(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2011-0057614
(43) 공개일자 2011년06월01일

(51) Int. Cl.
H03M 13/11 (2006.01)
(21) 출원번호 10-2009-0114085
(22) 출원일자 2009년11월24일
심사청구일자 2009년11월24일

(71) 출원인
한국전자통신연구원
대전 유성구 가정동 161번지
인하대학교 산학협력단
인천 남구 용현동 253 인하대학교
(72) 발명자
오정열
대전 유성구 지족동 898-1 가나파로스빌1차 419호
이형수
대전 유성구 노은동 열매마을 1104동 302호
(뒷면에 계속)
(74) 대리인
팬코리아특허법인

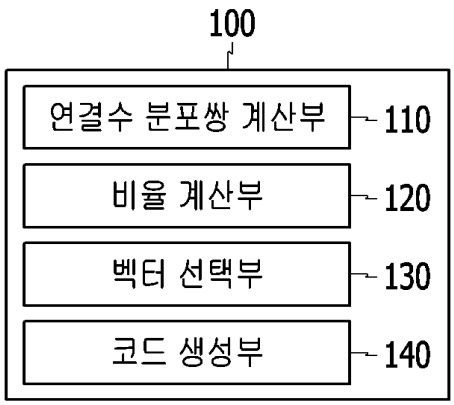
전체 청구항 수 : 총 10 항

(54) 오류 제어 코드 생성 시스템 및 그 방법

(57) 요 약

변수 노드와 검사 노드를 포함하는 패리티 검사 행렬을 토대로 오류 제어 코드를 생성하는 오류 제어 코드 생성 시스템은 변수 노드의 최대 연결수 이하의 정수를 원소로 포함하는 벡터들에 대해 각각 연결수 분포 쌍을 계산하고, 계산한 연결수 분포 쌍에 대해, 임의의 연결수에서의 오차 확률과 코드어의 평균 오차 확률간에 비율을 계산한다. 다음, 오류 제어 코드 생성 시스템은 계산한 비율을 토대로 연결수 분포 쌍에 대응하는 벡터를 선택하여, 선택한 벡터를 토대로 오류 제어 코드를 생성한다.

대 표 도 - 도1



(72) 발명자

김재환

대전 유성구 전민동 세종아파트 101동 407호

김재영

대전 유성구 노은동 열매마을8단지 812동 1302호

곽경섭

인천 연수구 송도동 3-40, 풍림 602동 102호

곽대한

경기 수원시 팔달구 인계동 159번지 선경3차 306동 501호

특허청구의 범위

청구항 1

변수 노드와 검사 노드를 포함하는 패리티 검사 행렬을 토대로 오류 제어 코드를 생성하는 시스템에 있어서,

상기 변수 노드의 최대 연결수 이하의 정수를 원소로 포함하는 벡터들에 대해 각각 연결수 분포 쌍을 계산하는 연결수 분포 쌍 계산부,

상기 연결수 분포 쌍들에 대해, 임의의 연결수에서의 오차 확률과 코드어의 평균 오차 확률 간에 비율을 각각 계산하는 비율 계산부,

상기 비율 중 가장 큰 비율을 가지는 연결수 분포 쌍에 대응하는 벡터를 파라미터로서 선택하는 벡터 선택부, 그리고

파라미터로 선택한 벡터를 토대로 오류 제어 코드를 생성하는 코드 생성부

를 포함하는 코드 생성 시스템.

청구항 2

제1항에 있어서,

상기 연결수 분포 쌍 계산부는

상기 변수 노드의 연결수 분포 및 상기 검사 노드의 연결수 분포를 포함하는 상기 연결수 분포 쌍을 계산하는 오류 제어 코드 생성 시스템.

청구항 3

제2항에 있어서,

상기 변수 노드의 연결수 분포는 제1 연결수를 갖는 변수 노드로부터 나오는 엣지의 프랙션 집합이며,

상기 검사 노드의 연결수 분포는 제2 연결수를 갖는 검사 노드로부터 나오는 엣지의 프랙션 집합인 오류 제어 코드 생성 시스템.

청구항 4

제1항에 있어서,

상기 비율 계산부는

평균 및 분산을 포함하는 가우스 변수 및 연결수를 포함하는, 상기 연결수 분포 쌍에 기초하여 상기 비율을 계산하는 오류 제어 코드 생성 시스템.

청구항 5

제1항에 있어서,

상기 코드 생성부는

상기 파라미터로 선택한 벡터와 코드율, 오차 방지 레벨에 따른 비트들의 프렉션을 포함하는 생성함수를 통해 상기 오류 제어 코드를 생성하는 오류 제어 코드 생성 시스템.

청구항 6

변수 노드와 검사 노드를 포함하는 패리티 검사 행렬을 토대로 오류 제어 코드를 생성하는 방법에 있어서,

상기 변수 노드의 최대 연결수 이하의 정수를 원소로 포함하는 벡터들에 대해 각각 연결수 분포 쌍을 계산하는 단계,

상기 연결수 분포 쌍에 대해, 임의의 연결수에서의 오차 확률과 코드어의 평균 오차 확률간에 비율을 계산하는 단계, 그리고

계산한 비율을 토대로 연결수 분포 쌍에 대응하는 벡터를 선택하여, 상기 벡터를 토대로 오류 제어 코드를 생성하는 단계

를 포함하는 오류 제어 코드 생성 방법.

청구항 7

제6항에 있어서,

상기 연결수 분포 쌍은 상기 변수 노드의 연결수 분포 및 상기 검사 노드의 연결수 분포로 구성되며,

상기 변수 노드의 연결수 분포는 제1 연결수를 갖는 변수 노드로부터 나오는 엣지의 프랙션 집합이며, 상기 검사 노드의 연결수 분포는 제2 연결수를 갖는 검사 노드로부터 나오는 엣지의 프랙션 집합인 오류 제어 코드 생성 방법.

청구항 8

제6항에 있어서,

상기 오류 제어 코드를 생성하는 단계는

상기 비율 중 가장 큰 비율을 가지는 연결수 분포 쌍에 대응하는 벡터를 파라미터로서 선택하는 단계, 그리고

파라미터로 선택한 벡터를 토대로 상기 오류 제어 코드를 생성하는 단계

를 포함하는 오류 제어 코드 생성 방법.

청구항 9

제6항에 있어서,

상기 비율을 계산하는 단계는

상기 연결수 분포 쌍이 포함하는 연결수, 평균 및 분산을 포함하는 가우스 변수에 기초하여 상기 비율을 계산하는 단계를 포함하는 오류 제어 코드 생성 방법.

청구항 10

제6항에 있어서,

상기 비율을 계산하는 단계는

상기 연결수 분포 쌍에 대해 가우스 근사법을 적용하여 상기 비율을 계산하는 단계를 포함하는 오류 제어 코드 생성 방법.

명 세 서

발명의 상세한 설명

기 술 분 야

[0001] 본 발명은 오류 제어 코드 생성 시스템 및 그 방법에 관한 것이다. 특히 본 발명은 가우스 근사법에 기반한 오류 제어 코드 생성 시스템 및 그 방법에 관한 것이다.

배 경 기 술

[0002] 통신 시스템의 정보 전송 과정에서 수행하는 채널 부호화(channel coding) 방법은 균등 오류 보호(Equal Error Protection, 이하 "EEP"라고 함) 방식과 차등 오류 보호(Unequal Error Protection, 이하 "UEP"라고 함) 방식을 포함한다.

[0003] EEP 방식은 모든 정보 비트를 동일한 부호를 토대로 부호화하는 채널 부호화 방식이다.

[0004] UEP 방식은 정보 비트들을 중요성에 대응하게 일정 개수의 스트림(stream)으로 구분하고, 구분된 상기 스트림의 중요도에 따라서 중요도가 낮은 데이터 스트림은 부호정정능력이 낮은 부호를 토대로 부호화하고, 중요도가 높은 데이터 스트림은 부호정정능력이 높은 부호를 토대로 부호화한다. 즉, UEP 방식은 중요도를 고려하지 않는 EEP 방식에 비해서 정보 비트들의 특성을 고려하므로 사용상의 이득을 갖는다.

[0005] 또한, 통신 시스템의 정보 전송 과정에서는 채널의 상황에 따라 잡음(noise)과, 간섭(interference) 및 페이딩(fading) 등으로 인해 불가피하게 에러(error)가 발생하고, 에러 발생으로 인한 정보의 손실이 발생한다. 이러한 에러 발생으로 인한 정보 손실을 방지하기 위하여, 통신 시스템은 채널의 특성에 대응하는 에러 제어 방식을 적용한다. 여기서, 에러 제어 방식은 터보(turbo) 방식, 저밀도 패리티 검사(Low-Density Parity-Check, 이하 "LDPC"라고 함) 방식 등을 포함한다.

[0006] LDPC 방식은 터보 부호에 비해서 낮은 복호 복잡도 및 병렬 처리가 가능한 장점을 가지고 있으나, 상대적으로 높은 부호화 복잡도와 부호기에서 생성행렬을 저장하기 위한 많은 메모리를 요구하는 문제점이 있다.

[0007] 따라서, 정보 전송 과정에서는 UEP 방식과 LDPC 방식의 문제점을 보안하는 오류 제어 코드를 생성하여, 생성한 오류 제어 코드를 전송해야한다.

발명의 내용

해결 하고자하는 과제

[0008] 본 발명이 이루고자 하는 기술적 과제는 UEP 방식과 LDPC 방식을 이용하는 오류 제어 코드를 생성하는 시스템 및 방법을 제공하는 것이다.

과제 해결수단

[0009] 상기한 목적을 달성하기 위한 본 발명의 특징에 따른, 변수 노드와 검사 노드를 포함하는 패리티 검사 행렬을 토대로 오류 제어 코드를 생성하는 시스템은

[0010] 상기 변수 노드의 최대 연결수 이하의 정수를 원소로 포함하는 벡터들에 대해 각각 연결수 분포 쌍을 계산하는 연결수 분포 쌍 계산부, 상기 연결수 분포 쌍들에 대해, 임의의 연결수에서의 오차 확률과 코드어의 평균 오차 확률 간에 비율을 각각 계산하는 비율 계산부, 상기 비율 중 가장 큰 비율을 가지는 연결수 분포 쌍에 대응하는 벡터를 파라미터로서 선택하는 벡터 선택부, 그리고 파라미터로 선택한 벡터를 토대로 오류 제어 코드를 생성하는 코드 생성부를 포함한다.

[0011] 본 발명의 다른 특징에 따른, 변수 노드와 검사 노드를 포함하는 패리티 검사 행렬을 토대로 오류 제어 코드를 생성하는 방법은

[0012] 상기 변수 노드의 최대 연결수 이하의 정수를 원소로 포함하는 벡터들에 대해 각각 연결수 분포 쌍을 계산하는 단계, 상기 연결수 분포 쌍에 대해, 임의의 연결수에서의 오차 확률과 코드어의 평균 오차 확률간에 비율을 계산하는 단계, 그리고 계산한 비율을 토대로 연결수 분포 쌍에 대응하는 벡터를 선택하여, 상기 벡터를 토대로 오류 제어 코드를 생성하는 단계를 포함한다.

효 과

[0013] 본 발명의 실시예에 따르면, UEP 방식과 LDPC 방식을 이용하여 보다 효율적으로 오류 제어 코드를 생성할 수 있다. 또한, 본 발명의 실시예에 따르면, 생성된 오류 제어 코드를 이용하여 정보를 전송함에 따라, 재구성되는 정보의 품질을 향상 시킬 수 있다.

발명의 실시를 위한 구체적인 내용

[0014] 아래에서는 첨부한 도면을 참고로 하여 본 발명의 실시예에 대하여 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자가 용이하게 실시할 수 있도록 상세히 설명한다. 그러나 본 발명은 여러 가지 상이한 형태로 구현될 수 있으며 여기에서 설명하는 실시예에 한정되지 않는다. 그리고 도면에서 본 발명을 명확하게 설명하기 위해서 설명과 관계없는 부분은 생략하였으며, 명세서 전체를 통하여 유사한 부분에 대해서는 유사한 도면 부호를 붙였다.

[0015] 명세서 전체에서, 어떤 부분이 어떤 구성요소를 "포함"한다고 할 때, 이는 특별히 반대되는 기재가 없는 한 다

른 구성요소를 제외하는 것이 아니라 다른 구성요소를 더 포함할 수 있는 것을 의미한다.

[0016] 이하에서는, 본 발명의 실시예에 따른 오류 제어 코드 생성 시스템 및 그 방법에 대하여 첨부한 도면을 참고로 하여 상세히 설명한다.

[0017] 먼저, 오류 제어 코드 생성 시스템은 정보 제공자로부터 제공되는 다양한 정보들을 하나의 채널로 전송하기 위해서는 각 정보를 재구성하여, 앙상블(Ensemble) 형태로 전송한다.

[0018] 본 발명의 실시예에 따른 오류 제어 코드 생성 시스템은 일반적인 잡음이 있는 채널 예를 들어, AWGN(Additive White Gaussian Noise) 채널 상에서, 가우스 근사법(Gaussian approximation)을 기반으로하여 코드를 생성한다. 여기서, 생성한 오류 제어 코드는 UEP 방식과 저밀도 패리티 검사(Low-Density Parity-Check, 이하 "LDPC"라고 함) 방식이 적용된 UEP- LDPC 코드이다.

[0019] 본 발명의 실시예에 따라 생성한 오류 제어 코드는 채널로 전송되는 경우, 앙상블 형태 즉, 코드 앙상블 형태로 전송된다. 또한, 코드 앙상블 형태로 전송됨에 따라 재구성되는 정보의 품질을 향상 시킬 수 있다.

[0020] 다음, UEP-LDPC 코드 앙상블을 아래와 같이 정의할 수 있다.

[0021] **1. UEP-LDPC 코드 앙상블**

[0022] 먼저, LDPC 코드의 패리티 검사 행렬(H)은 희소 행렬(Sparse Matrix)이며, 이분 그래프(bipartite graph)에 상응하게 나타낸다. 여기서, 패리티 검사 행렬은 수신된 신호에 대한 정상적인 복호 여부를 확인하기 위한 행렬이며, 부호화된 수신 신호와 패러티 검사 행렬의 곱이 '0'이 되었을 경우 에러가 발생하지 않은 것으로 판단된다.

[0023] 패리티 검사 행렬(H)에서 예를 들어, "0(zero)"이 아닌 엔트리(entry)가 'j'번째 행과 'i'번째 열에 존재하는 경우, 'i'번째 비트(변수) 노드(이하, "변수 노드"라고 함)와 'j'번째 검사 노드 사이에는 엣지(edge)가 존재한다. 각 노드에 연결된 엣지의 수는 노드의 연결수(degree)와 동일하다. 노드의 연결수는 패리티 검사 행렬(H)에서 열 또는 행의 엔트리 중 '0'이 아닌 엔트리의 수에 대응한다.

[0024] LDPC 코드 앙상블은, 수학식 1 및 수학식 2에 따른 연결수 분포($\lambda(x)$ 및 $\rho(x)$)에 해당하는 생성함수(generating function)($\lambda(x)$, $\rho(x)$)에 의해 결정된다.

수학식 1

[0025]
$$\lambda(x) = \sum_{i=2}^{d_{v\max}} \lambda_i \cdot x^{i-1}$$

수학식 2

[0026]
$$\rho(x) = \sum_{j=2}^{d_{c\max}} \rho_j \cdot x^{j-1}$$

[0027] 여기서, $d_{v\max}$는 변수 노드의 최대 연결수이며, $d_{c\max}$는 검사 노드의 최대 연결수이다. 또한, λ_i 및 ρ_i는 각각 변수 노드의 연결수(i) 및 검사 노드의 연결수(j) 로부터 나오는 엣지의 프랙션(fraction)이다.

[0028] 즉, LDPC 코드 앙상블은 연결수 분포 쌍(λ, ρ)으로 나타낸다. 여기서, λ는 집합

$\{\lambda_i, 2 \le i \le d_v\}$를 ρ는 집합 $\{\rho_i, 2 \le i \le d_c\}$으로 나타낸다.

[0029] LDPC 코드어(codeword)의 비트들 각각은 서로 다른 오차 방지 레벨에 해당하는 블록들(예를 들어, 's'개의 블록)로 구성된다. 여기서, 오차 방지 레벨(k)의 비트들의 프랙션이 a_k인 경우, 프랙션 시퀀스 (a)는 수학식 3 및 수학식 4와 같이 나타낸다.

수학식 3

[0030] $$a = (a_1, a_2, \cdots, a_s)$$

수학식 4

[0031] $$\sum_{k=1}^{s} \alpha_k = 1$$

[0032] 여기서, 'k'는 '1' 이상, 's'이하의 값을 가지며, 's'는 오차 방지 레벨에 해당하는 블록의 수이다.

[0033] 이하 본 발명의 실시예에 따른, 벡터에 대해 각각 연결수 분포 쌍(λ_{UEP}, ρ_{UEP})을 계산하는 방법을 설명한다.

[0034] 계산의 단순화를 위해서는 오차 방지 레벨(k)의 비트들에 동일한 연결수를 할당한다. LDPC 코드어의 패리티 검사 비트들이 가장 작은 연결수를 가지는 변수 노드와 연결되는 경우, 기준 오차 방지 레벨보다 높은 오차 방지 레벨은 기준 오차 방지 레벨에 해당하는 연결수 보다 큰 연결수가 할당된다.

[0035] 즉, 본 발명의 실시예에 따른, 노드 연결수의 시퀀스(d) 즉, 벡터는 수학식 5와 같이 나타낸다.

수학식 5

[0036] $$d = (d1, d2, \cdots, ds)$$

[0037] 여기서, '1' 이상 's' 이하의 값을 갖는 'k'에 대해 'd_k'는 감소하는 순서(예를 들어, $d1 > d2 > \cdots > ds$)에서 오차 방지 레벨(k)의 비트들의 연결수일 수 있다.

[0038] 이때, 연결수(d_k)의 변수 노드로부터 나오는 엣지의 프랙션(λ_k)은 수학식 6과 같이 나타낸다.

수학식 6

[0039] $$\lambda_k = \frac{\alpha_k d_k}{\sum_{i=1}^{s} \alpha_i d_i} \qquad (1 \le k \le s)$$

[0040] 또한, 변수 노드의 연결수 분포(λ_{UEP})는 수학식 7과 같이 나타내고, 변수 노드의 생성함수($\lambda(x)$)는 수학식 8과 같이 나타낸다.

수학식 7

$$\lambda_{UEP} = \left\{ \lambda_{d_s}, \lambda_{d_{s-1}}, \ldots, \lambda_{d_2}, \lambda_{d_1} \right\}$$

[0041]

수학식 8

$$\lambda(x) = \sum_{k=1}^{s} \lambda_k x^{d_k - 1}$$

[0042]

[0043] 모든 검사 노드에 해당하는 엣지의 수가 d_c 이고, 검사 노드들의 생성함수가 수학식 9와 같은 경우, 검사 노드의 연결수 분포(ρ_{UEP})는 수학식 10과 같이 나타낸다.

수학식 9

$$\rho(x) = x^{d_c - 1}$$

[0044]

수학식 10

$$\rho_{UEP} = \left\{ \rho_{d_c} = 1 \right\}$$

[0045]

[0046] 코드율(code rate)이 R인 경우, 모든 검사 노드에 해당하는 엣지의 수 (d_c)는 수학식 11과 같이 나타낸다.

수학식 11

$$d_c = \frac{1}{1-R} \sum_{k=1}^{s} \alpha_k d_k$$

[0047]

[0048] 이와 같이, 본 발명의 실시예에 따른 UEP-LDPC 코드 앙상블은 변수 노드의 연결수 분포(λ_{UEP})와 검사 노드의 연결수 분포(ρ_{UEP})를 포함하는 연결수 분포 쌍$(\lambda_{UEP}, \rho_{UEP})$으로 나타낸다. 또한, 연결수 분포 쌍 $(\lambda_{UEP}, \rho_{UEP})$은 파라미터(*R, a, d*)로 나타낼 수 있다. 즉, 연결수 분포 쌍은 코드를 분석하기 편리하고, 파라미터는 실제로 UEP-LDPC 코드 앙상블의 표현을 더 간결하게 할 수 있다.

[0049] 다음, 불규칙하게 구성된 패리티 검사 행렬(H)에 UEP 방식을 적용하는 방법을 아래와 같이 설명한다.

[0050] **2. 불규칙한 LDPC 코드를 위한 UEP의 구현**

[0051] 먼저, 정보 시퀀스(u)의 길이는 'K'이고, LDPC 코드어의 길이는 'N' 라고 가정한다.

[0052] 시퀀스가 수학식 3과 같은 경우, 정보 시퀀스(m)는 수학식 12와 같이 나타낸다.

수학식 12

[0053]
$$\boldsymbol{m} = (m_{11}, m_{12}, \ldots, m_{1,\alpha_1 N}, m_{21}, m_{22}, \ldots, m_{2,\alpha_2 N}, \ldots, m_{s-1,1}, m_{s-1,2}, \ldots, m_{s-1,\alpha_{s-1} N})$$

[0054] 여기서, '$\alpha_1 N$, $\alpha_2 N, \cdots, \alpha_s N$'는 각각 오차 방지 레벨($k$)을 포함하는 정보 비트들의 길이이다. 여기서, 오차 방지 레벨(k)은 '$1, 2, \cdots, s-1$'를 포함한다. '$\alpha_1 N, \alpha_2 N, \cdots, \alpha_s N$'의 합은 오차 방지 레벨($k$)과 동일하다.

[0055] 즉, 서로 다른 오차 방지 레벨의 비트들은 LDPC 코드의 체계적인 부호화로 인하여, 서로 다른 연결수의 비트 노드들과 매핑될 수 있다.

[0056] 패리티 검사 행렬(H)은 수학식 13과 같이 두개의 하위행렬(A 및 B)로 나눠질 수 있다.

수학식 13

[0057] $H = [A \mid B]$

[0058] 여기서 행렬 A 및 행렬 B의 차원은 '$M \times K$' 및 '$M \times M$'일 수 있다. 또한, '$M = N - K$'는 검사 노드의 수 또는 패리티 검사 식(equation)의 수를 나타낸다.

[0059] 예를 들어, 행렬 B가 최대-랭크(full-rank)이고, 행렬 B의 역행렬이 'B-1'라 가정하면, 패리티 검사 행렬(H)은 수학식 14와 같이 나타낸다.

수학식 14

[0060] $H = [B^{-1}A \mid I_M]$

[0061] 여기서, I_M는 '$M \times K$' 차원의 단위행렬이다.

[0062] 이때, 패리티 검사 행렬(H)과 생성 행렬(G)간의 관계를 수학식 15와 같이 나타낸다.

수학식 15

[0063]
$$G = [I_{(N-M)} \mid (B^{-1}A)^T]$$

[0064] 이때, 정보 시퀀스가 'm'인 경우, 코드어(c)는 상기 수학식 1을 토대로 수학식 16 과 같이 나타낸다.

수학식 16

[0065] $c = m \times G = [m \mid m(B^{-1}A)^T] = [m \mid p]$

[0066] 여기서, 'p'는 패리티 검사 시퀀스이며, 수학식 17과 같이 나타낸다.

수학식 17

$$p = (p_1, p_2, \ldots, p_{\alpha_s N})$$

[0067]

[0068] 다음, 패리티 검사 식은 수학식 18과 같이 나타낸다.

수학식 18

$$H \times c^T = [A \mid B] \times [m \mid p]^T = 0$$

[0069]

[0070] 여기서, 수학식 15 및 수학식 18을 토대로 수학식 19를 얻을 수 있다.

수학식 19

$$Am^T + Bp^T = 0$$

[0071]

[0072] 예를 들어, 행렬 A 및 행렬 B가 수학식 20 및 수학식 21의 형태를 갖는 경우, 패리티 검사 식은 수학식 12 및 수학식 19를 토대로 수학식 22와 같이 나타낸다.

수학식 20

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1K} \\ a_{21} & a_{22} & \ldots & a_{2K} \\ & \ldots & & \\ a_{M1} & a_{M2} & \ldots & a_{MK} \end{bmatrix}$$

[0073]

수학식 21

$$\mathbf{B} = \begin{bmatrix} b_{11} & b_{12} & \ldots & b_{1M} \\ b_{21} & b_{22} & \ldots & b_{2M} \\ & \ldots & & \\ b_{M1} & b_{M2} & \ldots & b_{MM} \end{bmatrix}$$

[0074]

수학식 22

$$\begin{cases} a_{11}m_{11} + \ \ldots\ + a_{1K}m_{s-1,\alpha_{s-1}N} + b_{11}p_1 + \ \ldots\ + b_{1M}p_M = 0 \\ a_{21}m_{11} + \ldots\ + a_{2K}m_{s-1,\alpha_{s-1}N} + b_{21}p_1 + \ \ldots\ + b_{2M}p_M = 0 \\ \qquad\qquad \ldots \\ a_{M1}m_{11} + \ldots + a_{MK}m_{s-1,\alpha_{s-1}N} + b_{M1}p_1 + \ldots + b_{MM}p_M = 0 \end{cases}$$

[0075]

[0076] 상기 수학식 22 즉, 패리티 검사식은 정보 비트에 관계된 식의 수가 패리티 검사 행렬(H)의 대응되는 행에 의해 결정되는 것을 나타낸다. 예를 들어, 수학식 22의 첫번째 행 $(a_{11},\ a_{12},\cdots,a_{M1})$에서 '0'이 아닌 값이 있는 경우, m_{11} 은 패리티 검사식에서 필요로 한다. 만약, 행렬 A의 행이 '1'의 수가 감소하는 순서인 경우, 가장 중요한 비트는 패리티 검사식의 가장 큰 수에 관여할 것이다.

[0077] 패리티 검사 행렬(H)의 태너 그래프(Tanner graph)에 기초한 반복적인 부호화 동안, 이러한 비트들은 이웃한 검사 노드로부터 가장 외부의 정보(most extrinsic information)를 받을 수 있으며, 더 쉽게 부호화될 수 있다.

[0078] 랜덤하게 구성된 패리티 검사 행렬(H)에서 조건으로 지정된 연결수의 노드 위치를 아는 것은 어렵다. 반면에, 패리티 검사 행렬(H)로 구성된 PEG(Progressive edge-growth) 알고리즘을 위해 노드들은 불규칙한 LDPC 코드를 위한 UEP를 더 쉽게 실현하기 위한 노드들의 연결수에 의해 그룹 지어질 수 있다.

[0079] 다음, 본 발명의 실시예에 따른 오류 제어 코드 생성 방법이 기반으로 하는 가우스 근사법을 설명한다.

[0080] **3. LDPC 코드의 가우스 근사법**

[0081] 먼저, 가우스 근사법은 부호화기의 로그 우도비(log likelihood ratio, 이하 "LLR"라고 함) 메시지의 가우스 가정(Gaussian assumption)에 기초한 LDPC 코드의 1-D 분석 방법이다. 이러한 방법은 밀도 진화(density evolution)보다 부정확하더라도, 낮은 계산 복잡성으로 인해 AWGN 채널상에서 LDPC 코드 앙상블을 분석하고 평가하기에는 효과적일 수 있다.

[0082] LDPC 썸-프로덕트(sum-product) 부호화의 밀도 진화 분석 및 무한-길이 가정(infinite-length assumption)에 기초하여, 하나의 노드에 이웃 노드로부터 들어오는 모든 LLR 메시지는 독립적인 그리고 같은 분포를 따르는 가우스 또는 가우스 혼합 변수에 근접해질 수 있다. 동시에 가우스 변수들은 대칭적이고, 평균 'm'과 분산 '$\sigma^2=2m$'을 가질 수 있다. 그러므로, LLR 메시지의 초기 확률 밀도 함수(Probability Density Function, PDF)가 주어지는 경우에는 LLR 메시지 평균의 결과(evolution)를 추적함으로써, 부호화 메시지의 확률 밀도 함수 및 임의의 반복에서의 오차 확률이 계산될 수 있다.

[0083] 썸-프로덕트 부호화의 'l' 번째 반복에서, 검사 노드에서 비트 노드로 보내지는 검사 비트(check-to-bit) 메시지 및 비트 노드에서 검사 노드로 보내지는 비트 검사(bit-to-check) 메시지는 각각 '$u^{(l)}$' 및 '$v^{(l)}$'와 같이 표현될 수 있다.

[0084] '$u_j^{(l)}$'는 연결수 'j'의 검사 노드에 의해 보내지는 메시지일 수 있고, '$v_i^{(l)}$'는 연결수 'i'의 비트 노드에 의해 보내지는 메시지일 수 있다. 이때, 'l'은 반복 수일 수 있으며, 'u_0'는 초기 채널 메시지일 수 있다.

[0085] 비트 노드가 처리되는 동안, 연결수 'i'의 비트 노드는 'i-1' 검사 노드들로부터 메시지를 수집할 수 있으며, 이와 같은 방법으로 초기 채널 메시지를 수집하여 수학식 23을 얻을 수 있다.

수학식 23

$$m_{v,i}^{(l)} = m_0 + (i-1)m_u^{(l-1)}$$

[0086]

[0087] 여기서, m_0, $m_u^{(1)}$ 및 $m_{v,i}^{(1)}$는 각각 u_0, $u^{(1)}$ 및 $v_i^{(1)}$의 평균값이며, $m_u^{(0)}$ 은 '0'일 수 있다. AWGN 채널 상에서 모두 '0'의 값을 갖는 코드어가 전송되고, '-1'에 매핑된다고 가정하면, m_0는 $2/\sigma^2$와 동일하다. 여기서, σ는 노이즈의 표준편차이다.

[0088] 체크 노드가 처리되는 동안, 연결수 'j'를 갖는 검사 노드는 탄젠트 규칙(tanh rule)에 의해 비트 노드들로부터 메시지를 수집할 수 있고, 기대값을 획득하여, 수학식 24를 얻을 수 있다.

수학식 24

$$E\left[\tanh\frac{u_j^{(l)}}{2}\right] = E\left[\tanh\frac{v^{(l)}}{2}\right]^{j-1}$$

[0089]

[0090] 예를 들어, 'u'가 평균 'm_u' 및 분산 '$2m_u$'를 갖는 가우스 변수인 경우, 탄젠트 '$u/2$'의 기대값은 하기 수학식 25와 같이 나타낸다.

수학식 25

$$E\left[\tanh\frac{u}{2}\right] = \frac{1}{\sqrt{4\pi m_u}}\int\left(\tanh\frac{u}{2}\right)e^{-\frac{(u-m_u)^2}{4m_u}}du$$

[0091]

[0092] 여기서, 함수 '$\phi(x)$'는 수학식 26과 같이 나타낸다.

수학식 26

$$\phi(x) = \begin{cases} 1-\dfrac{1}{\sqrt{4\pi x}}\displaystyle\int\tanh\frac{u}{2}e^{-\frac{(u-x)^2}{4x}}du, & \text{if } x>0 \\ 1, & \text{if } x=0 \end{cases}$$

[0093]

[0094] 이러한 수적인 계산에서, 함수 '$\phi(x)$'는 수학식 27과 같이 근사값을 가진다.

수학식 27

$$\phi(x) \approx \begin{cases} e^{-0.4527x^{0.86}+0.0218}, & \text{for } 0 \le x < 10 \\ \sqrt{\frac{\pi}{x}} e^{-\frac{x}{4}} \left(1-\frac{10}{7x}\right), & \text{for } x \ge 10 \end{cases}$$

[0095]

[0096] 또한, 상기 수학식 24 내지 상기 수학식 26으로부터, 수학식 28을 얻을 수 있으며, 수학식 28의 역을 수학식 29와 같이 나타낸다.

수학식 28

$$E\left[\tanh\frac{u_j^{(l)}}{2}\right] = E\left[\tanh\frac{v^{(l)}}{2}\right]^{j-1} = 1-\phi(m_{u,j}^{(l)})$$

[0097]

수학식 29

$$m_{u,j}^{(l)} = \phi^{-1}\left(1-E\left[\tanh\frac{v^{(l)}}{2}\right]^{j-1}\right)$$

[0098]

[0099] 한편, 비트 검사 메시지 $'v^{(l)}'$ 는 가우스 혼합 변수일 수 있으며, 가우스 혼합 변수의 확률 밀도 함수는 수학식 30과 같이 나타낸다.

수학식 30

$$f_v^{(l)}(t) = \sum_{i=1}^{d_{v\max}} \lambda_i \cdot \frac{1}{\sqrt{4\pi m_{v,i}^{(l)}}} e^{-\frac{(t-m_{v,i}^{(l)})^2}{4m_{v,i}^{(l)}}}$$

[0100]

[0101] 여기서, 탄젠트 $' v^{(l)}/2'$ 의 평균은 상기 수학식 26 및 수학식 30을 통해 수학식 31과 같이 나타낸다.

수학식 31

$$E\left[\tanh\frac{v^{(l)}}{2}\right] = \int_{-\infty}^{\infty} \tanh(t/2)\cdot f_v^{(l)}(t)dt$$

$$= 1-\sum_{i=2}^{d_v} \lambda_i \phi\left(m_{v,i}^{(l)}\right)$$

[0102]

[0103] 또한, 검사 비트 메시지$(u_j^{(l)})$의 평균$(m_{u,j}^{(l)})$은 상기 수학식 29 및 수학식 31을 통해 수학식 32와 같이 나타낸다.

수학식 32

$$m_{u,j}^{(l)} = \phi^{-1}\left(1-\left(1-\sum_{i=2}^{d_{v\max}} \lambda_i \phi\left(m_{v,i}^{(l)}\right)\right)^{j-1}\right)$$

[0104]

[0105] 여기서, 검사 비트 메시지$(u_j^{(l)})$의 평균$(m_{u,j}^{(l)})$은 $\{\rho_j,\ 2 \le j \le d_c\}$만큼의 가중치를 갖는 연결수$(2,\ \cdots,\ d_c)$의 검사 노드 메시지의 평균과의 선형 결합을 토대로 수학식 33과 같이 나타낸다.

수학식 33

$$m_u^{(l)} = \sum_{j=2}^{d_{c\max}} \rho_j \cdot m_{u,j}^{(l)}$$

[0106]

[0107] 이때, 검사 비트 메시지$(u^{(l)})$의 수정된 평균($m_u^{(l)}$)은 상기 수학식 32, 수학식 33 및 수학식 23을 토대로 수학식 34와 같이 나타낸다.

수학식 34

$$m_u^{(l)} = \sum_{j=2}^{d_{c\max}} \rho_j \phi^{-1}\left(1-\left(1-\sum_{i=2}^{d_{v\max}} \lambda_i \phi\left(m_0+(i-1)m_u^{(l-1)}\right)\right)^{j-1}\right)$$

[0108]

[0109] 이러한 반복을 이용하여, 연결수 'i'의 비트 노드는 이용 가능한 정보를 수집할 수 있다. 또한, 메시지의 오차 확률$(P_{e,i}^{(l)})$은 수학식 35와 같이 나타낸다.

수학식 35

$$P_{e,i}^{(l)} = Q\left(\sqrt{\frac{m_0 + i \cdot m_u^{(l)}}{2}}\right)$$

[0110]

[0111] 여기서, Q()는 수학식 36과 같은 Q 함수이다.

수학식 36

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_{x}^{\infty} e^{-t^2/2} dt$$

[0112]

[0113] 이때, 연결수 'i'가 증가할수록, 메시지의 오차 확률은 감소한다. 이는 불규칙한 LDPC 코드가 고유의 UEP 특성들을 처리하기 때문이다. 높은 연결수의 비트 노드의 오차 확률은 낮은 연결수의 비트 노드의 오차 확률보다 더 낮다.

[0114] 따라서, 서로 다른 연결수의 노드들의 평균을 구하는 경우, i번째 반복에서 평균 오차 확률 $(P_e^{(l)})$은 수학식 37과 같이 나타낸다.

수학식 37

$$P_e^{(l)} = \sum_{i=2}^{d_{v\max}} \lambda_i' Q\left(\sqrt{\frac{m_0 + i \cdot m_u^{(l)}}{2}}\right)$$

[0115]

[0116] 여기서, λ_i' 은 연결수 'i' 비트 노드의 프랙션이며, 수학식 38과 같이 나타낸다.

수학식 38

$$\lambda_i' = \frac{\lambda_i / i}{\sum_{k=2}^{d_{v\max}} \lambda_k / k}$$

[0117]

[0118] 수학식 37을 참고하면, 검사 비트 메시지$(u^{(1)})$의 수정된 평균$(m_u^{(l)})$이 '$l \to \infty$'로서 무한대에 수렴하는 경우, 평균 오차 확률이 '0'에 수렴함을 알 수 있다. 이때, LDPC 부호화의 반복 절차는 연결수 분포 쌍 (λ, ρ)과 초기 채널 메시지(u_0)의 평균값(m_0)의 값에 의해 결정된다.

[0119] LDPC 코드 앙상블 즉, 연결수 분포 쌍(λ, ρ)의 스레스홀드(threshold)는 모든 'm_0'의 하한일 수 있으며, 이로 인하여 평균 오차 확률이 '0'에 수렴하거나 '$m_u^{(1)}$'이 '$l \to \infty$'로서 무한대에 수렴한다.

[0120] 이처럼, 소정의 반복 절차를 통해 서로 다른 연결수의 비트들에 대한 점근적 오차 확률을 분석할 수 있으며, 이

를 이용하여 좋은 UEP-LDPC 코드를 발견할 수 있다.

[0121] 다음, 본 발명의 실시예에 따른 계산한 연결수 분포 쌍 $(\lambda_{UEP}, \rho_{UEP})$에 대해 가우스 근사법을 이용하여 성능을 평가하는 방법을 설명한다.

[0122] **4. UEP-LDPC 코드의 성능 평가**

[0123] 본 발명의 실시예에 따른, UEP-LDPC 코드 앙상블(R, a, d)과 대응하는 연결수 분포 쌍 $(\lambda_{UEP}, \rho_{UEP})$은 가우스 근사법을 적용할 수 있다. UEP 가우스 근사법은 스레드홀드를 계산할 수 있고, UEP-LDPC 코드 앙상블의 UEP 성능을 평가할 수 있다.

[0124] 'l'번째 반복에서, 연결수(d_k) 비트 검사 메시지의 평균$(m_{u,d_c}^{(l)})$은 수학식 39를 이용하여 갱신하다.

수학식 39

$$m_{v,d_k}^{(l)} = \frac{2}{\sigma^2} + (d_k - 1)m_u^{(l-1)} \quad (1 \le k \le s)$$

[0125]

[0126] 또한, 연결수(d_c)에 해당하는 검사 비트 메시지의 평균은 수학식 40을 이용하여 갱신되며, 검사 메시지의 평균은 수학식 41을 이용하여 갱신된다.

수학식 40

$$m_{u,d_c}^{(l)} = \phi^{-1}\left(1 - \left(1 - \sum_{k=1}^{s} \lambda_{d_k} \phi\left(m_{v,d_k}^{(l)}\right)\right)^{d_c - 1}\right)$$

[0127]

수학식 41

$$m_u^{(l)} = m_{u,d_c}^{(l)} = \phi^{-1}\left(1 - \left(1 - \sum_{k=1}^{s} \lambda_{d_k} \phi\left(\frac{2}{\sigma^2} + (d_k - 1)m_u^{(l-1)}\right)\right)^{d_c - 1}\right)$$

[0128]

[0129] 연결수(d_k) 비트의 오차 확률$(P_{e,d_k}^{(l)})$은 수학식 42와 같이 나타내며, 코드어의 평균 오차 확률 $(P_e^{(l)})$은 수학식 43과 같이 나타낸다.

수학식 42

$$P_{e,d_k}^{(l)} = Q\left(\sqrt{\frac{1}{2}\left(\frac{2}{\sigma^2} + d_k \cdot m_u^{(l)}\right)}\right)$$

[0130]

수학식 43

$$P_e^{(l)} = \sum_{k=1}^{s} \alpha_k Q\left(\sqrt{\frac{1}{2}\left(\frac{2}{\sigma^2} + d_k \cdot m_u^{(l)}\right)}\right)$$

[0131]

[0132] 여기서, 'a_k'는 연결수(d_k) 비트 노드의 프랙션이다.

[0133] AWGN 채널 상에서, 스레스홀드는 'σ' 에 대응되고, 'E_b/N_0 (dB)'에 의해 더 간결하게 표현될 수 있다. 스레스홀드 즉, 'E_b/N_0 (dB)' 은 수학식 44와 같이 나타낸다.

수학식 44

$$E_b/N_0 = 10 \log_{10}(1/(2R\sigma^2)) \text{ (dB)}$$

[0134]

[0135] 여기서 'R' 은 코드율이다.

[0136] UEP-LDPC 코드 앙상블의 종합적인 성능은 '$(E_b/N_0)^*$'의 값에 대응한다. 즉, 'E_b/N_0(dB)' 에서 기준 스레드홀드 보다 작은 스레스홀드는 UEP-LDPC 코드 앙상블의 오차 정정 성능을 높일 수 있다.

[0137] UEP-LDPC 코드 앙상블$(R, \ a, \ d)$의 UEP 특성들은 연결수(d_k)의 오차 확률과 'l'번째 반복에서의 평균 오차 확률간에 비율$(r(d_k, \ l, \ \sigma^2))$을 이용하여 평가한다. 여기서, 비율은 수학식 45와 같이 나타낸다.

수학식 45

$$r(d_k, l, \sigma^2) = \frac{P_e^{(l)}}{P_{e,d_k}^{(l)}} = \frac{\sum_{i=1}^{s} \alpha_i Q\left(\sqrt{\frac{1}{2}\left(\frac{2}{\sigma^2} + d_i \cdot m_u^{(l)}\right)}\right)}{Q\left(\sqrt{\frac{1}{2}\left(\frac{2}{\sigma^2} + d_k \cdot m_u^{(l)}\right)}\right)} \quad (1 \le k \le s)$$

[0138]

[0139] 여기서, 상기 비율$(r(d_k, \ l, \ \sigma^2))$은 UEP-LDPC 코드 앙상블$(R, \ a, \ d)$, l 및 노이즈 파워 (σ^2)의 함수이다. 또한, 확률간의 비율은 상기 수학식 45을 토대로 수학식 46과 같이 나타낼 수 있다.

수학식 46

$$r(d_k, \ l, \ \sigma^2) = G(R, \ a, \ d)$$

[0140]

[0141] 여기서, G()는 클로즈드 폼(closed form) 표현을 포함하지 않으며, 수치적으로만 포함될 수 있다. 또한, 수학식 46은 수학식 44를 토대로 수학식 47과 같이 나타낼 수 있다.

수학식 47

[0142] $$r(d_k,\ I,\ E_b/N_0) = G(R,\ a,\ d)$$

[0143] 여기서, 'I' 및 'E_b/N_0(dB)'가 주어지는 경우, '$r(d_k,\ I,\ E_b/N_0)$'는 연결수 (d_k) 비트들의 오차 방지 능력을 나타낼 수 있다.

[0144] 이러한 파라미터를 이용하면, UEP-LDPC 코드 앙상블을 최적화할 수 있다.

[0145] 동시에, AWGN 채널상에서 연결수 분포는 수학식 48과 같은 안정성 조건을 만족해야 한다.

수학식 48

[0146] $$\lambda'(0)\rho'(1) < e^{\frac{1}{2\sigma^2}}$$

[0147] UEP-LDPC 코드 앙상블에서는 상기 수학식 48이 수학식 49와 같은 형태로 안정성 조건을 만족해야 한다.

수학식 49

[0148] $$\lambda_2 \cdot (d_c - 1) < e^{\frac{1}{2\sigma^2}}$$

[0149] 여기서, 'λ_2'는 연결수(2)의 비트 노드의 프랙션이다. 'λ_2'의 최대값은 노이즈 스레스홀드 (σ^*)와 대응한다. 노이즈 스레드홀드는 수학식 50과 같이 나타낸다.

수학식 50

[0150] $$\lambda_2^* = e^{\frac{1}{2\cdot(\sigma^*)^2}} / (d_c - 1)$$

[0151] 다음, 수학식 47과 같이 계산한, 비율 $r(d_k,\ I,\ E_b/N_0)$ 중 파라미터를 선택하여, 오류 제어 코드를 생성하는 방법을 아래와 같이 설명한다.

[0152] **5. UEP-LDPC 코드 설계**

[0153] LDPC 코드의 최적화는 연속적인 공간 파라미터들을 갖는 비선형 함수의 비용을 최소화하기 위한 과제이다. 즉, 좋은 UEP-LDPC 코드 앙상블을 찾는 것은 선형 프로그래밍 문제를 감소시킬 수 있다.

[0154] 본 발명의 실시예에 따른, UEP-LDPC 코드 설계 시, 코드율(R)과 'a'의 시퀀스가 주어진다. 또한, 비트들에서의 오차 방지를 위해, 확정한 'I' 및 노이즈 파워 레벨 'E_b/N_0'에서 벡터 'd'에 대응되는 연결수 분포 쌍이 가장 큰 '$r(d_k,\ I,\ E_b/N_0)$'를 야기하는 벡터 'd'를 파라미터로 선택한다.

[0155] 여기서, UEP-LDPC 코드 앙상블을 아래 두 단계의 방법을 통해 찾을 수 있다.

[0156] 예를 들어, 상기 두 단계의 방법은 코드율(R), 프랙션 시퀀스$(\boldsymbol{a})$, 변수 노드의 최대 연결수 $(d_{v\max})$, 반복 횟수(l) 및 스레스홀드(E_b/N_0)가 주어지는 경우에 해당한다.

[0157] 제1 단계 절차는 상기 수학식 11을 통해, 수학식 51의 조건을 만족하는 벡터 $'\boldsymbol{d}'$의 앙상블 $'D'$를 확인한다.

수학식 51

[0158] $d_c,\ d_1,\ d_2, \cdots,\ d_s$ are integers, $2 \le d_s < \cdots < d_2 < d_1 \le d_{v\max}$

[0159] 다음, 상기 수학식 6, 7, 10 및 11을 이용하여 $'D'$의 원소$(\boldsymbol{d})$의 연결수 분포 쌍$(\boldsymbol{\lambda},\ \boldsymbol{\rho})$을 계산한다.

[0160] 제2 단계 절차는 상기 수학식 39 내지 상기 수학식 45에 해당하는 UEP 가우스 근사법을 토대로 $r(d_k,\ l,\ E_b/N_0)$를 계산한다.

[0161] 다음, $'D'$의 원소이며, 수학식 52의 조건을 만족하는 벡터 $'\boldsymbol{d}'$를 선택한다.

수학식 52

[0162] $$\boldsymbol{d} = \max\ r\ (d_1,\ l,\ E_b/N_0)$$

[0163] 예를 들어, 오차 방지 레벨(k)의 's'가 3인 경우, UEP-LDPC 코드 앙상블은 수학식 53과 같이 정의된다.

수학식 53

[0164] $$\boldsymbol{a} = (a_1,\ a_2,\ 1-R)$$

$$\boldsymbol{d} = (d_1, d_2, d_3)$$

[0165] 여기서, $'a_1 + a_2 = R'$이고, $'d_1'$은 오차 방지 레벨이 1과 관련될 수 있다. 여기서, $'\boldsymbol{a} = (a_1,\ a_2,\ 1-R)=(1/16,7/16,1/2)'$라고 가정할 수 있다.

[0166] AWGN 채널상에서 결정된 코드율(예를 들어, R = 1/2)의 연결수 시퀀스는 $'d_{v\max}=20,30,40,50'$에서 하기 표 1과 같이 코드를 설계할 수 있다.

표 1

d_{vmax}	20	30	40	50
d_1	20	28	36	46
d_2	4	4	4	6
d_3	2	2	2	2
d_c	8	9	10	13
λ_{d1}	0.312500	0.388889	0.450000	0.442308
λ_{d2}	0.437500	0.388889	0.350000	0.403846
λ_{d3}	0.250000	0.222222	0.200000	0.153846
λ_2^*	0.2558	0.2246	0.2006	0.1549
σ^*	0.9264	0.9245	0.9200	0.8980
$(E_b/N_0)^*$(dB)	0.664029	0.681862	0.724243	0.934473

[0167]

[0168] 표 1을 참고하면, 각각의 열은 하나의 UEP-LDPC 코드 앙상블에 대응한다. 각각의 코드에 대해 노이즈 스레드홀드(σ^*), dB에서 대응된 $'(E_b/N_0)^*'$, $'\lambda'$ 및 $'\rho'$의 계수도 주어진다. 또한, $'\lambda_2^*'$는 상기 수학식 48에서 기재하는 안정성 조건을 만족하는 $'\lambda_2'$의 최대값이다. 즉, 모든 UEP-LDPC 코드 앙상블에서 $'\lambda_2' < '\lambda_2^*'$임을 확인할 수 있으며, 이러한 연결수 시퀀스가 안정성 조건을 만족함을 확인할 수 있다.

[0169] 이상, 본 발명의 실시예에 따른 UEP-LDPC 코드 앙상블에 해당하는 연결수 분포 쌍을 계산하고, 계산한 연결수 분포 쌍에 대해 가우스 근사법을 적용하여 성능을 평가하는 방법을 설명하였다. 또한, 성능을 평가한 결과, 즉 비율 중 파라미터를 선택하여, 오류 제어 코드를 설계하는 방법을 구체적으로 설명하였다.

[0170] 이상 구체적인 설명을 토대로 본 발명의 실시예에 따른 오류 제어 코드 생성 시스템을 도 1을 참조하여 설명한다.

[0171] 도 1은 본 발명의 실시예에 따른 오류 제어 코드 생성 시스템을 개략적으로 나타내는 도면이다.

[0172] 도 1을 참고하면, 오류 제어 코드 생성 시스템(100)은 연결수 분포 쌍 계산부(110), 비율 계산부(120), 벡터 선택부(130) 및 코드 생성부(140)을 포함한다.

[0173] 먼저, 패리티 검사 행렬은 "0"이 아닌 엔트리가 $'j'$번째 행과 $'i'$번째 열에 존재하며, $'i'$번째 변수 노드와 $'j'$번째 검사 노드를 포함하며, 노드 사이에는 엣지가 존재한다. 또한, 각 노드에 연결된 엣지의 수는 노드의 연결수와 동일하다.

[0174] 연결수 분포 쌍 계산부(110)는 변수 노드의 최대 연결수(d_{vmax}) 보다 작은 정수를 원소로 포함하는 벡터에 대해 각각 연결수 분포 쌍$(\lambda_{UEP}, \rho_{UEP})$을 계산한다. 여기서, 연결수 분포 쌍$(\lambda_{UEP}, \rho_{UEP})$은 변수 노드의 연결수 분포 및 검사 노드의 연결수 분포를 포함한다. 또한, 변수 노드의 연결수 분포는 연결수(i)를 갖는 변수 노드로부터 나오는 엣지의 프랙션 집합이며, 검사 노드의 연결수 분포는 연결수(j)를 갖는 검사 노드로부터 나오는 엣지의 프랙션 집합이다. 이때, $'i'$ 및 $'j'$는 각각 변수 노드의 최대 연결수 및 검사 노드의 최대 연결수 이하의 정수이다.

[0175] 본 발명의 실시예에 따른, 연결수 분포 쌍 계산부(110)는 상기 수학식 51의 조건을 만족하는 벡터에 대해 연결수 분포 쌍을 계산한다. 또한, 본 발명의 실시예에 따른, 연결수 분포 쌍 계산부(110)는 변수 노드의 연결수 분포를 상기 수학식 6 및 7을 토대로 계산하고, 검사 노드의 연결수 분포를 상기 수학식 10 및 11을 토대로 계산하며, 이에 한정하지 않는다.

[0176] 비율 계산부(120)는 계산한 연결수 분포 쌍$(\lambda_{UEP}, \rho_{UEP})$에 대해 가우스 근사법을 이용하여, 임의의 연결수(d_k)에서의 오차 확률$(P_{e,d_k}^{(l)})$과 임의의 반복 예를 들어, $'l'$ 번째 반복에서 코드어의 평균 오차 확률 $(P_e^{(l)})$

간의 비율 $r(d_k,\ 1,\ E_b/N_0)$ 을 계산한다. 이때, 연결수 분포 쌍은 평균 및 분산을 포함하는 가우스 변수와 연결수를 포함한다.

[0177] 본 발명의 실시예에 따른 비율 계산부(120)는 상기 수학식 39 내지 상기 수학식 45를 이용하여 비율을 계산하며, 이에 한정되지 않는다.

[0178] 벡터 선택부(130)는 계산한 비율 중 가장 큰 비율에 대응하는 벡터'd'를 파라미터로 선택한다. 예를 들어, 벡터 선택부(130)는 상기 수학식 52의 조건을 만족하는 벡터'd'를 파라미터로 선택한다.

[0179] 코드 생성부(140)는 코드율(R), 오차 방지 레벨(k)에 따른 비트들의 프랙션 및 선택된 벡터를 파라미터로 하는 생성함수를 통해 UEP-LDPC 기반의 오류 제어 코드를 생성한다.

[0180] 다음, 본 발명의 실시예에 따른 오류 제어 코드 생성 방법을 도 2를 참조하여 상세하게 설명한다.

[0181] 도 2는 본 발명의 실시예에 따른 오류 제어 코드 생성 방법을 나타내는 흐름도이다.

[0182] 도 2를 참고하면, 오류 제어 코드 생성 시스템(100)은 변수 노드의 최대 연결수 (d_{vmax}) 보다 작은 정수를 원소로 포함하는 벡터에 대해 각각 연결수 분포 쌍 $(\lambda_{UEP},\ \rho_{UEP})$ 을 계산한다(S210).

[0183] 오류 제어 코드 생성 시스템(100)은 계산한 연결수 분포 쌍 $(\lambda_{UEP},\ \rho_{UEP})$ 에 대해 가우스 근사법을 이용하여, 임의의 연결수 (d_k) 에서의 오차 확률 $(P_{e,d_k}^{(l)})$ 과, 임의의 반복에서 코드어의 평균 오차 확률 $(P_e^{(l)})$ 간의 비율 $r(d_k,\ 1,\ E_b/N_0)$ 을 계산한다(S220).

[0184] 오류 제어 코드 생성 시스템(100)은 계산한 비율 중 가장 큰 비율에 대응하는 벡터 'd'를 파라미터로 선택한다(S230).

[0185] 오류 제어 코드 생성 시스템(100)은 코드율(R), 오차 방지 레벨(k)에 따른 비트들의 프랙션 및 선택된 벡터를 파라미터로 하는 생성함수를 통해 UEP-LDPC 기반의 오류 제어 코드를 생성한다(S240).

[0186] 이로써, 본 발명의 실시예에 따른 오류 제어 코드 생성 시스템 및 그 방법은 UEP 방식과 LDPC 방식을 이용하여 보다 효율적으로 오류 제어 코드를 생성할 수 있다.

[0187] 이상에서 설명한 본 발명의 실시예는 장치 및 방법을 통해서만 구현이 되는 것은 아니며, 본 발명의 실시예의 구성에 대응하는 기능을 실현하는 프로그램 또는 그 프로그램이 기록된 기록 매체를 통해 구현될 수도 있으며, 이러한 구현은 앞서 설명한 실시예의 기재로부터 본 발명이 속하는 기술분야의 전문가라면 쉽게 구현할 수 있는 것이다.

[0188] 이상에서 본 발명의 실시예에 대하여 상세하게 설명하였지만 본 발명의 권리범위는 이에 한정되는 것은 아니고 다음의 청구범위에서 정의하고 있는 본 발명의 기본 개념을 이용한 당업자의 여러 변형 및 개량 형태 또한 본 발명의 권리범위에 속하는 것이다.

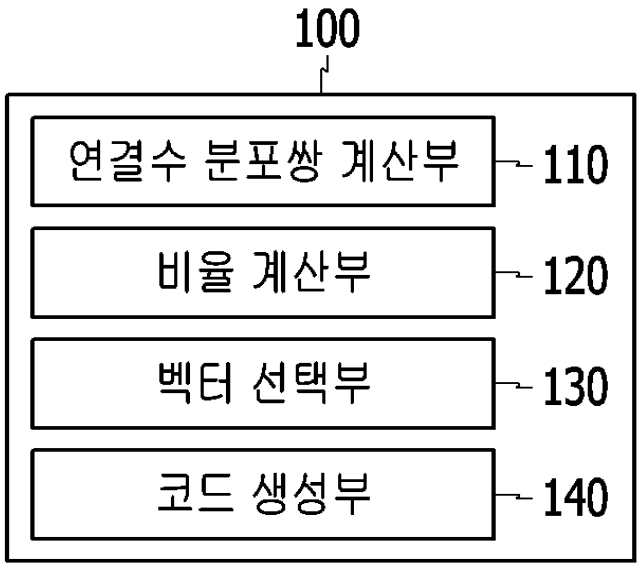
도면의 간단한 설명

[0189] 도 1은 본 발명의 실시예에 따른 오류 제어 코드 생성 시스템을 개략적으로 나타내는 도면이다.

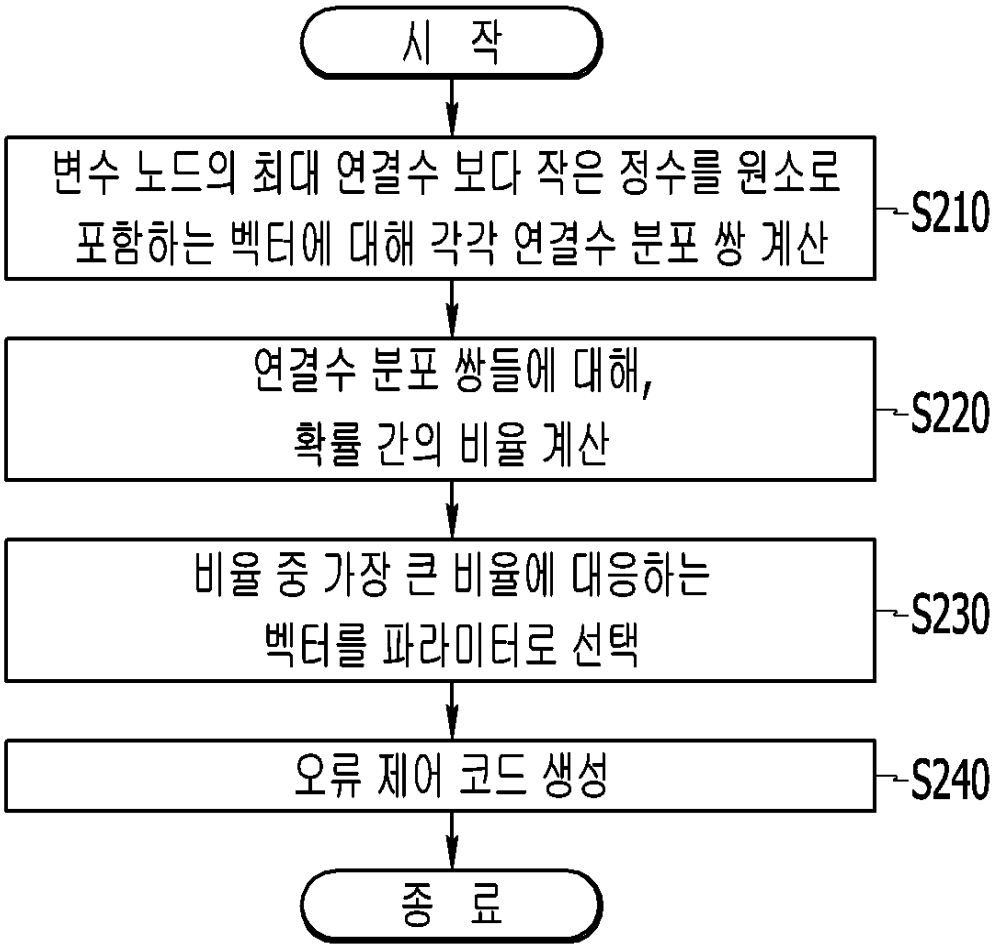
[0190] 도 2는 본 발명의 실시예에 따른 오류 제어 코드 생성 방법을 나타내는 흐름도이다.

도면

도면1



도면2


시 작
변수 노드의 최대 연결수 보다 작은 정수를 원소로 포함하는 벡터에 대해 각각 연결수 분포 쌍 계산
S210
연결수 분포 쌍들에 대해, 확률 간의 비율 계산
S220
비율 중 가장 큰 비율에 대응하는 벡터를 파라미터로 선택
S230
오류 제어 코드 생성
S240
종 료