

논문 2016-53-12-4

적응형 강제 수렴 기법을 이용한 저전력 LDPC 복호기

(Low Power LDPC Decoder Using Adaptive
Forced Convergence algorithm)

최 병 준*, 배 정 현**, 선우 명 훈***

(Byung Jun Choi, JeongHyeon Bae, and Myung Hoon Sunwoo[©])

요 약

LDPC 부호는 최근 Wi-Fi, WiGig, 10GBased-T Ethernet과 같은 최신 통신 표준들에서 순방향 오류 정정 부호로 적용되고 있다. 그러나 LDPC 부호는 우수한 성능을 위해 큰 블록 크기와 많은 반복 복호 횟수를 요구되어 많은 연산량을 필요로 한다. 이러한 문제를 해결하기 위해서 오류 정정 성능을 크게 저하시키지 않고 연산량을 감소시키기 위한 다양한 연구가 지속적으로 수행되고 있다. 본 논문에서는 특정 조건식을 만족하는 변수 노드나 체크 노드에 대하여 메시지 갱신 연산을 중지시켜 연산 복잡도를 감소시키는 AFC 알고리즘을 제안한다.

Abstract

LDPC code has been applied in recent communication standards, such as Wi-Fi, WiGig, 10GBased-T Ethernet as a forward error correction code. However, LDPC code is required a large amount of computational complexity due to large iterations and block lengths for high performances. To solve this problem, various research has been continuously performed for reducing computational complexity. In this paper, we propose AFC algorithm to deactivate the variable and check node for reduce the computational complexity.

Keywords: LDPC, Forced convergence algorithm, Low power, Threshold value, Min-sum

I. 서 론

최근 5세대 이동통신 기술의 핵심 오류 정정 부호로 각광 받는 LDPC (Low Density Parity Check) 부호는 1962년 Gallager에 의해 처음 제안되었다^[1]. 하지만 당시 하드웨어 기술로는 LDPC 부호의 설계 및 검증이 어려워 구현 불가능하였지만, 1996년 Mackay와 Neal에 의해 재발견되어 현재까지 각광받고 있다^[2]. LDPC 부호가 가진 Shannon 한계 근처의 오류 정정 능력과

LDPC 복호기가 가진 높은 병렬처리 가능성으로 인해 많은 주목을 받고 있다.

LDPC 부호는 10GBase-T (10 Gigabit Ethernet)^[3], DVB-S2 (Digital Video Broadcasting-Satellite-Second Generation)^[4], Wi-Fi (IEEE 802.11n^[5], IEEE 802.11ac^[6]) 그리고 WiGig (IEEE 802.11ad^[7])와 같은 최신의 유·무선 통신 표준들에 FEC (Forward Error Correction) 표준으로 채택되고 있으며 HDD (Hard disk drive), SSD (Solid State Drive)와 같은 각종 저장

* 정회원, (주)옵티시스 (Opticis)

** 정회원, 아주대학교 전자공학과 (Department of Electrical and Computer Engineering, Ajou University)

*** 평생회원, 아주대학교 전자공학과 (Department of Electrical and Computer Engineering, Ajou University)

© Corresponding Author (E-mail: sunwoo@ajou.ac.kr)

※ 본 연구는 산업통상자원부의 산업핵심기술개발사업으로 지원된 연구 결과입니다. (10053186, 소형 이동로봇 플랫폼 제어 통합형 SoC 개발, 10060200 도전방지 및 보안 강화형 스마트미터 SOC 개발)

※ 본 연구는 미래창조과학부 및 정보통신기술진흥센터의 대학ICT연구센터 육성지원사업 (IITP-2016-R2718-16-0015)의 결과로 수행되었음.

매체에도 널리 사용되고 있다.

오프셋 MS (Min-sum)^[8], 정규화된 MS 알고리즘^[8]과 변수 노드값을 감소시켜 반복 복호에 따른 변수 노드 발산을 억제한다. 그러나, 강제 수렴 알고리즘은 임계값을 설정하여 반복 복호시 필요한 연산량을 감소시킨다.

기존 강제 수렴 (Forced Convergence) 알고리즘은 미리 설정해 놓은 체크 노드 임계값보다 큰 체크 노드를 비활성화 시키고, 변수 노드 임계값보다 큰 변수 노드를 비활성화 시켜 복호 과정은 연산량 감소시킨다.

제안하는 강제 수렴 알고리즘은 기존 강제 수렴 알고리즘과 달리 변수 노드 임계값만을 설정하여 변수 노드를 비활성화 시킨다. 체크 노드에 연결된 변수 노드들이 모두 비활성화 되면 체크 노드 갱신 역시 연산이 되지 않아 비활성화 된다.

본 논문의 구성은 제 2장에서는 일반적인 LDPC 복호 과정과 기존 강제 수렴 알고리즘^[9~14]에 대해 설명하고, 제 3장에서는 제안하는 강제 수렴 알고리즘에 대해 설명한다. 제 4장에서 MATLAB Simulator를 이용하여 제안하는 강제 수렴 알고리즘의 성능검증 결과를 기술하였으며, 제 5장에서 결론을 맺는다.

II. MS 알고리즘 및 기존 강제수렴 알고리즘

1. LDPC 부호의 MS 복호 알고리즘

(1) 초기화 단계

모든 (i, j) 위치의 α_{ij} , β_{ij} 는 식 (2.1)와 식 (2.2)와 같이 0과 LLR값인 λ_j 로 초기화 된다.

$$\alpha_{ij} = 0 \quad (2.1)$$

$$\beta_{ij} = \lambda_j \quad (2.2)$$

(2) 체크 노드 갱신 단계

체크 노드 갱신은 식 (2.3)와 같이 체크 노드의 이웃 변수 노드들 중에서 전달된 β_{ij} 값을 제외한 나머지 $\beta_{ij'}$ 값들을 이용하여 α_{ij} 값을 계산한다.

$$\alpha_{ij} = \prod_{j' \in V(i) \setminus j} \text{sign}(\beta_{ij'}) \times \min_{j' \in V(i) \setminus j} (|\beta_{ij'}|) \quad (2.3)$$

(3) 변수 노드 갱신 단계

변수 노드 갱신은 식 (2.4)와 같이 변수 노드의 이웃 체크 노드들 중에서 전달된 α_{ij} 값을 제외한 나머지 $\alpha_{ij'}$ 값들과 초기값 λ_j 를 이용하여 β_{ij} 값을 계산한다.

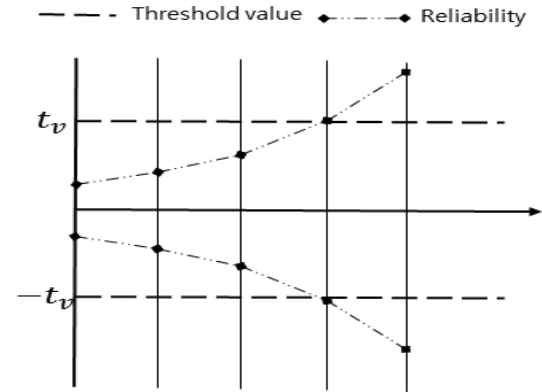


그림 1. 반복 복호 횟수에 따른 신뢰도 변화
Fig. 1. Difference of reliability in accordance with iteration decoding number.

$$\beta_{ij} = \lambda_j + \sum_{i' \in C(j) \setminus i} \alpha_{i'j} \quad (2.4)$$

(4) 판정 단계

코드워드의 비트 값을 갱신하기 위해 식 (2.5)과 같이 z_j 를 연산한다. z_j 는 해당 변수 노드와 이웃한 체크 노드들의 α_{ij} 값들과 기존 변수 노드 값을 모두 더한다.

$$z_j = \lambda_j + \sum_{i' \in C(j)} \alpha_{i'j} \quad (2.5)$$

각 z_j 를 식 (2.6)로 판별하여 복호된 코드워드 $\hat{Z} = \{\hat{z}_1, \hat{z}_2, \dots, \hat{z}_N\}$ 을 얻는다.

$$\hat{z}_j = \begin{cases} 1, & \text{if } z_j \leq 0 \\ 0, & \text{if } z_j > 0 \end{cases} \quad (2.6)$$

만약, $H \cdot \hat{Z}^T = 0$ 을 만족한다면 \hat{Z} 를 유효한 코드워드로 판정하여 복호 과정을 중단하며, 만족되지 않으면 복호과정을 반복하며, 최대 반복 복호 값 I_{max} 에 도달하면 복호를 중단하고 코드워드 \hat{Z} 를 출력한다.

2. 기존 강제 수렴 알고리즘

기존 강제 수렴 알고리즘은 복호 과정 중 체크 노드 값과 변수 노드 값이 발산하는 성질을 이용하여 체크 노드와 변수 노드를 각각 비활성 시키는 방법을 제안한다.

그림 1는 반복 복호 횟수에 따라 증가하는 신뢰도 값의 평균을 나타낸다. 따라서, 높은 신뢰도를 가지는 변수 노드들과 체크 노드들은 이미 복호가 완료된 것으로 간주하여 비활성시켜 복호과정에서 연산량 감소를 가져온다. 이러한 비활성화된 노드를 각각 ‘비활성 변수 노드’, ‘비활성 체크 노드’라 정의하고 이를 판별하기 위해 미리 정의된 임계값 t_v 과 t_c 를 사용한다. t_v 와 t_c 의 비

표 1. $|\beta_{ij}|$ 가 선택될 확률 ($|\beta_{ij}| > t_v$)

t_v	E_b/N_0				
	2 dB	2.5 dB	3 dB	3.5 dB	4 dB
5	0%	0.000785%	0.013238%	0.038083%	0.053334%
6	0%	0.000643%	0.011722%	0.035252%	0.050135%
7	0%	0.000345%	0.009290%	0.032529%	0.046992%

교 연산 수식은 각각 (2.7), (2.8)과 같다.

$$|z_j| > t_v \tag{2.7}$$

$$|\alpha_{ij}| > t_c \tag{2.8}$$

j 번째 비트의 신뢰도를 의미하는 $|z_j|$ 가 t_v 보다 크다면 충분히 높은 신뢰값이며, 이미 수렴한 것으로 간주하여, 해당 j 번째 변수 노드를 비활성 변수 노드로 선언한다. 기존 강제 수렴 알고리즘은 변수 노드 갱신뿐만 아니라 체크 노드 갱신 시의 연산량을 줄이기 위해 체크 노드에 대해서도 강제 수렴 조건 만족 여부를 판단하여, 강제 수렴 조건을 만족하는 체크 노드는 비활성 체크 노드로 설정하여 더 이상 해당 체크 노드 갱신을 수행하지 않는 방법이 제안되었다. 이와 같이, 체크 노드와 변수 노드 모두가 강제 수렴 조건을 만족하는지 여부에 기초하여, 강제 수렴 조건을 만족하는 체크 노드와 변수 노드를 모두 비활성화시킴으로써 복호화 연산량을 줄이는 알고리즘을 강제 수렴 알고리즘이라 한다.

III. 제안하는 강제 수렴 알고리즘

1. 단순화한 강제 수렴 알고리즘

제안하는 강제 수렴 알고리즘은 기존 강제 수렴 알고리즘과 달리 체크 노드 비활성화 식을 사용하지 않고, 변수 노드 비활성화 식만을 이용하여 변수 노드뿐만 아니라 체크 노드도 비활성 한다.

위의 표 1은 체크 노드 값이 체크 노드 임계값보다 클 확률을 SNR에 따라 나타낸다. 미리 정의한 체크 노드 임계값이 5일 때 2.5 dB부터 4dB까지 0.5 dB씩 증가해감에 따라 비활성화될 확률은 0%, 0.000785%, 0.013238%, 0.038083%, 0.053334%로 미미하게 증가되며 비활성화 될 확률은 거의 없다. 그러므로 기존 강제 수렴 조건에서 체크 노드 갱신 연산량이 줄었다고 보기에 힘든 문제점이 있어, 이를 개선한 새로운 강제 수렴 알고리즘을 제안한다.

그림 2에서 회색 원으로 표시한 변수 노드 V_2 는 비

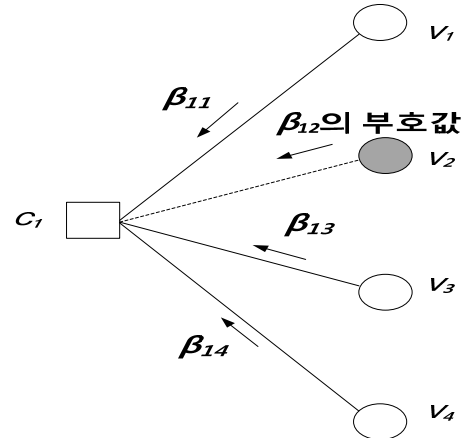


그림 2. 제안하는 체크 노드 갱신 연산
Fig. 2. Proposed check node operation.

활성 변수 노드라 가정하고, 점선으로 표현된 예지는 체크 노드와 비활성 변수 노드 간에 연결된 예지를 의미한다. 제안하는 체크 노드 갱신 연산에서는 비활성 변수 노드 연산 결과의 크기값을 필요로 하지 않기 때문에, 비활성 변수 노드인 V_2 에서는 체크 노드 C_1 으로 β_{12} 의 부호값만을 전달한다. 나머지 변수 노드들 V_1, V_3, V_4 에서는 기존과 같은 방법으로 각각 $\beta_{11}, \beta_{13}, \beta_{14}$ 를 체크 노드 C_1 으로 전달한다. 식 (4.4)는 그림 2의 체크 노드 C_1 에서 이웃 변수 노드들로부터 받은 입력으로 연산하는 수식들을 나타낸다.

$$\begin{aligned} \alpha_{11} &= \text{sign}(\beta_{12})\text{sign}(\beta_{13})\text{sign}(\beta_{14})\min(|\beta_{13}|, |\beta_{14}|) \\ \alpha_{13} &= \text{sign}(\beta_{11})\text{sign}(\beta_{12})\text{sign}(\beta_{14})\min(|\beta_{11}|, |\beta_{14}|) \\ \alpha_{14} &= \text{sign}(\beta_{11})\text{sign}(\beta_{12})\text{sign}(\beta_{13})\min(|\beta_{11}|, |\beta_{13}|) \end{aligned} \tag{3.1}$$

식 (3.1)의 첫 번째 식에서는, α_{11} 의 부호값은 $\beta_{12}, \beta_{13}, \beta_{14}$ 의 부호값을 모두 곱한 값으로 결정하고, α_{11} 의 크기값은 β_{13} 와 β_{14} 의 크기를 비교하여 얻은 최소값으로 결정한다. 이와 같이 α_{13}, α_{14} 를 연산한다.

그림 3을 보면, 체크 노드 C_1 에서는 변수 노드 V_1, V_3, V_4 에 각각 $\alpha_{11}, \alpha_{13}, \alpha_{14}$ 를 전달하지만 V_2 에는 α_{12} 를 전달하지 않는다. 식 (3.1)에서도 α_{12} 에 대한 연산이 제외되었는데, 비활성 변수 노드에서는 변수 노드 갱신 연산을 진행하지 않기 때문에 해당 연산을 위한 입력값을 전달받을 필요가 없다. 따라서, 그림 3와 같이 해당 변수 노드로 전달될 α_{12} 를 연산에서 제외한다.

2. 적응형 강제 수렴 알고리즘

제안하는 적응형 강제 수렴 알고리즘에서는 변수 노

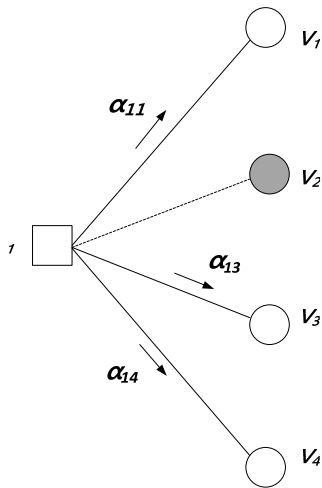


그림 3. 제안하는 체크 노드 갱신 연산의 출력
Fig. 3. Output of proposed check node operation.

드 및 체크 노드 비활성화를 더욱 가속화 하는 알고리즘을 제안한다. 비활성 변수 노드의 수를 더욱 증가시킨다면 변수 노드와 체크 노드의 연산량을 동시에 더 줄일 수 있음을 알 수 있다. 본 논문에서는 제안하는 강제 수렴 알고리즘의 변수 노드 연산량을 더 효율적으로 감소시키기 위해, 식 (3.2)와 같이 변수 노드의 강제 수렴 조건식에 사용되는 t_v 의 크기를 동적으로 결정하는 방법을 새롭게 제안하였다.

$$t_v = t_0 - (t \times i) \quad (3.2)$$

t_v 는 변수 노드의 강제 수렴 조건식에 사용되는 비교 값이다. t_0 는 첫 번째 반복복호에서 사용되는 t_v 의 초기값이며, t 는 각 반복복호마다 t_v 를 감소시킬 상수이다. i 는 반복복호의 횟수를 의미한다. 그림 4는 제안하는 조건식인 식 (3.2)를 기존의 조건식인 식(3.1)과 비교한 그림이다.

그림 4는 제안하는 적응형 강제수렴 알고리즘의 동적 변수노드 임계값을 나타낸다. 반복 복호 횟수가 증가함에 따라 변수 노드 임계값을 감소하여 기존 강제수렴 알고리즘 보다 더욱 많은 비활성 노드를 생성시킨다. 이미 발산하려는 방향성을 가진 변수 노드들은 반복 복호를 계속하여도 크기가 작아질 확률이 거의 없기 때문에, 더 작은 크기의 값을 가지더라도 비활성 시키도록 한다.

따라서 식 (3.2)와 같이 t_v 의 크기를 조금씩 줄여 가면, 기존 방식에 비해 비활성 변수 노드를 더 많이 생성하게 되어 변수 노드의 연산량을 줄일 수 있다. 또한, 이 방식을 제안하는 체크 노드 갱신과 연동한다면 체크

표 2. $t_v = t_c = 8$ 에 대한 체크 노드 연산량 감소율 비교

알고리즘	(t_v, t_c)	부호율	체크 노드 연산량 감소율		
			$E_b/N_0 = 5 \text{ dB}$	$E_b/N_0 = 6 \text{ dB}$	$E_b/N_0 = 7 \text{ dB}$
강제 수렴 알고리즘 [10]	(8, 8)	1/2	0.89%	0.91%	0.71%
		5/8	0.06%	0.09%	0.09%
		3/4	0%	0.01%	0.01%
		13/16	0%	0%	0%
제안하는 강제 수렴 알고리즘	(8, -)	1/2	37.03%	26.58%	15.58%
		5/8	33.71%	24.21%	14.13%
		3/4	27.04%	24.70%	15.36%
		13/16	18.36%	22.61%	13.88%

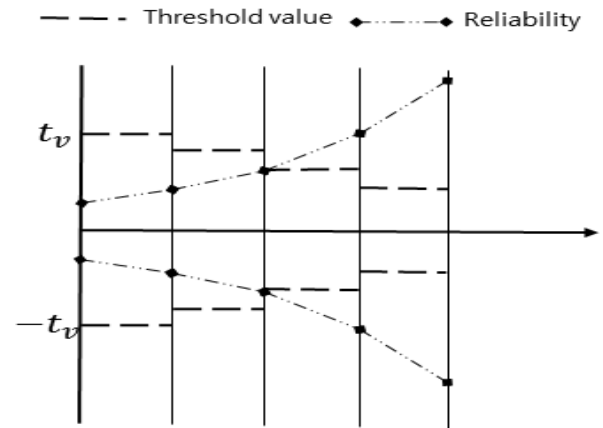


그림 4. 제안하는 적응형 강제수렴 알고리즘에 따른 변수노드 임계값 변화

Fig. 4. Difference of reliability in accordance with proposed adaptive forced convergence algorithm.

노드의 연산량도 함께 줄일 수 있다. 이미 방향성을 가지게 된 메시지들을 비활성 시키는 것이기 때문에 성능 저하도 거의 없을 것이라 예상할 수 있다.

IV. 실험 결과

제안하는 강제 수렴 알고리즘과 기존 알고리즘들은 MATLAB을 사용하여 FER (Frame error rate) 성능과 연산량을 비교하였다. IEEE 802.11ad^[7] 표준 LDPC 부호를 사용하였고, 부호율은 1/2, 5/8, 3/4, 13/16의 4가지 부호율을 지원한다. 부호화된 데이터는 BPSK로 변조되어 AWGN 환경 하에 실험하였으며 최대 반복 복호 횟수는 5회로 제한하였다.

표 2는 MS 알고리즘의 체크 노드 연산량을 기준으로 하여 기존 강제 수렴 알고리즘과 제안하는 강제 수렴 알고리즘의 체크 노드 연산량 감소율을 보여준다.

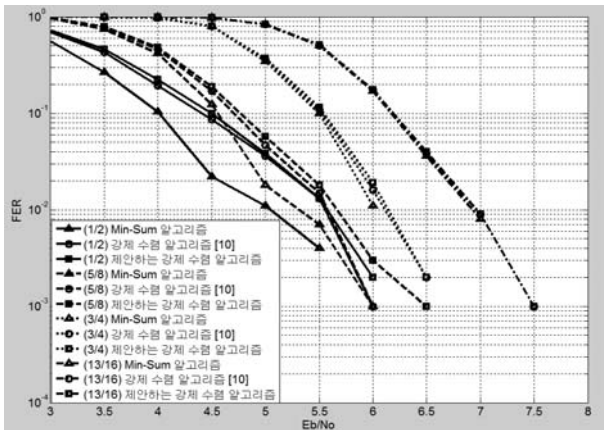


그림 5. 기존 알고리즘과 제안하는 강제수렴 알고리즘 비교

Fig. 5. Comparison with existing forced convergence algorithm and proposed adaptive forced convergence algorithm.

기존 강제 수렴 알고리즘에서는 체크 노드의 강제 수렴 조건식을 거의 만족하지 않으므로 비활성 체크 노드를 생성하지 못하지만, 제안하는 강제 수렴 알고리즘에서는 제안하는 체크 노드 갱신 연산식을 사용하여 효과적으로 체크 노드 연산량을 줄이는 것을 볼 수 있다.

그림 5는 기존 MS 알고리즘, 기존 강제수렴 알고리즘^[10], 제안하는 강제 수렴 알고리즘에 따른 성능 비교를 나타낸다. 기존 강제수렴 알고리즘과 제안하는 강제 수렴 알고리즘은 $t_v = 8$ 로 하고, 강제수렴 알고리즘은 $t = 0.5$ 로 설정하여 FER 성능 비교를 나타낸다. 그림 4에서 보인바와 같이 MS 알고리즘, 기존 강제 수렴 알고리즘, 제안하는 능동형 강제 수렴 알고리즘은 큰 성능열화 없이 변수 노드 연산량, 체크 노드 연산량을 줄이는 것을 확인 할 수 있다.

V. 결 론

본 논문에서는 기존 강제 수렴 알고리즘에서 효과적으로 연산량을 감소시키기 위해, 새로운 체크 노드 갱신 연산식과 변수 노드의 강제 수렴 조건식을 제안하였다. 본 논문에서 제안하는 체크 노드 갱신 연산은 기존 연산 방법과 달리 체크 노드를 비활성 시키기 위해 강제 수렴 조건식을 사용하지 않고 이웃한 비활성 변수 노드를 활용한다. 제안하는 체크 노드 갱신 연산은 비활성 변수 노드의 결과 값을 제외하여 체크 노드 갱신을 진행하기 때문에 연산에 필요한 변수 노드 결과 값의 수가 줄어드는 효과를 볼 수 있다. 제안하는 강제 수

렴 알고리즘에 t_v 와 Δt 를 각각 8과 0.5로 설정할 경우, 제안하는 알고리즘은 1/2의 부호율에서 MS 알고리즘과 비교하여 0.1 dB의 성능 열화를 보인다. 이때, 제안하는 강제 수렴 알고리즘은 기존 강제 수렴 알고리즘과 비교하여 $E_b/N_0=5 \sim 7$ dB 구간의 각 부호율에서 평균적으로 2.62%, 3.55%, 5.77%, 7.39%의 변수 노드 연산량 감소율과 26.85%, 25.64%, 25.40%, 22.45%의 체크 노드 연산량 감소율을 얻는다.

REFERENCES

- [1] R. G. Gallager, "Low-density parity check codes," IRE Transactions on Information Theory, vol. IT-8, pp. 21-28, Jan. 1962.
- [2] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," IEEE Trans. Inform. Theory, vol. 45, no. 2, pp. 399-431, Mar. 1999.
- [3] IEEE P802.3an, 10GBASE-T task force. <http://www.ieee802.org/3/an>.
- [4] T.T.S.I. digital video broadcasting (DVB) second generation framing structure for broadband satellite applications. <http://www.dvd.org>.
- [5] IEEE 802.11n. Wireless LAN Medium Access Control and Physical Layer specifications: Enhancements for Higher Throughput. IEEE P802.16n/D1.0, Mar 2006.
- [6] R. Stacey, E. Perahia, A. Stephens et al. IEEE P802.11ac. Specification framework for TGac. IEEE802.11 09/0992r21. January 2011.
- [7] IEEE Draft Standard for Local and Metropolitan Area Networks - Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications - Amendment 3: Enhancements for Very High Throughput in the 60 GHz Band, IEEE Std 802.11ad. 2011.
- [8] Xiaofu Wu, Yue Song, Ming Jiang, and Chunming Zhao, "Adaptive-Normalized/Offset Min-Sum Algorithm," IEEE Communications Letters, VOL. 14, NO. 7, JULY 2010
- [9] E. Zimmermann, G. Fettweis, P. Pattisapu and P. K. Bora, "Reduced complexity LDPC decoding using forced convergence," 7th Int. Symp. On Wireless Personal Multimedia Communications, Sep. 2004.
- [10] E. Zimmermann, W. Rave and G.K. Fettweis, "Forced convergence decoding of LDPC Codes --- EXIT chart analysis and combination with

node complexity reduction techniques,” 11th Eur. Conf. Wireless (EW’2005), Apr. 2005.

[11] E. Zimmermann, P. Pattisapu, and G. Fettweis, “Bit-flipping post-processing for forced convergence decoding of LDPC codes,” presented at the Eur. Signal Processing Conf., Antalya, Turkey, 2005.

[12] Aoyama R, Kaji Y., “Improvement of the forced-convergence decoding for LDPC codes,” Int. Symp. On Information Theory and Its Applications(ISITA 2008), Auckland, 2008.

[13] Fan J X, Yang H W, “A new forced convergence decoding scheme for LDPC codes,” IEEE International Conference on Communications Technology and Applications (ICCTA), Beijing, 2009.

[14] Byung J. Choi and Myung H. Sunwoo, “Simplified Forced Convergence Decoding Algorithm for Low Power LDPC Decoders,” In Proc. IEEE Asia Pacific Conference on Circuits and Systems, Nov. 2014.

저 자 소 개



최 병 준(정회원)
2008년~2011년 아주대학교
전자공학부 학사
2012년~2014년 아주대학교
전자공학부 석사
2014년~현재 (주)유평티시스
전임연구원

<주관심분야 : LDPC, 오류정정 코드, 신호처리>



배 정 현(정회원)
2009년~2012년 아주대학교
전자공학부 학사
2013년~현재 아주대학교
전자공학부 석·박 통합과
정

<주관심분야 : LDPC, 오류정정 코드, 초소형 캡
슐내시경 송수신기, 신호처리, 통신, 저전력 의료
기기 디바이스 설계>



선우 명 훈(평생회원)
1980년 서강대학교 전자공학 학사
졸업.
1982년 한국과학기술원 전자공학
석사 졸업.
1982년~1985년 한국전자통신연구
소(ETRI) 연구원

1985년~1990년 Univ. of Texas at Austin 전자공
학 박사.

1990년~1992년 Motorola, DSP Chip Division
(미국)

1992년~1996년 아주대학교 전기전자공학부 조교수

1996년~2001년 아주대학교 전자공학부 부교수

2001년~현재 아주대학교 전자공학부 교수

2011년~현재 IEEE Fellow Member

2011년~현재 IEEE CASS Board of Governor

2012년~2014년 대한전자공학회 반도체 소사이어
티 회장

2014년~현재 대한전자공학회 반도체 소사이어티
자문위원

2016년~현재 아주대학교 ITRC uDSD 연구센터장

<주관심분야 : ICT SoC 설계, ASIP 설계, 저전
력 의료기기 디바이스 설계>