

최대 비트슬립 정정범위를 가지는 복합 버스트 동기/에러 검출 시스템

正會員 최 양 호*

Combined Burst Synchronization/Error Detection Systems Maximizing Bit Slip Correction Ranges

Yang-Ho Choi* *Regular Member*

요 약

그동안 버스트 동기 및 에러의 검출을 위한 코셋부호의 설계 및 검출방법은 기준시점으로 부터 좌우에 발생한 슬립을 찾는다는 개념에 기초해 왔다. 본 논문에서는 이의 개념을 일반화하여 관찰구간에 하나의 부호어만이 존재할 조건으로 부터 코셋부호를 설계하는 새로운 방법과 관련 이론을 제시하였다. 제시된 이론에 의거 최대 슬립 정정 범위를 갖는 버스트 동기 및 에러 검출 시스템을 제안하고, 한 번의 CRC(Cyclic Redundancy Code)디코딩을 통해 버스트 동기 및 에러를 검출하는 방법을 제시하였다. 제안된 시스템에서 채널에러로 인해 실제 전송된 부호어가 아닌 다른 부호어로 오검출할 확률을 유도하고, 컴퓨터 계산을 통해 그 성능을 분석하였다.

ABSTRACT

Conventionally the decoding methods and the design of coset codes for burst synchronization and error detection have been based on the concept that slips occurring to the right or to the left with respect to a reference timing are corrected. In this paper we newly approach to the design of coset codes relying on the condition that only a single code word can exist in an observation interval, which provides an extended view on the conventional approach. A theorem concerning the condition is presented. A combined burst synchronization and error detection system with maximum slip correction capability have been devised based on the theorem and a detection method is presented which requires CRC decoding only once. A burst different from the actually transmitted one may be falsely accepted in the presence of channel errors. The false acceptance probabilities of the system are derived and its performance is analyzed through computer computation using the derived results.

*동양대학교 전자제어통신공학부
論文番號:97051-0210
接受日字:1997年 2月 10日

I. 서 론

순환부호(cyclic code)를 에러 검출 뿐만아니라 버스트 동기에도 이용하면 이를 위한 별도의 동기필드가 필요치 않아 전송 효율을 높일 수 있는 이점이 있다. 이러한 복합 검출 시스템에서는 순환부호에 마커(marker)를 더한 코셋부호(coset code)를 이용하여 버스트 동기가 어긋나면 이를 알 수 있도록 하고 있다. 코셋부호를 이용한 버스트 동기화 방식은 데이터 전송시스템에서 2진 비트로 표현되는 문자 또는 숫자의 타임밍의 설정, 유지에 적용될 수 있으며, 최근에 효율적인 전송을 위해 이를 적용한 TDMA 개인통신시스템^{1,2}이 제안되었다.

그동안 버스트 동기 및 에러 검출에 문제에 대한 접근 방법은 부호어(code word)의 수신에 예상되는 어떤 시점을 기준 시점으로 삼아 이로부터 왼쪽 또는 오른쪽 방향으로 발생한 비트슬립을 검출한다는 개념으로부터 출발하여 코셋부호의 설계 및 버스트 동기 검출 방법이 연구되어 왔다^{1,3,4}. 이러한 개념에 의거, (n, k) 순환 부호어의 첫 번째와 마지막 비트(n 번째 비트)를 반전시킨 코셋부호를 이용하여 슬립이 일어난 방향과 크기를 알아내고 있다. 여기서 n 은 부호어 길이, k 는 메세지 길이이다. 이러한 2 비트 반전시스템은 기준 시점으로 부터 좌우 $(n-k-2)/2$ 비트 이하의 슬립이 발생한다면 정정이 가능하다.

본 논문에서는 기준 시점으로 부터 좌우에 발생한 비트 슬립을 검출 또는 정정한다는 개념을 보다 일반화 시켜 코셋 부호어가 하나만 존재할 수 있는 관찰구간의 크기를 구하고 이 관찰구간에 있는 부호어를 찾는 접근 방법을 취하고 있다. 전송에러 없이 수신된 부호어를 정확히 검출하기 위해서는 관찰 구간내에 하나의 부호어만이 있어야 함은 당연하다. 만약 2 개 이상의 부호어가 존재할 수 있다면 에러 없이 수신됐더라도 전송된 부호어를 정확히 찾아낼 수 없게 된다. 하나의 부호어만이 존재할 수 있는 관찰 구간과 관련된 이론을 제시하고, 제시된 이론에 기초하여 최대 비트슬립 정정범위를 갖는 버스트 동기 및 에러 검출 시스템을 제안하였다. 제안된 시스템에 대한 오검출 확률을 유도하여 differential encoding을 한 경우와 안한 경우에 대해 각각 그 성능을 분석하였으며, 위에서 언급한 2 비트 반전시스템과 비교하였다.

II. 최대 비트슬립 정정범위를 갖는 시스템

부호어의 길이가 n 이고 메세지의 길이가 k 인 (n, k) 순환부호 다항식(polynomial) $C(X)$ 는

$$C(X) = c_0 X^{n-1} + c_1 X^{n-2} + \dots + c_{n-2} X + c_{n-1} \quad (1)$$

와 같이 $n-1$ 차 이하의 다항식으로 표현된다. 여기서 c_i 는 부호어의 $i+1$ 번째 비트를 나타낸다. systematic 부호에서는 c_0, \dots, c_{k-1} 의 메세지 필드와 c_k, \dots, c_{n-1} 의 CRC 필드로 구성된다. 수신된 부호어를 부호 생성 다항식(generator polynomial) $g(X)$ 로 나누어 나머지 즉 신드롬을 계산하면 채널 에러가 발생했는지를 알 수 있으며, 부호어의 일부 비트를 반전시킨 코셋부호를 이용하면 이에 더하여 버스트 동기의 검출도 가능하다.

1. 코셋부호의 설계

코셋 부호어 $Q(X)$ 는 순환 부호어 $C(X)$ 에 마커(marker) $M(X)$ 를 더하여 생성된다. 코셋부호를 이용한 버스트 동기의 검출에 있어, 전의 결과는 기준시점(또는 기준버스트)으로 부터 비트 슬립이 일어난 방향과 크기를 검출한다는 관점으로 부터 출발하고 있다. 본 논문에서는 이의 개념을 일반화하여 단지 하나의 코셋 부호어가 존재할 구간의 크기를 구하고 이를 효율적으로 찾는 접근방법에 기초하고 있다. 버스트의 위치를 찾아내고, 찾아낸 버스트에 에러가 발생했는지를 알기 위해서는 관찰구간에 부호어가 있다면 하나만이 존재해야한다. 만약 관찰구간에 두 개의 부호어가 존재할 수 있다면 채널 에러 없이 수신됐더라도 전송된 버스트를 항상 정확히 찾아내는 것은 불가능하게 된다. 주어진 코셋부호에 대해 부호어가 하나만 존재할 관찰구간의 크기에 관한 조건을 구하자.

코셋 부호어 $Q(X)$ 와 s 비트만큼 어긋난 벡터 U_s 에 대한 다항식 $U_s(X)$ 는 일반적으로 다음과 같이 표현할 수 있다.

$$U_s(X) = X^s Q(X) + A_s(X) \quad (2)$$

s 가 +일 경우, U_s 는 $(q_s, q_{s+1}, \dots, q_{n-1}, b_0, b_1, \dots, b_{s-1})$ 와 같다. 여기서 q_i 는 $Q(X)$ 의 $i+1$ 번째 비트이고 b_i 는 0 또는 1의 값을 가지는 랜덤비트를 나타낸다. $X^m = 1$

인 관계가 있으며, $A_s(X)$ 는 최고차가 $s-1$ 이하인 임의의 다항식(arbitrary polynomial)으로

$$A_s(X) = a_0 X^{s-1} + a_1 X^{s-2} + \dots + a_{s-2} X + a_{s-1} \quad (3)$$

와 같이 주어진다. 여기서 $a_i (= b_i + q_i)$ 는 0 또는 1의 값을 가지는 랜덤변수를 나타낸다. s 가 -1 일 경우, U_s 는 $(b_0, b_1, \dots, b_{|s|-1}, q_0, q_1, \dots, q_{n-|s|-1})$ 와 같으며, 이 때 임의의 다항식 $A_s(X)$ 는

$$A_s(X) = a_0 X^{n-1} + a_1 X^{n-2} + \dots + a_{|s|-2} X^{n-|s|-1} + a_{|s|-1} X^{n-|s|} \quad (4)$$

이고, $a_i (= b_i + q_{n-|s|+i})$ 는 0 또는 1의 값을 가지는 랜덤변수이다. $M(X)$ 의 적절한 선택에 의해 s 의 어떤 범위내에서 어느 $A_s(X)$ 에 대해서도 $U_s(X)$ 를 코셋 부호에 속하지 않게 할 수 있으며, 이 범위는 선택된 $M(X)$ 에 의해 결정된다. $U_s(X)$ 가 코셋 부호어가 아니라면 $U_s(X) + M(X)$ 의 신드롬은 0이 아니므로

$$\{X^s(C(X) + M(X)) + A_s(X) + M(X)\} = \{(X^s + 1)M(X) + A_s(X)\} \neq 0, \quad s > 0, \quad (5.1)$$

$$= \{X^s[(X^s + 1)M(X) + A_{-s}(X)]\} \neq 0, \quad s < 0 \quad (5.2)$$

이다. 여기서 $\{C(X)\}$ 는 $D(X)$ 의 신드롬을 나타낸다. 식 (5.1)이 0이 아니면 식 (5.2)도 0이 아니며, 거꾸로 식 (5.2)가 0이 아니면 식 (5.1)도 0이 아니다. 이는 부호어로 부터 어느 한 방향으로 $|s|$ 비트 만큼 떨어진 벡터가 부호어가 아니라면 반대 방향으로 $|s|$ 비트 만큼 떨어진 벡터도 부호어가 아님을 의미한다. 주어진 $M(X)$ 에 대해 식 (5.1)을 만족하는 양수 s 의 최대값을 s_{max} 라 하자. 동기 에러의 발생 여부만 조사한다면, 기준시점으로 부터 양쪽 방향으로 일어난 s_{max} 비트슬립 이하의 동기 에러 검출이 가능하다. 코셋 부호어를 중심으로 어느 한 방향으로 s_{max} 비트 이하 만큼 떨어진 벡터는 부호어가 아니므로 오른쪽 또는 왼쪽에 s_{max} 비트를 추가하여 구성된 $n + s_{max}$ 비트 블록에는 하나의 부호어만이 존재한다. 그림 1에서와 같이 길이 $n + s_{max}$ 의 관찰블럭에 버스트의 위치를 오른쪽부터 순서대로 위치 0, 위치 1, ..., 위치 s_{max} 이라 하자. 부호어가 위치 0에 있다면 앞에서 설명한 대로 관찰블럭에

는 하나의 부호어(위치 0에 있는)만이 존재한다. 부호어가 위치 1에 있는 경우에도 관찰블럭 내 다른 위치에 있는 버스트의 신드롬을 구하면 모두 식 (5)에 따라 0이 아님을 쉽게 알 수 있고 따라서 관찰블럭에는 이 부호어의 다른 부호어는 존재하지 않게 된다. 부호어가 위치 2, 위치 3, ..., 위치 s_{max} 에 있을 때에도 관찰블럭은 각 부호어를 중심으로 좌우에 s_{max} 개 이하의 비트를 추가하여 구성되어 단지 하나의 부호어만이 존재하게 된다. 결론적으로 이러한 사실은 다음의 정리를 말해준다.

정리 : 어떤 주어진 마커 $M(X)$ 에 대해 양수 s_{max} 가 식 (5.1)을 만족하는 최대 $s (= s_{max})$ 라면 $n + s_{max}$ 개의 비트로 구성된 블록에는 단지 하나의 코셋 부호어 $Q(X) (= C(X) + M(X))$ 만이 존재할 수 있다. $n + s_{max}$ 개 보다 많은 비트를 가진 블록에는 두 개 이상의 부호어가 존재할 수 있다.

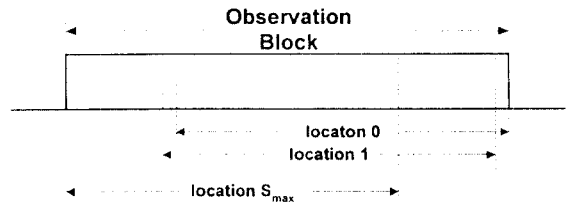


그림 1. 관찰블럭에서 부호어의 위치
Fig. 1 Locations of a code word in an observation block

위의 정리에 따라 하나의 코셋 부호어가 존재할 수 있도록 관찰구간의 크기를 결정하여 이 사실을 이용하면 기준시점으로 부터 좌우에 일어난 비트 슬립을 찾는다는 개념에 기초한 기존의 검출 방법^{1, 3, 4}을 일반화시킬 수 있다. 예를 들어 $M(X)$ 가 $X^{n-1} + 1$ 일 때, 식 (5.1)에 이를 대입하면

$$\{(X^s + 1)(X^{n-1} + 1) + A_s(X)\} = \{X[X^{n-1} + X^s + X A_s(X) X + 1]\} \neq 0, \quad s > 0 \quad (6)$$

와 같이 구해진다. 식 (6)에서 $[\]$ 안 다항식은 최고차가 $s+1$ 로, 길이가 $n-k$ 이하인 0이 아닌 버스트는 순환 부호어일 수 없으므로, 이 최고차가 $n-k-1$ 이

하이먼 어떤 임의 다항식 $A_s(X)$ 에 대해서도 식 (6)을 만족하게 된다. 따라서 s_{max} 는 $n-k-2$ 이며, $2n-k-2$ 비트 블록에 부호어가 있다면 단지 하나만 존재하게 된다. 참고문헌 [1]에 따르면 양쪽 방향으로 똑같이 $(n-k-2)/2$ 비트 이하의 슬립의 검출할 수 있으나, 제시된 정리의 결과는 한쪽 방향으로 $d(0 \leq d \leq n-k-2)$ 비트슬립까지 검출한다면, 다른 방향으로 $n-k-2-d$ 비트슬립까지 검출할 수 있음을 보여준다. [1]의 방법은 $d=(n-k-2)/2$ 인 특별한 경우로 이 방법에서는 관찰블럭의 중앙에 위치한 버스트를 기준버스트로 삼아, 슬립이 일어난 방향을 알아야하나 d 가 0 또는 $n-k-2$ 인 경우에는 기준버스트로부터 한 방향으로만 슬립이 일어나기 때문에 이를 알 필요가 없다.

위의 이론에 기초하여 최대 s_{max} 를 갖는 코셋부호를 설계할 수 있다. 식 (5.1)에서 $\{ \}$ 안 다항식의 최고차는 어떤 $M(X)$ 에 대해서도 s 차 이하로 될 수 없으며 따라서 s 차 일 때 버스트 동기 및 에러의 검출범위를 최대로 할 수 있다. $M(X)=1$ 일 때 $\{ \}$ 안의 다항식의 최고차는 s 가 되며 이 때 식 (5.1)은

$$\{(X^s + 1) + A_s(X)\} \neq 0, \quad s > 0 \tag{7}$$

이고, $\{ \}$ 안의 다항식의 최고차 s 가 $n-k-1$ 이하이면 그 신드롬은 0이 아니게된다. 따라서 최대 s_{max} 는 $n-k-1$ 로 주어진다. $M(X)=X^{n-1}$ 일 때도 $s_{max}=n-k-1$ 임을 쉽게 확인할 수 있다. $M(X)=1$ 또는 X^{n-1} 인 코셋부호를 이용하면 $2n-k-1$ 비트 블록내에 부호어는 하나만 존재할 수 있게 된다.

참고문헌 [3]에서는 코셋부호의 설계를 다음 식에 의거하고 있다.

$$\{(X^s + X^w)M(X) + A_s(X) + A_w(X)\} \neq 0 \tag{8}$$

이 식은 기준버스트로부터 좌우에 각기 발생한 s, w 비트 슬립에 대해 신드롬이 달라야 한다는 조건으로부터 유도되었다. 식 (8)에 $M(X)=1$ 을 대입하면

$$\{(X^s + X^w + A_s(X) + A_w(X)\} \neq 0 \tag{9}$$

와 같이 주어진다. s, w 가 모두 음수인 경우(-1 이더라도) 식 (9)의 $\{ \}$ 안 다항식은 0이 될 수 있으며, 이

수식에 의거하여 최대 s_{max} 를 갖는 코셋부호를 설계할 수 없다.

2. 버스트 동기 및 에러의 검출

$M(X)=1$ 인 코셋부호를 이용하면, 송신측에서 부호어 C_T

$$C_T = (c_0, c_1, \dots, c_{n-2}, c_{n-1}) \tag{10}$$

가 전송된다. 앞에서 설명한 바와 같이 $M(X)=X^{n-1}$ 인 코셋부호를 이용하더라도 최대 슬립 정정범위를 얻을 수 있지만 메세지블럭의 첫번째 비트를 반전시킨 관계로 메세지 블럭을 복원하기 위해서는 수신측에서 다시 이 비트를 반전시켜야 하는 복잡함이 따른다. 이 전송 부호어는 수신측에서 설정한 $2n-k-1$ 비트의 관찰블럭 내에 수신된다고 가정한다. 만약 이 블럭 밖에 까지 부호어가 걸쳐 있다면 에러 없이 수신됐더라도 정확히 검출할 수 없다. 버스트 동기와 에러의 복합 검출 방법은 이 $2n-k-1$ 비트 관찰블럭내의 부호어를 효율적으로 찾는 문제로 볼 수 있으며, 다음 두가지 방법에 의해 찾을 수 있다.

한가지 방법은 먼저 버스트 위치를 알아내고 이 버스트에 에러가 있는지 조사하는 방법이다. $2n-k-1$ 비트 블록에 가장 오른쪽(그림 1에서 위치 0)에 있는 버스트의 동기신드롬을 구하여 이로부터 부호어의 위치를 알아 낼 수 있다. 가장 오른쪽에 있는 버스트를 기준으로 삼았기 때문에 비트슬립은 항상 이 버스트로부터 왼쪽으로 발생하게 된다. 동기신드롬은 부호어에 위치에 따라 표 1의 꼬리신드롬 패턴과 같이 주어진다. 여기서 x 는 0 또는 1 이든지 상관치 않는 비트를 나타낸다. 동기신드롬 패턴에서 연속된 0 비트 다음, 처음 나타나는 1의 비트로부터 부호어 위치를 알 수 있다. 예를 들어 i 번째 비트가 처음으로 나타난 1이라면 버스트는 위치 $n-k-i$ 에 있다. 알아낸 위치에 있는 버스트를 다시 CRC 디코딩하여 에러가 있는지를 검사한다. 이 방법은 두 번의 디코딩을 필요로 하여 구현이 복잡해지고 처리 시간이 길어지는 단점이 있다.

참고문헌 [5]에 제시된 방법을 적용, 한 번의 CRC 디코딩을 통해 머리, 꼬리 신드롬을 계산하여 버스트의 검출이 가능하다. 비트 시퀀스 ... $v_{-1} v_0 v_1 \dots$ 이 수

신된다고 하자. $2n-k$ 비트로 구성되는 수신블럭 R 을 생각하자.

$$R = (\gamma_{-(n-k)}, \gamma_{-(n-k-1)}, \dots, \gamma_{-1}, \gamma_0, \dots, \gamma_{n-2}, \gamma_{n-1}) = (\gamma_h, \gamma_m, \gamma_t) \quad (11)$$

여기서 γ_h 는 R 에서 앞의 $n-k$ 비트로, γ_m 은 다음 k 비트로, γ_t 은 뒤의 $n-k$ 비트로 구성되는 블럭이다. 전송 버스트는 식 (12)의 관찰블럭

$$(\gamma_{-(n-k-1)}, \gamma_{-(n-k-2)}, \dots, \gamma_{-1}, \gamma_0, \dots, \gamma_{n-2}, \gamma_{n-1}) \quad (12)$$

에 있다고 가정하자. 이 때 마킹된 부호어의 마지막 비트(\bar{c}_{n-1})는 항상 꼬리 블럭에 놓이게 된다. 참고문헌 [5]에서는 첫 번째 비트와 마지막 비트를 마킹한 코셋 부호를 사용한 관계로 머리블럭에도 마킹된 비트가 있으며 관찰블럭 크기가 본 논문의 경우보다 1 비트 작은 것이 차이점이다. [5]에 제시된 방법을 근간으로 한 다음의 알고리즘으로부터 버스트 동기와 에러를 검출할 수 있다.

- 1) 중간블럭 γ_m 오른쪽에 $n-k$ 개의 0 비트를 첨가 하여 구성된 n 비트의 동기검출 블럭을 CRC 디코딩하여 신드롬 s_r 을 구한다.
- 2) 이 신드롬에 각각 γ_h 와 γ_t 를 더하여 머리신드롬 패턴 s_h 와 꼬리신드롬 패턴 s_t 를 구한다. 즉,

$$\begin{aligned} s_h &= \gamma_h + s_r, \\ s_t &= \gamma_t + s_r \end{aligned} \quad (13)$$

- 3) 구한 머리신드롬과 꼬리신드롬 모두가 표 1에 보인 동기신드롬 패턴과 일치하면 채널에러 없이 성공적으로 버스트동기의 위치를 검출한 것으로 간주되어 검출된 위치정보에 의해 메세지 블럭이 출력된다. 만약 동기신드롬 패턴과 일치하지 않는다면 에러가 발생한 것으로 처리한다.

부호어가 관찰블럭내 위치 d 에 있다면, $\gamma_h, \gamma_m, \gamma_t$ 는

$$\begin{aligned} \gamma_h &= (\gamma_{-(n-k)}, \gamma_{-(n-k-1)}, \dots, \gamma_{-(d+1)}, c_0, c_1, \dots, c_{d-1}), \\ \gamma_m &= (c_d, c_{d-1}, \dots, c_{d+k-1}), \end{aligned} \quad (14)$$

$$\gamma_t = (c_{d+k}, c_{d+k+1}, \dots, \bar{c}_{n-1}, \gamma_{n-d}, \gamma_{n-(d-1)}, \dots, \gamma_{n-1})$$

와 같이 주어져서 표 1과 같이 꼬리신드롬과 머리신드롬이 구해짐을 쉽게 확인할 수 있다. 부호어가 위치 d 에 있을 때, 꼬리신드롬은 앞에서 부터 $n-d-1$ 개의 연속되는 0 다음에 처음 1이 나타난다. 머리신드롬은 버스트의 위치에 관계없이 같은 패턴을 가질 수 있어 이로 부터 그 위치를 알 수 없으며 단지 검출된 버스트가 부호어 인지를 검사한다고 볼 수 있다. 부호어라면 머리신드롬은 뒤로 부터 최소 d 개의 연속하는 0을 가진다.

표 1. 최대 슬립 정정범위를 갖는 시스템에서의 동기신드롬 패턴

Table 1. Timing syndroms for the system maximizing slip detection capability

부호어 위치	동기신드롬 패턴	
	꼬리신드롬	머리신드롬
0	0000...00001	xxxx...xxxx
1	0000...0001x	xxxx...xxx0
2	0000...001xx	xxxx...xx00
3	0000...01xxx	xxxx...x000
.	.	.
.	.	.
.	.	.
n-k-3	001x...xxxxxx	xxx00...0000
n-k-2	01xx...xxxxxx	xx000...0000
n-k-1	1xxx...xxxxxx	x0000...0000

그림 2는 최대 비트슬립 정정범위를 갖는 복합 검출 시스템의 기능 구성도이다. 수신블럭은 버스트 동기가 검출될 때까지 저장된다. 디코더에 중간블럭이 입력되면 CRC 디코딩이 시작된다. 디코더에 중간블럭이 전부 입력되면 $n-k$ 개의 0 비트가 입력되어 중간블럭에 $n-k$ 개의 0 비트를 더한 블럭에 대한 신드롬을 계산한다. SYNCH/ERROR LOGIC은 이 신드롬에 STORE부로부터 읽은 머리블럭과 꼬리블럭을 더하여 꼬리, 머리신드롬을 구한다. 메세지블럭은 쉬프트 레지스터를 이용하여 다음과 같이 쉽게 복원될 수 있다. 꼬리, 머리신드롬은 각각 $n-k$ 의 크기를 갖는 머리, 꼬리신드롬 레지스터에 저장되며 첫 번째 비트가 가장 왼쪽에, 마지막 비트가 가장 오른쪽에 있다고 상정한다. 꼬리신드롬이 구해지면, STORE부는 수신블럭의 두 번째 비트($\gamma_{-(n-k-1)}$)부터 출력하기 시작

하여 매 클럭마다 한 비트씩 출력하고, 동시에 꼬리신드롬 레지스터는 왼쪽으로 쉬프트되면서 왼쪽 비트부터 순차적으로 출력한다. MESSAGE DETECTOR는 꼬리신드롬 레지스터의 출력 비트를 조사하여 출력 비트가 1이면 이 때부터 STORE부로부터 출력된 비트를 수신하여 메시지블럭을 복원하기 시작한다. 동시에 SYNCH/ERROR LOGIC은 머리신드롬 레지스터를 오른쪽으로 쉬프트되면서 뒤로부터 연속된 0의 개수를 카운트하여 메시지블럭에 에러 있는지를 알아낸다. 꼬리신드롬 비트를 전부 조사했으나 1인 비트가 발견되지 않았을 경우, 전송에러가 있는 것으로 처리된다. $n-k$ 번째에서 처음으로 1인 비트가 발생했다면 전송에러가 없는 것으로 처리되고 이 경우, 머리신드롬 패턴의 조사는 필요치 않다.

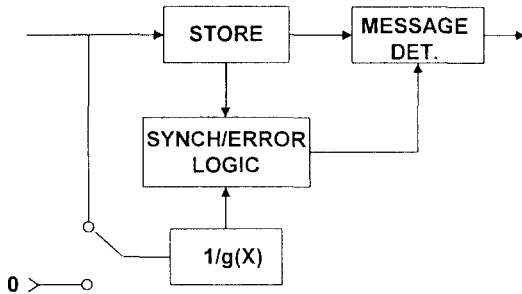


그림 2. 하나의 CRC 디코더를 가진 복합 검출 시스템의 블럭도

Fig. 2 Block diagram of a combined receiver with a single CRC decoder

Ⅲ. 성능 분석

관찰블럭 내에 실제 전송 버스트가 위치 s 에 위치하나, 수신기는 위치 \hat{s} 에 부호어가 있는 것으로 검출했다고 가정하자. 이때 오검출 확률 $\Pr(f_a)$ 는 다음과 같이 표현할 수 있다.

$$\begin{aligned} \Pr(f_a) &= \sum_s \Pr(f_a|s) \Pr(s) \\ &= \sum_s \sum_{\hat{s}} \Pr(\hat{s}|s) \Pr(s) \end{aligned} \quad (15)$$

여기서 $\Pr(s)$ 는 전송버스트가 관찰블럭내 위치 s 에 있

을 확률로 시스템 특성에 종속된 값이며 $\Pr(\hat{s}|s)$ 는 전송버스트가 s 에 있으나 \hat{s} 에 있는 것으로 검출될 확률로 전송에러 특성에 의해 결정된다. 식 (12)의 관찰블럭에서 \hat{s} 으로 검출된 블럭을 C_R 이라 하자. 예를들어 $\hat{s}-s$ 가 3이고 s 가 0이라면 C_R 은 전송버스트와 워드 에러벡터 성분으로 분해하여 아래와 같이 표현된다.

$$\begin{aligned} C_R &= (r_{-3}, r_{-2}, r_{-1}, r_0, r_1, \dots, r_{n-4}) \\ &= (r_{-3}, r_{-2}, r_{-1}, c_0, c_1, \dots, c_{n-4}) \\ &\quad + (0, 0, 0, e_0, e_1, \dots, e_{n-4}) \end{aligned} \quad (16)$$

여기서 e_i 는 c_i 에 대한 에러 비트를 나타낸다. $\Pr(\hat{s}|s)$ 를 구하기 위해서는 C_R 을 코셋 부호어로 만드는 모든 에러패턴을 구하는 것이 필요하다. 랜덤에러 채널에서 i 개의 전송에러가 발생할 때 그 확률이 P_i 이고 이러한 에러패턴이 N_i 개 있다고 하자. 각각의 에러패턴의 발생은 서로 독립적이므로

$$\Pr(\hat{s} = s + 3|s) = 2^{-3} \sum_{i=1}^{n-3} N_i P_i \quad (17)$$

로 주어진다. 2^3 으로 나눈 이유는 검출된 블럭이 코셋 부호어이기 위해서는 3개의 랜덤비트 r_{-3}, r_{-2}, r_{-1} 은 어떤 특정한 값을 가져야 하기 때문이다. 식 (16)에서 C_R 의 마지막 비트를 반전시키고 3비트를 왼쪽으로 쉬프트 한 벡터의 신드롬은 0이므로 에러벡터의 신드롬은

$$\{(e_0, e_1, \dots, e_{n-4}, 0, 0, 0)\} = (0, 0, \dots, 0, 1, y_1, y_2, y_3) \quad (18)$$

와 같이 주어지며, y_l ($l=1, 2, 3$)은 랜덤비트를 나타낸다. N_i 는 i 개의 전송에러가 발생할 때 식 (18)을 만족하는 워드 에러패턴의 수이다. 전송버스트가 위치 0이 아닌 다른 곳에 있더라도 $\Pr(\hat{s} = s + 3|s)$, 에러벡터 신드롬은 각각 식 (17), (18)과 똑같이 주어짐을 쉽게 확인할 수 있다. 전송버스트 중 뒤의 3 비트, $c_{n-3}, c_{n-2}, c_{n-1}$ 에서의 에러 발생여부는 실제 전송버스트의 위치가 s 일때 $s+3$ 으로 오검출될 확률과 무관하다.

$\hat{s}-s$ 가 -3 인 경우도 N_i, P_i 값은 다를지 몰라도 $\Pr(\hat{s} = s - 3|s)$ 는 식 (17)과 같은 형태로 표현된다. 이때 N_i 는 다음식을 만족시키는 에러패턴의 수이다.

$$\{(e_3, e_4, \dots, e_{n-1}, 0, 0, 0)\} = (0, 0, \dots, 0, 1, y_1, y_2, y_3) \quad (19)$$

differential encoding을 하지않은 경우, 워드 에러패턴은 채널에러 패턴과 동일하게 된다. 식 (18)과 식 (19)를 비교하면 $|\hat{s}-s|$ 가 같으면 N_i 가 같게 되어 결국 $\Pr(\hat{s}|s)$ 는 동일하게 된다. 그러나 differential encoding을 한 경우, 하나의 채널에러가 부호어의 마지막 번째를 제외하고 두 비트에 영향을 주므로 $\hat{s}-s$ 의 부호에 따라 N_i 는 다를 수 있다. 문제를 간단히 하기 위해 편의상 differential 디코딩은 완벽하다고 가정한다. $\hat{s}-s$ 가 -3일 때 c_3 에서의 에러는 이 비트 자체의 에러뿐만 아니라 바로 이전에 전송된 비트에서의 에러에 의해 발생될 수 있으므로 이에 대한 조사가 필요하다. 그러나 $\hat{s}-s$ 가 3인 경우 검출된 부호어에서 에러 조사가 필요한 첫 번째 비트(식 (16)에서 r_0)는 첫 번째 전송비트에 대응하기 때문에 이전의 전송비트에 에러가 발생했는지 조사할 필요가 없다. 따라서 $\hat{s}-s=-3$ 인 경우, 길이 $n-2$ 인 버스트에서 채널 에러패턴의 조사가 필요하여 $n-3$ 길이의 버스트에서 채널 에러패턴의 조사가 필요한 $\hat{s}-s=3$ 인 경우보다 N_i 가 크거나 같다.

differential encoding을 하지 않았을 경우 $|\hat{s}-s|=m$ 이거나, differential encoding을 했을 경우 $\hat{s}-s=m$ 일 때 $\Pr(\hat{s}|s)$ 은

$$\Pr(\hat{s}=s-m|s) = 2^{-m} \sum_{i=1}^{n-m+1} N_i p^i (1-p)^{n-m-i}, \quad 0 \leq m \leq n-k-1 \quad (20)$$

와 같이 표현된다. 여기서 Binary Symmetric Channel (BSC)채널을 가정하였으며 p 는 BER이다. N_i 는 i 개의 채널에러가 발생할 때 식 (21)과 같은 신드롬을 갖는 워드 에러패턴의 수이며, differential encoding을 안한 경우와 한 경우에 대해 그 값은 서로 다를 것이다.

$$(0, 0, \dots, 0, 1, y_1, \dots, y_m), \quad 1 \leq m \leq n-k-1 \quad (21)$$

$m=0$ 인 경우는 버스트 위치는 제대로 검출했으나 이 검출된 버스트에 에러가 있는 경우로 이 때 워드 에러벡터의 신드롬은 $(0, 0, \dots, 0, 0)$ 과 같이 주어진다.

식 (20)은 2 비트 반전시스템에 대한 오검출 확률에 대한 표현식⁵과 유사하나, 검출범위가 1 비트 확장되어 있고 에러벡터 신드롬이 달라 N_i 값이 다르게된다. differential encoding을 했을 경우 $\hat{s}-s=-m$ 일 때 $\Pr(\hat{s}|s)$ 은 다음과 같이 주어진다.

$$\Pr(\hat{s}=s-m|s) = 2^{-m} \sum_{i=1}^{n-m+1} N_i p^i (1-p)^{n-m+1-i}, \quad 1 \leq m \leq n-k-1 \quad (22)$$

여기서 N_i 는 i 개의 채널에러가 발생할 때 다음식을 만족하는 워드 에러패턴 수이다.

$$\{(e_m, e_{m+1}, \dots, e_{n-1}, 0, 0, \dots, 0)\} = (0, 0, \dots, 0, 1, y_1, y_2, \dots, y_m) \quad (23)$$

식 (20)과 비교하면 식 (20)의 $n-m$ 이 식 (23)에서는 $n-m+1$ 으로 대치되어 있음을 알 수 있다. 이는 $n-m+1$ 의 크기를 갖는 버스트에서 채널에러의 조사가 필요하기 때문이다.

BER이 매우 작아 에러가 3개 이상 발생할 확률이 2개 이하 발생할 확률보다 훨씬 작다면 오검출 확률은 $i=1$ 과 2에 대한 합으로 근사 시킬 수 있다⁵. 그림 3, 4는 실제 전송워드의 위치가 s 로 주어졌을 때 두 개 이하의 채널에러만을 고려한 근사식을 이용하여 구한 조건 오검출 확률 $\Pr(fals)$ 을 보여준다. (161, 147) 순환부호어¹의 마지막 비트를 반전시킨 코셋 부호어를 BSC 채널로 전송한다고 상정하였다. 그림 3은 differential encoding을 안 한 경우이고 그림 4는 한 경우이며, 각각에 대한 에러패턴의 수를 표 2, 3에 나타내었다. 표 3에서 에러패턴 수는 $\hat{s}-s$ 가 양수일 때 음수인 경우보다 같거나 작음을 확인할 수 있다. 부호어의 마지막 1 비트만이 반전되어 있어 하나의 채널에러가 발생하더라도 오검출 될 수 있음을 보여준다. differential encoding을 한 경우와 안 한 경우에 대한 오검출 확률은 거의 같다. 두 경우에 대해 1개의 채널에러에 의한 에러패턴 수는 비슷하여 이와 같은 결과를 나타낸다. BER이 작아 오검출확률은 채널에러가 한 개 발생한 경우에 의해 거의 결정되고, $|\hat{s}-s|$ 대해서는 2의 지수승으로 감소하므로 비교적 큰 $|\hat{s}-s|$ 에서 N_i 의 차이의 성능에 대한 영향은 미미하기 때문

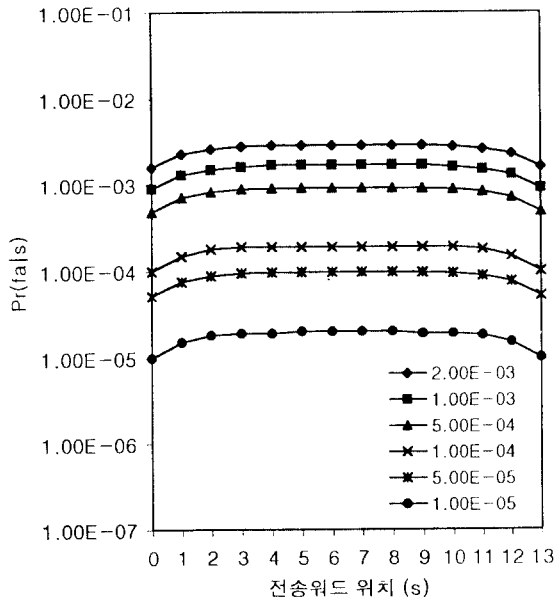


그림 3. 최대 슬립 정정범위를 갖는 시스템의 조건 오검출 확률 (differential encoding을 안한 경우)

Fig. 3 Conditional false acceptance probabilities for the system maximizing slip correction capability (no differential encoding)

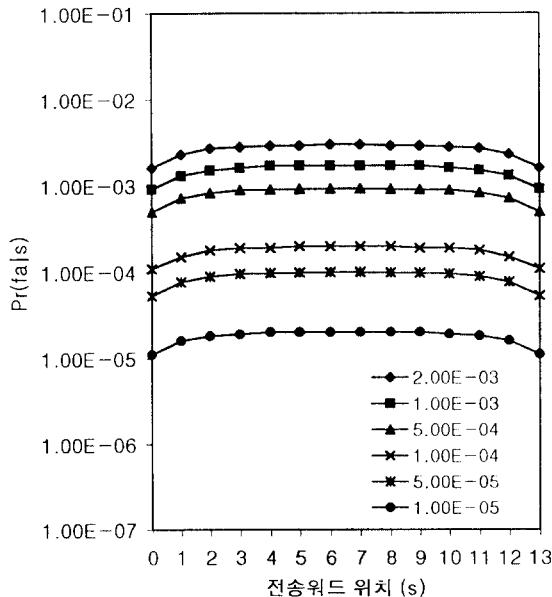


그림 4. 최대 슬립 정정범위를 갖는 시스템의 조건 오검출 확률 (differential encoding을 한 경우)

Fig. 4 Conditional false acceptance probabilities for the system maximizing slip correction capability (differential encoding)

표 2. 최대 슬립 정정범위를 갖는 시스템에서 오검출을 일으키는 에러패턴 수(differential encoding을 안한 경우)

Table 2. The numbers of error patterns to cause false detection in the system maximizing slip detection capability (with no differential encoding)

$ \hat{s}-s $	0	1	2	3	4	5	6	7	8	9	10	11	12	13
N_1	0	1	1	1	1	1	1	1	3	4	8	20	38	80
N_2	0	5	10	15	20	31	28	85	178	334	656	1400	2774	5440

표 3. 최대 슬립정정 범위를 갖는 시스템에서 오검출을 일으키는 에러패턴 수(differential encoding을 한 경우)

Table 3. The numbers of error patterns to cause false detection in the system maximizing slip detection capability (with differential encoding)

$\hat{s}-s$	1	2	3	4	5	6	7	8	9	10	11	12	13	
N_1	1	1	1	1	1	2	2	2	4	8	22	40	76	
N_2	0	50	10	21	32	62	99	173	338	668	1402	2800	5448	
$\hat{s}-s$	0	1	2	3	4	5	6	7	8	9	10	11	12	13
N_1	0	1	1	1	1	1	2	2	2	4	8	22	40	75
N_2	0	0	5	10	21	31	62	99	171	334	654	1384	2764	5475

이다. 그림 3, 4의 결과는 BER에 비례하는 경향을 나타내어 1개의 채널 에러에 의해 오검출확률은 거의 결정된다는 사실을 뒷받침하고 있다. $|\hat{s}-s|$ 이 작은 값에서 에러패턴수는 1로 같아, 실제 위치로부터 1비트 왼쪽 또는 오른쪽으로 슬립이 발생된 것으로 잘못 검출할 확률이 다른 크기의 슬립으로 검출될 확률보다 크게 된다. 양 가장자리 위치($s=0, 13$)에서는 이 중 한쪽 방향으로만 오검출이 발생할 수 있기 때문에 그 확률은 최소로 된다.

부호어의 첫번째와 마지막 비트의 2비트를 반전하는 기존의 시스템과 최대 슬립 정정범위를 갖는 시스템과의 성능 비교를 그림 5에 나타내었다. 두 시스템 모두 differential encoding을 한다고 상정하였으며, 표 5는 2비트 반전시스템에서의 에러패턴 수를 보여준다. 2비트 반전시스템은 제안된 시스템보다 1비트 정정범위가 작으며, 식 (12)의 관찰블럭에서 r_{n-1} 비트를 제외한 $2n-k-2$ 비트 블럭내에 있는 전송 버스트를 복원한다고 상정하였다. 최대 슬립 정정범위를 갖는 시스템은 하나의 채널에러가 발생해도 인접

한 곳에 있는 버스트를 전송버스트로 잘못 검출할 수 있으나 2 비트 반전시스템에서는 최소한 8 비트 떨어진 지점에 있는 버스트가 검출되는 관계로 그 성능이 우수하다.

IV. 결 론

본 논문에서는 코셋부호를 이용한 버스트 동기 및 에러 검출에 있어, 기준시점으로 부터 좌우에 발생한 비트슬립을 정정한다는 개념에 기초한 종래의 접근 방법과 다른 접근방법을 제시하였다. 관찰구간에 코셋 부호어가 하나만 존재할 이론을 제시하였으며, 이 이론에 의거 코셋부호를 설계하고, 관찰구간 내에 있는 부호어를 찾는 접근방법을 취한다. 이러한 접근방법은 코셋부호 설계와 버스트 동기 검출 방법에 있어 종래의 방법을 일반화할 수 있는 효과를 준다.

제시된 이론을 이용하여 최대의 슬립 정정 범위를 갖는 복합 버스트 동기 및 에러 검출 시스템을 제안하였다. 제안된 시스템은 단지 한 번의 CRC 디코딩을 요구할 뿐만 아니라 메시지 복원까지 어떤 비트 반전도 필요치 않아 구현이 매우 간단해진다. (161, 147) 순환부호를 제안된 시스템에 적용하여 성능을 분석한 결과, differential encoding을 한 경우와 안한 경우에 대해 거의 동일한 오검출 성능을 나타내었다. 성능 분석에서 differential 시스템에서 differential 디코딩은 완벽하다고 가정하였으나, 실제 시스템에서는 differential 디코딩을 위해 시작 비트의 추가가 필요할 것이다. 최대 슬립 정정범위를 갖는 시스템은 2 비트를 반전하는 종래의 시스템에 비해 성능은 다소 떨어지나 슬립 정정범위를 확장시킬 수 있고 하드웨어 구현이 간단하여 전송에러가 작은 채널에서 매우 유용할 것으로 생각된다.

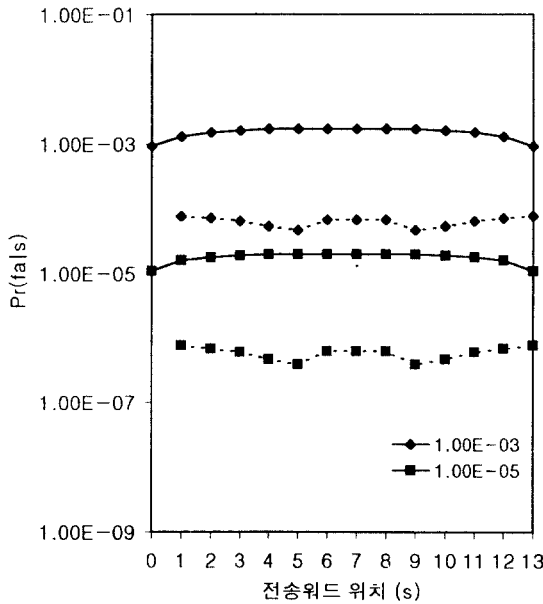


그림 5. 최대 슬립 정정범위를 갖는 시스템(실선)과 2 비트 반전 시스템(점선)간 성능 비교

Fig. 5 Performance comparison between the system maximizing slip correction capability (solid line) and the 2-bit inverting system (dotted line)

표 4. 기존의 2 비트 반전 시스템에서 오검출을 일으키는 에러패턴 수 (differential encoding을 한 경우)

Table 4. The numbers of error patterns to cause false detection in the conventional system with the first and last bits inverted (with differential encoding)

$\hat{s}-s$	-1	-2	-3	-4	-5	-6	-7	-8	-9	-10	-11	-12	
N_1	0	0	0	0	1	0	0	2	4	14	18	35	
N_2	7	6	11	12	33	39	75	169	338	758	1436	2825	
$\hat{s}-s$	0	1	2	3	4	5	6	7	8	9	10	11	12
N_1	0	0	0	0	0	1	0	0	2	4	14	18	35
N_2	0	6	5	10	11	32	37	73	167	334	750	1414	2785

참 고 문 헌

1. L. F. Chang, N. R. Sollenberger and S. Ariyarisitakul, "Performance of a TDMA portable radio system using a cyclic block code for burst synchronization and error detection," IEEE Trans. Commun., vol. COM-41, pp. 22-31, Jan. 1993.
2. A. R., Noerpel, et. al., PACS: Personal Access Communications Systems-A Tutorial, IEEE Pers. Commun., pp. 32-43, June, 1996.
3. S. E. Trvares and M. Fukada, "Furtuer results on the synchronization of bynary cyclic codes," IEEE Trans. Inform. Theory, vol. IT-11, pp 238-241,

Mar. 1970.

4. S. Y. Tong, "Synchronization recovery techniques for binary cyclic codes," Bell Syst. Tech. J., vol. 45, pp. 561-596, Apr. 1966.
5. 최양호, "순환코드를 이용한 효율적 버스트 동기/에러 검출 방법 및 성능 분석," 한국통신학회 논문지 제21권 8호, 1996년 8월.



최 양 호(Yang-Ho Choi) 정회원

1982년 2월:연세대학교 전자공학과(공학사)

1984년 8월:한국과학기술원 전기 및 전자공학과(공학석사)

1989년 2월:한국과학기술원 전기 및 전자공학과(공학

박사)

1989년 3월~1997년 2월: 한국통신 선임연구원

1997년 3월~현재: 동양대학교 전자제어통신공학부 조교수

※주관심분야: 이동통신, 무선접속기술, 적용안테나