

리드-솔로몬과 Convolutional 코드에 의한 Concatenated 코딩 시스템

한 원 섭, 강 창 언
연세대학교 전자공학

A Concatenated Coding System by the (7,3) Reed-Solomon and (2,1,6) Convolutional Codes

Won Soub HAN and Chang Eon Kang
Graduate school of Engineering, Yonsei University

ABSTRACT For the purpose of error correcting, a concatenated coding system has been proposed by cascading two codes-(7,3) Reed-Solomon and (2, 1,6) convolutional codes.

As a result of the computer simulation and the experiment, the (98,21) concatenated code has been shown to be able to correct 12 random errors and 16 burst errors.

When the channel error is about 1.2×10^{-2} , this system indicates most efficient.

1. 서 론

디지털 통신 시스템에서 정보 전송 중 에러가 많이 발생하는 경우에 에러정정 능력이 큰 코드가 요구된다. 즉, 최소거리가 큰 코드가 요구되어 지는데, 최소거리가 커짐에 따라서 코드의 길이가 길어져 설계상의 복잡성이 증가하게 된다. 따라서 구조가 간단하면서도 여러 정정 능력이 우수한 복합 코드에 대한 방법들이 연구되어 졌다.

복합 코드는 1954년 Elias에 의해 처음으로 두 개의 단일 코드를 이용한 곱셈코드(product code)가 소개된 이후¹⁾ 활발한 연구가 진행되어, 1965년 Forney에 의해 2진 코드와 비2진 코드를 이용하여 concatenated 코드를 구성하는 방법이 소개되어 졌고²⁾, 이어 1966년에는 hybrid코드가 Falconer에 의해 제안되었다³⁾. 1968년 Abramson에 의해 순환 곱셈 코드의 디코딩 방식중 performance가 좋은 cascade 디코딩 방식이 연구되어 졌고⁴⁾, 1970년에는 Odenwalder에 의해 convolutional 코드와 비2진 코드인 Reed-Solomon 코드를 concatenation시키는 방법이 제시되어 졌다⁵⁾.

본 논문에서는 비2진 BCH 코드의 특수 형태의 에러

정정 능력이 우수한 (7,3) Reed-Solomon 코드를 외부 코드로 하고, 내부 코드인 (2,1,6) convolutional 코드를 concatenation 시켜 코드를 구성하였다.

concatenated 코드의 디코딩은 외부 코드와 내부 코드를 cascade로 연결하여 실현하였는데, 외부 디코더는 에러 트래핑(errortrapping) 방식을, 내부 디코더는 다수결논리(majority logic)방식을 적용하여 구성하였다.

시스템의 컴퓨터 시뮬레이션 및 performance를 고찰하고, 실제회로를 구성하여 동작여부의 실험을 수행하였다.

2. concatenated 코드

concatenated 코딩은 코드 길이가 짧은 코드들로부터 길이가 긴 코드를 실현하는 방법으로, 간단한 concatenate 코딩은 2단계의 concatenation을 하는 경우이다.

이와 같은 시스템에서는 외부코드는 (n_2, k_2) Reed-Solomon 코드를 사용하고, 내부코드는 (n_1, k_1, m) 의 convolutional 코드(혹은 블록코드)를 일반적으로 사용한다.

convolutional 코드를 (n_1, k_1, m) Reed-Solomon 코드를 (n_2, k_2) 라 하고 또, 두 코드의 최소 거리를 각각 d_1, d_2 라고 하면 concatenated 코드는 그림 1과 같이 구성된다. 구성된 concatenated 코드의 매개 변수는 식(1)과 같다.

$$\begin{aligned} n &= (m + 1)n_1 n_2 \\ k &= (m + 1)k_1 k_2 \\ d &= (m + 1)d_1 d_2 \end{aligned} \tag{1}$$

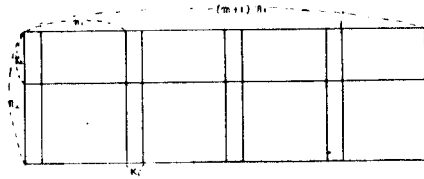


그림 1 concatenated 코드의 배열
The arrangement of concatenated code

3. (98,21) concatenated 코드 설계

내부코드를 $m=6$ 단인 (2,1) convolutional 코드를 사용하고 외부코드를 (7,3) Reed-Solomon 코드를 사용하여 cascade로 연결하여 구성되는 (98,21) concatenated 코드의 인코더와 디코더의 block diagram은 그림 2와 같다.

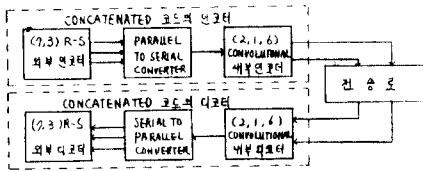


그림 (98,21) concatenated 코드의 블록도
A Block diagram of the (98,21) concatenated code

외부코더가 (7,3) Reed-Solomon 코드의 인코더이므로 인코딩 출력은 3-tuple로 나오게 된다. 따라서 내부 인코더 (2,1,6) convolutional 코드의 인코더에 인코딩시킬 때는 parallel to serial 변환이 필요하게 된다. 이와 같은 원리로 설계된 (98,21) concatenated 코드의 인코더는 그림 3과 같다.

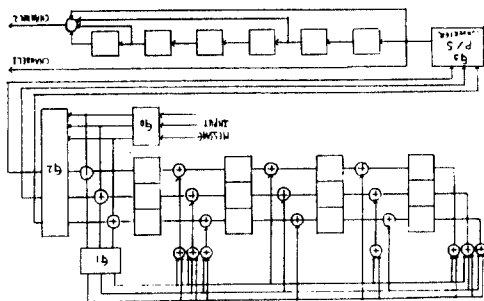


그림 3 (98,21) concatenated 코드의 인코더 회로
The encoder of (98,21) concatenated code

인코딩 과정은 3개의 정보심볼들 중 하나의 정보 심볼이 외부 인코더에 입력됨과 동시에 심볼이 3개의 비트로 변환되어 내부 인코더에 입력됨으로서

하나의 (2,1,6) 내부코드를 형성하게 된다.

이 동작이 3번 반복되면 외부 인코더에 패리티 검사 (parity check) 심볼들이 형성되고 이 4개의 패리티 검사 심볼들도 정보부분에서와 같이 parallel to serial로 변환되어 내부 인코더에 의해 각각의 (2,1,6) 내부 코드가 형성된다.

그림 2의 블록 다이어그램에서와 같이 (98,21) concatenated 코드의 디코더 설계에서는 내부 디코더에서 정성된 비트들을 심볼로 변환하여 외부 디코더에 입력시켜야 하므로 serial to parallel 변환이 필요하게 된다.

앞에서 검토된 방법으로 설계된 (98,21) concatenated 코드의 디코더를 설계하면 그림 4와 같다.

또 여러 정정을 위한 디코딩 과정은 다음과 같다.

- 1단계 : 2bit의 입력을 내부 디코더에 차례로 6번 인가한다. 이때 G1은 "OFF" 상태를 유지한다.
- 2단계 : G1을 "ON" 시킨후 2 bit의 입력을 계속 3번 cyclic shift시킨다. 이때 다수결 gate에 의해 정정된 코드는 parallel to serial 변환기를 통하여 외부 디코더의 입력에 인가된다.
- 3단계 : G2를 "HIGH"로 G3를 "ON"으로 하고 1 Symbol (3bits)을 받아들인다.

위의 1,2,3단계 과정을 7회 반복하여 외부 디코더가 7심볼을 다 받아들였으면 G2를 "LOW"로, G3를 "OFF"으로 한후 다음 단계로 간다.

4단계 : G2를 "LOW"으로 G3를 "OFF"으로 G4를 "ON"으로 하고 G5를 "OFF" 시킨후 외부 디코더의 shift 레지스터와 신드럼 레지스터의 상태에 의한 에러 트래핑 (error trapping) 방법에 의해 shift 레지스터에 저장된 정보가 1 symbol씩 정정된다.

5단계 : 4단계와 같은 gate 상태에서 다시 shift 레지스터와 신드럼 레지스터를 7번 shift 시킨후 신드럼 레지스터의 값이 "0"이 아니면 올바른 정보가 아니므로 버린다.

6단계 : G2를 "OFF", G3를 "OFF", G4를 "OFF"로 하고 G5를 "ON" 시킨후 정정된 심볼들의 정보를 G5를 통해 보낸다.

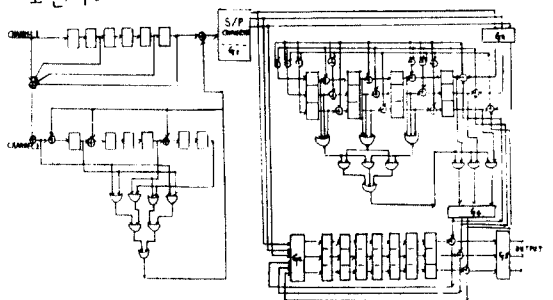


그림 4 (98,21) concatenated 코드의 디코더 회로

The decoder of (98,21) concatenated code

4. 컴퓨터 시뮬레이션 및 performance

1) 컴퓨터 시뮬레이션

앞장에서 설계된 (98,21) concatenated 코드의

인코더와 디코더의 구성에 따라 computer simulation 프로그램을 작성하였다.

Simulation 결과에 의하면 (98,21) concatenated 코드는 12개의 random 에러와 16개의 bust 에러를 정정할 수 있는 것으로 나타났다.

2) error performance

인코더 (2,1,6) convolutional 코드의 에러확률은

다음 식(2)와 같다

$$P_e = \sum_{i=1}^{\infty} \binom{q}{i} P_e^i (1-P_e)^{q-i} \quad (2)$$

(7,3) Reed-Solomon 코드를 외부코드로 구성되는

(98, 21) concatenated 디코더의 에러확률은 다음 식 (3)과 같다.

$$P = \sum_{i=1}^q \binom{q}{i} P_e^i (1-P_e)^{q-i} \quad (3)$$

(2,1,6) convolutional 코드, (7,3) Reed-Solomon 코드,

(49,9) concatenated 코드, (84,9) concatenated 코드 및 (98,21) concatenated 코드의 에러확률을 비교한 내용은 그림5와 같다.

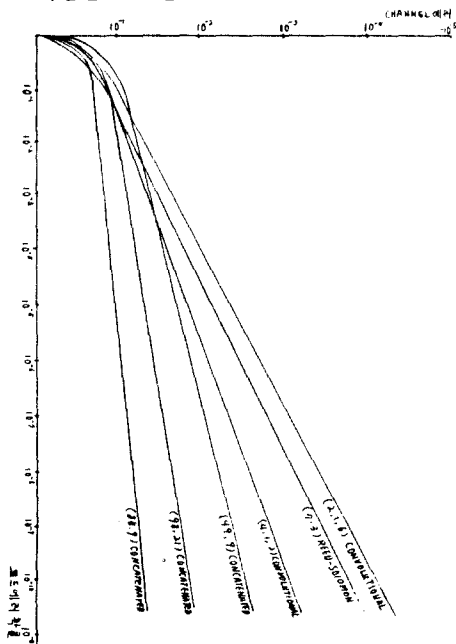


그림 5 에러확률 비교

The comparison of error performance

5. 실험

본 논문에서는 인코더와 디코더의 실험을 위한

제어장치로 그림6과 같이 micro-computer (Apple II) 를 사용하였고, interface (I/O Port) 회로는 IC8255를 사용하여 구성하였다.

제어 프로그램에 의한 인코딩 및 디코딩 수행과정은 다음과 같다.

* 마이크로 컴퓨터의 메모리 \$4000 번지에 정보

심벌(1 symbol 이 3bits)을 keyboard를 통해 저장한다.

* 정보 심벌을 I/O 1을 통해 인코더에 보낸후 인코딩된 값을 다시 I/O 1을 통해 마이크로 컴퓨터에 받아들이 36000번지에 저장한다.

* 프로그램에 의해 생성된 에러는 36000번지의 코드워드에 더해 I/O2를 통하여 디코더에 보낸다.

* 디코더에서 디코딩된 값을 다시 I/O2를 통해서 마이크로 컴퓨터에 받아들이 39000번지에 기억시킨다.

실험에서 하나의 코드워드(code word)를 인코딩하는 데는 63clock이 소요되었으며, 디코딩하는 데는 80clock이 소요되었다.

실험결과는 computer simulation결과와 동일하였다.

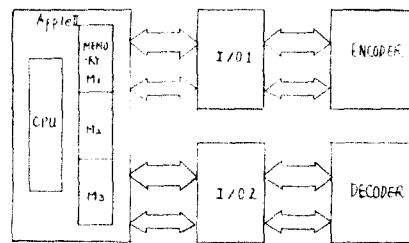


그림 6 실험구성 블록도

A block diagram of the experimental apparatus

6. 결 론

본 논문에서는 random 에러와 bust 에러를 동시에 정정할 수 있는 코드로서 (7, 3) Reed-Solomon 코드와 (2, 1, 6) convolution 코드를 결합한 concatenated 코딩 시스템을 실현하였다.

외부 코더인 (7, 3) Reed-Solomon 코드의 디코더는 에러르래핑 방법을 적용하였고, 내부 코더인 (2, 1, 6) convolutional 코드는 다수결 논리방법을 이용하였다. 외부 코더와 내부 코더의 연결은

parallel to serial과 serial to parallel 회로를 이용, cascade로 연결 (98,21) concatenated

코드의 인코더 및 디코더를 제작하였다.

(98, 21) concatenated 코드의 computer simulation과 실험 결과에 의하면 12개의 random 에러와 16개의 bust 에러를 정정할 수 있는 것으로 나타났다.

performance 계산 결과 1비트의 에러가 1.5×10^{-2} - 1.2×10^{-2} 일때 에러확률은 1.3×10^{-9} - 1.7×10^{-10} 로서, 채널 에러가 1.2×10^{-2} 인 경우에 효율적인 것으로 나타났다. 또, 종래의 단일 코드와 복합 코드와의 performance를 비교해 본 결과 에러 performance가 개선됨을 알 수 있었다. 디코더를 구성하여 실험해 본 결과 인코딩 하는 데는 63클럭이 소요되었고, 디코딩 하는 데는 80클럭이 소요되었다.

이러한 concatenated code는 우주 통신에서 매우 많은 영상전송에 응용될 수 있으며, 에러를 제어하기 위하여 외부코드인 Reed-Solomon 코드의 코드 길이나 코드율을 높이는 방향으로 연구가 진행되어져야 겠다.

참 고 문 헌

1. P.Elias, "Error Free Coding, "IRE Trans. on Information Theory, vol. PGIT-, No.6, pp.39-37, Jun, 1954.
2. G.D.Forney, "Concatenated Code, "M.I.T. Press, Massachusetts, 1966.
3. D.D.Falconer, "A Hybrid Sequential and Algebraic Decoding Schem, "Ph. D.dissertation, dep.Elec.Eng., M.I. T. Press, Cambridge, 1966.
4. N.M.Abramson, "Cascade Decoding of Cyclic Product Codes, "IEEE Trans. on Communication Technology, COM-16, pp.398-402, 1968.
5. J.P.Odenwalder, "Optimal Decoding of Codes," Ph.D. dissertation, Dep. Elec. Eng., Univ. California, Los-Angels, 1970.
6. 강창언, 김신령, "Cascad 방식을 이용한 순환 곱셈 코드의 시스템 설계, "대한 전자공학 회지, vol.22, No.5, pp.443-447, 1985, 9.
7. 강창언, 강범수, 김신령, "통신 시스템 개선

을 위한 Cocatenated Code의 연구, "연세대학, 산업 기술 연구소 논문집, vol.18, No.2, 1985, 12.

8. 강창언, 홍태식, "Error-Trapping 방법을 이용한 Reed-Solomon 코드의 인코더 및 디코더 설계, "대한 전자 공학회지, vol.7 pp.101-104, 1984,11.