 표현하여 처리 하였다. 즉 1 = (1000), α = (0100), α^2 = (0010), ……, α^{14} = (1001)와 같이 나타내어 입력값을 1100011000100100으로 하였고, M(x)는 식(16), F' (x)는 새로 제안된 식(27)을 적용하였다. 기본적인 제산과정은 D. E. Knuth의 알고리듬에 근거하였으며[3], 프로그램언어는 BASIC을 사용하였다. 전체적인 복호과정의 흐름도는 그림1과 같으며 처리결과 출력값은 1100011000100100으로 입력값과 동일한 값으로 오류가 정정된 결과를 얻을 수 있었다.

Berlekamp의 반복알고리듬, Chien의 직접복호법, A. Shiozaki의 복호법, 그리고 본 논문에서 제안된 방법을 수행하기 위한 시간의 요구도는 표4와 같다.

표 4. 복호알고리듬에 대한 수행시간의 비교

Table 4. Comparison of execution time for decoding algorithm.

복호 알고리듬	Berlekamp의 반복알고리듬	Chien의 직접복호법	A. Shiozaki 복호알고리듬	본 논문의 알고리듬
복호지연시간	2m · (n+4t)	m · (n+1)	3mn	2m · (n-t)

표4에서 알 수 있드시 복호지연시간은 Chien의 직접복호법이 m · (n+1)으로 Berlekamp의 2m ·

(n+4t)와 A. Shiozaki의 3mn 보다 적게 걸리나 승산과 제산에 필요한 항들을 ROM으로 처리하기 때문에 유한체가 커짐에 따라 ROM의 용량이 커져야 되는 단점이 있다.

본 논문에서 제안한 방법은 수행시간이 2m · (n-t)로 Berlekamp의 반복알고리듬과 A. Shiozaki의 방법보다 우수하고, Chien의 직접복호법에 근접하며, 복잡한 계산을 수행하기 위한 별도의 하드웨어가 필요없으므로 보다 신뢰성 있는 자료의 저장과 전송 시스템에서 성능향상을 기대할 수 있다.

Ⅵ. 결 론

본 논문에서는 GF(2^m)상에서 제산방법을 체계적으로 고찰하고, 제산방법을 이용한 RS부호의 새로운 복호방법을 제안하였다. 제안된 방법의 장점은 수신된 벡터 V'로부터 오류정정을 위한 다항식 F'(x)를 구하는 과정에서 별도의 복잡한 계산을 수행하지 않고 간단한 수식으로 구할 수 있다는 점이다.

그리고 제안된 방법을 RS부호의 복호과정에 직접 적용하여 다항식 F'(x)와 다항식 M(x)에 제산과정을 적용하여 나머지 $r_j(x)$의 차수가 정보의 길이와 정정가능한 오류수의 합보다 작을때까지 반복적으로 적용함으로써 효율적으로 복호가 수행됨을 확인하고 기존의 방법과 비교분석하였다. 그 결과 본 논문에서 제안된 방법은 오류위치다항식과 오류추정다항식이 생략될 수 있고, 또한 기존의 방법과는 달리 생략된 두 다항식을 구하는 것과 같은 양의 계산량도 필요없이 간단한 식으로 처리하였으므로 복잡한 계산을 수행하기 위한 하드웨어가 필요없고, 복호과정이 훨씬 단축될 수 있으며, 이 과정을 복호기 설계에 직접 적용함으로써 신뢰성 있는 정보의 전송및 자료저장 시스템에 성능향상이 있을 것으로 기대된다.

參 考 文 獻

[1] Man Young Rhee, "Error Correcting Coding Theory", McGraw-Hill Publishing Co. 1989.

[2] R. P. Brent and H. T. Kung, "Systolic VLSI arrays for GCD computation", *IEEE Trans. on Computers*, vol. 33, no. 8, PP. 731-736, 1984.

[3] Donald E. Knuth, "The art of computer programming", vol. 2, Seminumerical algorithms, Addison-Wesley Publishing Company, 1981.

[4] D. M. Mandelbaum, "On iterative arrays for the Euclidean algorithm over finite field", *IEEE Trans. on Computers*, vol. 38, no. 10, PP. 1473-1478, 1989.

[5] H. Okano and H. Imai, "A construction method of high-speed decoders using ROM 's for BCH and RS codes", *IEEE Trans. on Computers*, vol. 36, no. 10, PP. 1165-1171, 1987.

[6] H. M. Shao, T. K. Truong, L. J. Deutsch, J. H. Yuen and I. S. Reed, "A VLSI design of a pipeline Reed-Solomon decoder", *IEEE Trans. on Computers*, vol. 34 no. 5, PP. 393-403, 1985.

[7] Akira Shiozaki, "Decoding of redundant residue polynomial codes using Euclid's algorithm", *IEEE Trans.* IT. vol. 34, no. 5, PP. 1351-1454, 1988.

[8] L. R. Welch and R. A. Scholtz, "Continued fractions and Berlekamp's algorithm", *IEEE Trans.* IT. vol. 25, no. 1, PP. 19-27, 1979.

[9] R. J. Mceliece, " The theory of information and coding", Addison-Wesley Publishing Company, 1977.

[10] 정제홍, 박진수, "유크리트 알고리듬을 이용한 Reed-Solomon부호의 복호", 대한 전자공학회 하계종합학술발표대회 논문집, PP. 15-17, 1990.

[11] Je Hong Jeong, Jin Soo Park, " A decoding method on Reed-Solomon codes by division algorithm", Joint Technical Conference on Circuits/Systems, computers and Communications, PP. 548-552, 1990.

著 者 紹 介

鄭 濟 洪(正會員)

1955年 11月 16日生. 1978年 공주대학교 이학사 (수학). 1984年 한양대학교 공학석사(전산학). 1993年 청주대학교 공학박사(전자공학). 1984年 ~ 1993년 2월 혜전전문대학 전산학과, 1993年 3月 ~ 현재 건양대학교 전산학과 조교수. 주관심분야는 컴퓨터 알고리듬, 부호이론 등임.

朴 鎭 秀(正會員)

1948年 8月 20日生. 1975年 한양대학교 공학사(전자공학). 1977年 한양대학교 공학석사(전자통신). 1985年 한양대학교 공학박사(전자통신). 1987年 1月 ~ 1988年 1月 Univ. of Colorado at Colorado Spring (Post Doc.). 1978年 2月 ~ 현재 청주대학교 전자공학과 교수. 1988年 2月 ~ 현재 청주대학교 산업과학연구소장. 1992年 2月 ~ 현재 IEEE Daejeon Section Chairman, 주관심분야는spread spectrum, 부호이론 등임.