

선박용 화재탐지장치의 통신 에러를 감소시키기 위한 수정된 터보코딩 알고리즘 개발에 관한 연구

정병홍* · 최상학** · 오종환*** · 김경석***

A Study on the Enhancement of Turbo Decoder for Reducing Communication Error of a Fire Detection System for Marine Vessels

B-H Jung* · S-H Choi** · J-H Oh*** · K-S Kim***

Key words : Communication Error(통신에러), Cyclic Redundancy Check(CRC, 순환중복검사), Turbo Code Algorithm(터보 코드 알고리즘), Very High Speed Integrated Circuit Hardware Description Language(VHDL), Fire Detection System(화재탐지장치)

Abstract

Abstract : In this study, an adapted Turbo Coding Algorithm for reducing communication error of a fire detection system for marine vessels, especially image transmission via power line. Because it is necessary that this system communicate larger and faster than previous method, this study carried out enhancement a decoding speed by adaptation CRC with Turbo Code Algorithm, improvement of metric method, and reduction of decoding delay by using of Center-to-Top method. And the results are as follows :

(1) Confirmed that a Turbo Code is so useful methods for reducing communication error in lots of noise environments.

(2) Proposed technology in this study of speed increasing method of Turbo Coding Algorithm proves 2 times faster than normal Turbo Code and communication error reducing as well in the board made by VHDL software & chips of ALTERA company.

1. 서 론

본 연구는 선박용 화재탐지장치의 개발에 있어,

현재 선박에 설치되어 있는 화재탐지장치가 탐지기 및 탐지센서 등의 잦은 오동작으로 경보가 발령되어도 당직자가 이를 무시해 버리거나 화재사실

* 이엔씨테크놀로지(원고접수일 : 2000년 11월)

** (주)씨플러스

*** 한국해양수산연수원 기관공학과

을 직접 현장까지 가서 확인하는 과정을 거치는 등으로 결국 초기소화의 기회를 잃고 능동대응으로 막대한 손실을 초래하는 단점을 보완하기 위하여 해당 탐지구역의 탐지센서가 작동하면 화재현장의 영상이미지를 화재탐지장치에서 원격으로 모니터링할 수 있도록 하였다. Fig. 1은 본 연구에서 채택한 선박에서의 전력선통신을 이용한 영상모니터링 시스템의 구성예를 나타낸 것이다. 이렇게 영상을 모니터링함에 있어 별도의 통신선을 추가하여 설치하지 않고 전원선에 화재에 관한 데이터 및 영상신호를 동시에 전송하는 방식을 채택하였다.

일반적으로 데이터통신에 있어서 통신 오류를 줄이기 위하여 순환중복검사(CRC ; Cyclic Redundancy Check) 및 통신부호화의 방법을 주로 사용하고 있다. 그러나 선박과 같이 잡음이 많은 환경에서는 통신 오류에 대한 대책을 별도로 강구하는 것이 바람직하다.⁽¹⁾⁽²⁾

잡음에 대한 해결방안의 하나로 1993년 Berrou가 터보코드(Turbo-Code)에 관한 논문을 발표한 이후 1~2년간에 걸쳐 터보코드의 성능이 입증되었고, 현재는 그 적용분야에 대한 연구가 집중되고 있으며 시스템 엔지니어링 차원에서 터보코드 설계 기술이 축적/고도화되고 있는 단계이다.⁽³⁾ 지금은 무선통신, 전력선통신 등 각종 통신방식에 터보코드 알고리즘이 주로 적용되고 있다.⁽⁴⁾⁽⁵⁾ 또한, 최근에 관심사항이 되고 있는 IMT2000 시스템의 무선전송기술규격(RSPS series)이 1999년 말에 완료되어 IMT2000 시스템에 터보코드를 적용하도

록 확정되었다.⁽⁶⁾

그러나 현재의 터보코드의 응용 및 적용범위는 기존의 비터비 복호기(Viterbi Decoder)와는 달리 복잡한 연산으로 처리속도가 늦어 고속서비스에는 부적합하다. 터보코드의 고속화가 실현되면 이의 적용 범위가 IMT2000 뿐만 아니라 위성통신 및 타 이동 무선통신으로의 적용이 확대되고, 음성, 데이터, 영상 등 무선 멀티미디어 서비스로도 점차 확대될 전망이다. 터보코드는 이동통신 시스템의 오류제어기법방식으로, 비터비 복호기 등 기존의 오류제어기법을 대체할 차세대 오류제어기법으로서 향후 차세대 이동통신 시장을 고려할 때 현시점에서의 설계기술의 개발이 절실히 요구되는 분야이다.

따라서 본 연구에서는 화재탐지장치의 통신 오류를 줄이기 위하여 터보코드 알고리즘을 적용하고, 기존의 터보코드 알고리즘의 복호시간이 오래 걸리는 단점을 개선하여 시스템의 실시간 처리를 가능하게 하였다.

2. 통신오류정정부호 방식

2.1 CRC 기술

CRC(Cyclic Redundancy Check)는 순환중복검사 방식으로 송신측에서는 FCS(Frame Check Sequence) n-bit를 구한 후 이를 원래의 데이터 k-bit에 추가하여 (n+k)-bit를 송신한다.

- T=전송될 (k+n)bit Frame (n<k인 경우)
- M=k-bit 메시지, T의 첫 k개 bit
- F=n-bit FCS, T의 마지막 n-bit
- P=(n+1)bit의 Pattern, 켄수(divisor)

여기에서 정보 T를 송신한 후 수신단에서 "T"를 수신하여 "P"로 나누어 떨어지도록 하여야 한다. 따라서, 송신시 보내야 하는 데이터는 다음과 같다.

- T=(메세지의 n-bit Left Shift) + (Frame Check Sequence)
- F는 메세지를 n-bit Left Shift한 후 이를 P로 나눈 나머지이다.

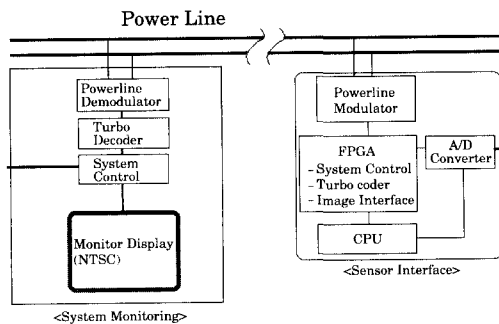


Fig. 1 Block Diagram of Image Monitoring System using Power-line Communication in Marine Vessels

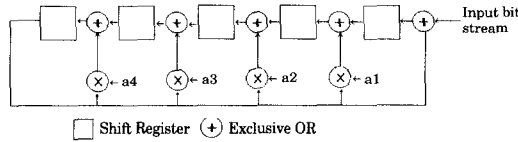


Fig 2. Configuration of Cyclic Redundancy Check reducing Communication Error

• 수신측에서는 전송된 데이터를 P로 나누어 나머지가 없어야 한다.

• Pattern P는 다음과 같다.

CRC-12=110000000111

CRC-16=1100000000000101

CRC-CCITT=100010000001000001

CRC-32=10000010011000001000111011
0110111

본 연구에서 채택한 시스템에서는 CRC-12를 적용하였다. CRC 검사 결과 나머지가 “0”이 되지 않아 오류가 발생하면 재전송을 요청한다.

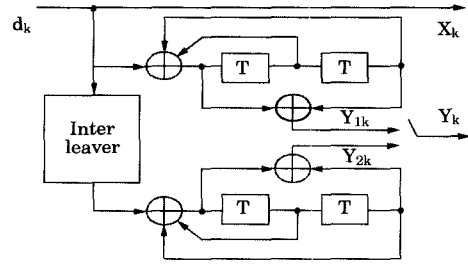
2.2 수정된 터보코드 알고리즘 개발

터보코드 기술은 기존의 코딩기술을 대체할 Shannon’s Limit에 가장 근접한 차세대 오류 정정 부호방식으로 1993년 Berrou가 발표한 방법이다.

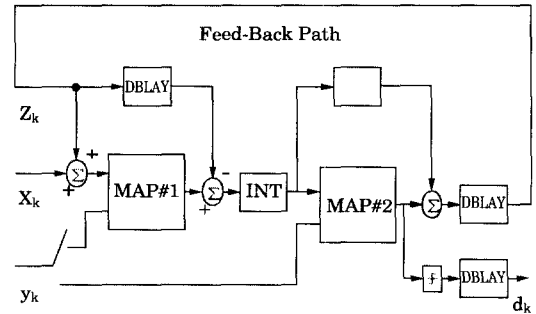
CRC에 추가로 오류를 완벽하게 정정할 수 있는 터보코드 알고리즘을 적용함으로써 완벽한 시스템을 구현할 수 있다. 터보 복호기는 MAP (Maximum a Posteriori)기반의 복호기로 잡음분산평가의 문제점도 있지만, 가장 큰 문제점은 MAP의 복잡성과 많은 연산량으로 인해 처리속도가 저속이고 메모리도 방대하고 설계가 어려우므로 다음과 같은 방법으로 속도를 증가시켰다.^(6x7x9x10x11)

2.2.1 복호속도 개선방안 : 고속 Radix-4 방식의 MAP Turbo 복호알고리즘

MAP기반의 터보코드 복호시, Radix-2 방식은 임의의 k시점에서 복호할 때 k-1 시점에서의 순방향 state metric α_{k-1}^m , k+1 시점에서 역방향 state metric β_{k-1}^m 그리고 k 시점에서 branch metric δ_k^m 를 이용하여 k 시점에서 “0” 과 “1”에 대한 LLR(Log Likelihood Ratio)을 구하여 1비



(a) Turbo Encoder



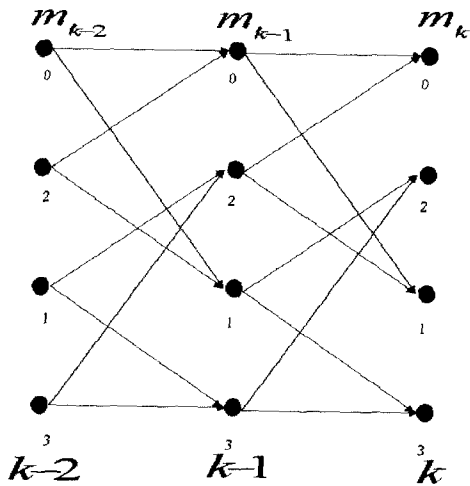
(b) Turbo Decoder

Fig. 3 Trellis Diagram of Turbo Code and Decoder

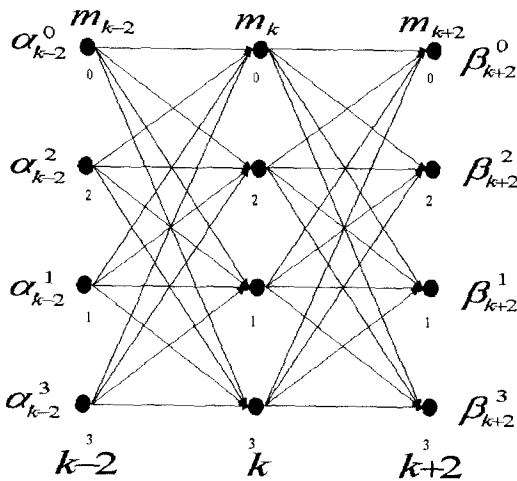
트를 복호한다. 이에 반해 radix-4방식은 두상태 이전인 k-2시점에서 α_{k-2}^m , 두상태 이후인 k+2시점에서 β_{k+2}^m 를 구하여 k시점에서 2비트를 동시에 복호하기 때문에 radix-2 방식보다 속도가 2배 빠르며, 저장되는 메모리도 절반으로 감소할 수 있다. 과거 2단의 수신 비트를 입력받아 한꺼번에 처리하는 Radix-4 방식의 trellis 구조는 Radix-2 방식의 2개 시점을 하나의 시점으로 간주하여 처리하며 Radix-2 방식에서 Radix-4방식으로의 trellis 구조변경은 다음 Fig. 4와 같다.

순방향, 역방향, BR(branch metric)을 이용하여 각 상태에서 LLR을 구하기 위한 연산은 Fig. 5와 같다.

δ^{000} 는 branch code-word 000에 대한 수신 비트의 BR을 의미하며, α_{k-2}^{000} 는 k-2 시점에서 상태 000에 대한 순방향 state metric을 의미하며, β_{k+2}^{000} 는 k+2 시점에서 상태 000에 대한 역방향 state metric을 의미하며, LLR_k 는 k 시점에서의 각 상태



(a) Structure of 4-state Radix Trellis



(b) Structure of 4-state Radix Trellis

Fig. 4 Structure of 4-state Radix Trellis

에서 "0"과 "1"이 될 확률값을 의미하며, d_k^{00} 은 k 시점에서 복호된 비트값이 "00"임을 의미한다.

실제 MAP의 구현은 Log 차원에서 수행되므로, 두 상태를 하나의 상태로 간주하여 결합하는 radix-4 구조는 branch code-word와 수신된 비트와의 해밍거리를 구하는 비터비 복호기의 구현 예처럼 단순하지가 않으며, Log차원에서 결합하기 때문에 복잡한 수식을 통하여 증명하고 난 뒤 구현해야 한다.

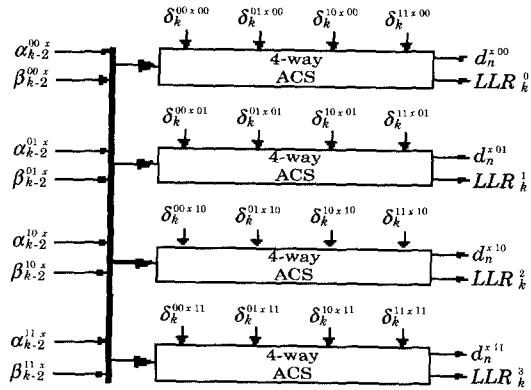


Fig. 5 Structure of Radix-4 State Metric

Log 차원에서의 radix-4 방식 적용시 다음의 항목에 대한 공식을 만들어 기존의 radix-2 방식과 성능이 동일함을 시뮬레이션을 통하여 증명하여야 한다.

- Radix-4방식의 새로운 branch metric δ_k^m 계산방법
- Radix-4방식의 새로운 순방향 state metric α_k^m 계산방법
- Radix-4방식의 새로운 역방향 state metric β_k^m 계산방법
- Radix-4방식의 Iteration시 새로운 extrinsic Z_k 계산방법
- Radix-4방식의 복호비트를 결정하기 위한 새로운 LLR 계산방법

2.2.2 연산방식 개선방안 : 고속 덧셈/곱셈 알고리즘

Radix-4 구조로의 병행과 더불어 고속화를 위해서는 각 상태 metric과 순방향, 역방향 state metric, LLR을 구하기 위한 고속 덧셈 알고리즘도 병행하여야 한다. 고속 덧셈 알고리즘으로는 Carry Save Adder, Carry Select Adder, Carry Lookahead Adder가 있으며 기존의 Full Adder보다 계산속도가 빠른 것이 이미 증명되었다.

(1) Carry Save Adder

- 2-operand에 대한 Carry Save Adder

$$S_i = A_i \oplus B_i$$

$$C_i = A_i \cdot B_i$$

• 3-operand에 대한 Carry Save Adder

$$S_i = A_i \oplus B_i \oplus C_i$$

$$C_i = A_i \cdot B_i + C_i \cdot (A_i \oplus B_i)$$

(2) Carry Look ahead Adder

$$G_i = A_i \cdot B_i \text{ (Carry Generation)}$$

$$P_i = A_i \oplus B_i \text{ (Carry Propagation)}$$

$$S_i = (A_i \oplus B_i) \oplus C_{i-1} = P_i \oplus C_{i-1}$$

$$C_i = A_i \cdot B_i + (A_i \oplus B_i) \cdot C_{i-1} = G_i + P_i \cdot C_{i-1}$$

(3) Carry Select Adder

전체 Adder를 몇 개의 Section으로 나눈 후 각 Section별로 Carry가 '0'인 경우와 '1'인 경우를 미리 연산하고, 하위 비트 Section의 Carry에 해당하는 상위 비트의 Section 출력을 선택하여, 하위 비트의 Section과 상위 비트의 Section의 조합으로 값을 출력하는 Adder이다.

2.2.3 복호지연 개선방안 : Center-to-Top 방식

한 클럭에 2비트를 복호하는 radix-4 알고리즘과는 달리 1비트만 복호하면서도 복호지연을 절반으로 감소하는 알고리즘을 제시한다. 여기서 제시한 알고리즘은 터보 복호기의 구현시 문제점은 메모리 크기 및 복호지연이 가장 큰 문제점인데 메모리 크기에 비례하여 복호지연이 발생하므로 메모리를 절반으로 줄이면서 복호지연 및 복호속도를 2배 향상시킨 알고리즘이다.

다음 Fig. 6은 제시한 고속 터보 복호기의 복호 과정에 대한 그림이다. 여기서 α 와 β 를 동시에 구한 다음 메모리 절반 위치에 있을 때 복호를 한다.

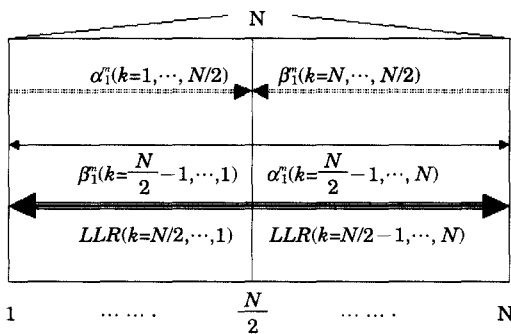


Fig. 6 Decoding Process of the Center to Top Algorithm

이렇게 함으로써 복호시점이 역방향 상태 메트릭을 모두 구한 다음 복호를 하는 기존의 방식과는 달리 복호 메모리의 절반 위치에서 복호를 하기 때문에 복호속도가 기존의 방식에 비해 2배 빠르다. 시뮬레이션 결과 기존의 방식과 복호성능이 일치함을 알 수 있다. 복호시 처리 과정은 다음과 같이 세 단계로 나눌 수 있다.

(1) 단계 1

$i=0, 1$ 에 대해 순방향 및 역방향 상태 메트릭을 초기화시킨다.

$$\alpha_i^i(S_b^i(m)) = 1 \text{ for } m=0,$$

$$= 0 \text{ for } m \neq 0$$

$$\beta_i^i(S_b^i(m)) = 1 \text{ for } m=0,$$

$$= 0 \text{ for } m \neq 0$$

(2) 단계 2

N symbols을 모두 수신하여 저장한 후 모든 상태 m 에 대해 $N/2$ 까지의 순방향 메트릭 및 N 에서 $N/2$ 까지의 역방향 상태 메트릭을 동시에 구한다.

$$\hat{\alpha}_k^i(m) = \exp\left(\frac{2}{\sigma^2}(x_k i + y_k Y_k(i, m))\right) \left(\sum_{j=0}^1 \hat{\alpha}_{k-1}^i(S_b^i(m))\right)$$

$$(k=1, \dots, N/2) \tag{2-1}$$

$$\hat{\beta}_k^i(m) = \sum_{j=0}^1 \hat{\beta}_{k+1}^i(m) \exp\left(\frac{2}{\sigma^2}(x_{k+j} + y_{k+j} Y_{k+j}(j, S_b^i(m)))\right)$$

$$(k=N/2, \dots, N) \tag{2-2}$$

(3) 단계 3

순방향 및 역방향 상태 메트릭을 구한 후 중간지점부터 LLR을 구하여 양쪽으로 진전하면서 복호한다.

$$L(d_k)_{right} = \log \frac{\sum_m \alpha_k^i(m) \beta_k^i(m)}{\sum_m \alpha_k^o(m) \beta_k^o(m)} \quad (k=N/2, \dots, N) \tag{2-3}$$

$$L(d_k)_{left} = \log \frac{\sum_m \alpha_k^i(m) \beta_k^i(m)}{\sum_m \alpha_k^o(m) \beta_k^o(m)} \quad (k=(N/2)-1, \dots, 1) \tag{2-4}$$

3. 터보 복호기의 제작 및 시뮬레이션

터보 복호기의 성능을 검증하기 위하여 제작된 보드는 QPSK 변·복조기를 함께 구현하였으며, QPSK 복조기는 크게 비트 동기를 포착하는 STR (Symbol Timing Recovery) 루프(loop)와 반송파를 포착하는 CPR(Carrier Phase Recovery) 루프로 구성된다. 설계한 복조기는 CPLD 칩 내에 모델화하여 구현하였다. 별도의 테스트 보드를 제작하지 않고도 설계한 복조기의 기능을 검증할 수 있었고, 검증과정은 주로 타이밍 시뮬레이션을 이용하였다. 설계한 변조기는 EPF10K10 LC84-3 칩에 합성하였고, 복조기는 STR, CPR 각각의 경우에는 EPF10K30RC208-3 칩에 합성하였으며, 두 블록을 병합한 경우에는 EPF10K70RC240-3 칩에 합성하였다. 합성된 칩은 변조기의 경우에 약 50(Mbps), 복조기는 약 12(Mbps)의 속도로 동작이 가능하였다

성능검증에는 VHDL(Very high speed integrated circuit Hardware Description Language) 코드로 작성하고, Altera사의 Design compiler를 이용하여 컴파일하고 합성하였다.

터보복호기는 FLEX10K70RC240-3 칩으로, TB memory는 외부 RAM을 사용하지 않고 FLEX10 칩안의 내부 EAB(Embedded Array Block)을 사용하여 구현하였다. 구현된 보드는 모델기능을 동시에 갖고 있다. Interpolator는 샘플링율(Sampling Rate)를 증가시켜 Symbol timing error를 작게 하기 위하여 사용하였다.

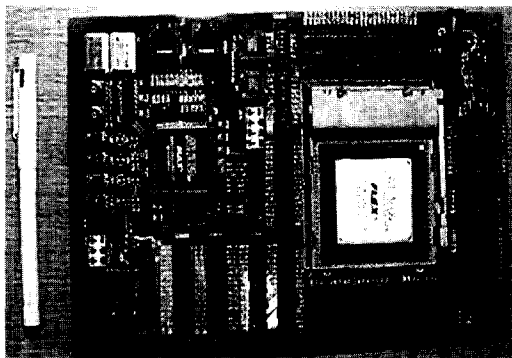


Fig. 7 Turbo Decoder Board using the FPGA Chip

VHDL의 시뮬레이션을 바탕으로 실제 칩에서 동작 여부를 확인하기 위해 EPF10K10LC84-3 칩을 사용하여 출력파형을 확인하였다. 구현 결과의 확인을 위하여 먼저, ALTERA사의 MAX PLUSII를 사용하여 로직을 구현하고, 다운로드시켰다. 그 후에 제작한 DAC(Digital to Analog Converter)를 사용하여 오실로스코프로 그 결과를 확인하였다.

Fig. 8은 구현된 보드의 비트동기를 포착하는 STR과 반송파 동기를 포착하는 CPR을 연동시켜서 동작시킨 시뮬레이션 결과를 나타낸다.

시뮬레이션에 이용한 입력신호는 임의의 타이밍 오류(timing error)를 부가한 정현파 신호이다. oput11과 oput22는 복조된 신호이다. 신호의 결과치는 2비트로 표현되므로 십진수 3은 이진수 "11"로서 -1을 나타내므로, I채널과 Q채널의 입력값이 동시에 1, 0, 1, 0이므로 정확히 복조됨을 알 수 있다.

기존의 터보복호 알고리즘을 적용할 경우에 터보 복호기 및 STR, CPR을 연동한 회로에서 128bit를 복호하는데 385.6(ns)가 소요된다. 그러나 본 연구에서 제안한 수정된 터보복호 알고리즘을 적용한 결과는 그림에서와 같이 시작점인 1.2(us)에서 복호 종료시점까지의 시간 간격, 즉 128bit를 복호하는데 소요되는 시간이 192.8(ns)이었다. 따라서 기존의 기존의 터보복호 알고리즘

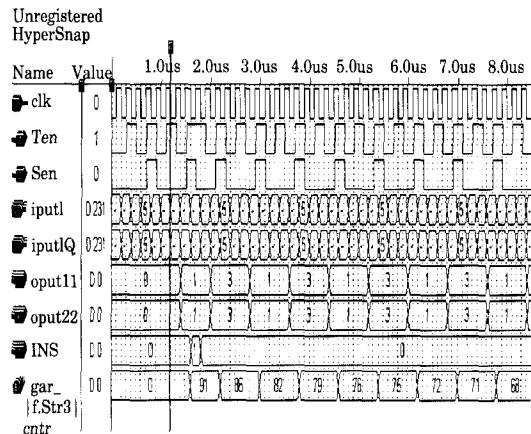


Fig. 8 Simulation Results of Turbo decoder Algorithm using FPGA Board

에 의한 복호속도에 비하여 2배 향상되었음을 알 수 있다.

4. 결 론

본 연구에서 제안한 터보코드를 이용한 통신 오류 정정 및 복호속도 개선기술을 선박의 화재탐지 장치에 적용한 결과 우수한 특성을 얻을 수 있었으며, 연구의 결과는 다음과 같다.

- (1) 통신 오류 정정을 위한 방안으로 터보코딩 기술이 적합함을 확인하였다.
- (2) 터보코딩 알고리즘의 단점인 속도개선방안을 제안하였다.
- (3) 이 기술을 이용하여 Altera사의 VHDL 소프트웨어 및 칩을 이용하여 약 12Mbps급의 만족스러운 속도를 얻을 수 있었다.

본 기술은 선박의 모든 통신방식에 적용할 수 있을 것으로 생각되며, 무선통신, 전력선통신 등에도 적용할 수 있을 것으로 기대된다.

후 기

이 연구는 산업자원부의 "산업기반기술개발"의 지원으로 수행한 연구결과의 일부이다. 산업자원부 및 한국산업기술평가원 관계자들에게 감사드립니다.

참고문헌

1. L. Bahl, J. Cocke, F. Jelinek and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate", *IEEE Trans. Inf. Theory*, vol. IT-20, pp. 284~287, Mar. 1974.
2. J. Hagenauer, E. Offer and L. Papke, "Iterative decoding of binary block and convolutional codes", *IEEE Trans. Inf. Theory*, vol. 42, no. 2, pp. 429~445, Mar. 1996.
3. C. Berrou, A. Glavieux and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding : Turbo-codes", *Proc. 1993 Int. Conf. Commun.*, pp. 1064~1070. 1993.
4. C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding : turbo-codes", *IEEE Trans. Commun.*, vol. 44, no. 10, pp. 1261~1271, Oct. 1996.
5. G. Battail, "A conceptual framework for understanding turbo codes", *IEEE J. Select. Areas Commun.*, vol. 16, no. 2, pp. 245~254, Feb. 1998.
6. P. Robertson, E. Villebrun and P. Hoeher, "A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain", *Proc. 1999 Int. Conf. Commun.*, vol. 2, pp. 1009~1013.
7. M. C. Valenti, "An introduction to Turbo codes", Unpublished Report, May 1996.
8. 김 용, 이필중, "IMT2000에 적합한 다중 테일터보 코드", *한국통신학회 추계종합학술발표회논문집*, 11/98. 624~627, 1999
9. 최영민, 이필중, "비대칭 변조 터보코드의 해석", *한국통신학회 부호 및 정보이론 워크숍 논문집*, 12/97 1998
10. 김 용, 이필중, "FPLMTS에서 음성전송을 위한 turbo code개발" *통신정보합동학술대회 논문집*, 제 1권, 4/97, pp.423~427 1998
11. 박창수, 김 용, 이필중, "전송율 1/2인 Turbo Code에서의 최적의 구성부호", *제3차 부호및 정보이론 워크숍 논문집*, 12/96 1999

저 자 소 개



정병홍(鄭秉弘)

1965년 2월생, 1988년 경북대학교 전자공학과 졸업, 현재 (주)이엔씨테크놀로지 대표이사.



최상학(崔相赫)

1962년 11월생, 1983년 해양대학교 기관공학과 졸업, 현재 (주)씨플러스 대표이사.



오종환(吳宗煥)

1958년 4월생, 1981년 한국해양대학교 기관학과 졸업, 1995년 2월 동대학원 전자통신학과 졸업(공학박사), 현재 한국해양수산연수원 해운교육부 교수.



김경석(金京錫)

1955년 12월생, 1978년 한국해양대학교 기관학과 졸업, 1989년 2월 한국해양대학 대학원 졸업(공학석사), 1992년 2월 동대학원 기관공학과 졸업(공학박사), 현재 한국해양수산연수원 해운교육부 교수.