

여러 입력 여러 출력 시스템에서 길이 먼저 살펴보기와 가지 길이 문턱값을 바탕으로 둔 준최적 복호

정회원 안 태 훈*, 강 현 구*, 오 중 호*, 종신회원 송 익 호*, 윤 석 호**

Near ML Decoding Based on Metric-First Searching and Branch Length Threshold for Multiple Input Multiple Output Systems

Taehun An*, Hyun Gu Kang*, Jongho Oh* *Regular Members,*
Ickho Song*, Seokho Yoon** *Lifelong Members*

요 약

이 논문에서는 여러 입력 여러 출력 시스템에 알맞은 준최적 복호 방법을 하나 다룬다. 제안한 방법은 길이 먼저 살펴보기를 바탕으로 둔 바, 슈노르-오히너 늘여놓기와 가지 길이 문턱값을 써서 신호를 복호하는 데 드는 계산량을 줄인다. 제안한 방법의 비트오류율 성능은 최적 성능에 매우 가까우면서 계산량은 다른 준최적 복호기들보다 적다는 것을 모의실험으로 보인다.

Key Words : multiple input multiple output, near maximum likelihood decoder, metric-first search, branch length threshold, Schnorr-Euchner enumeration

ABSTRACT

In this paper, we address a near maximum likelihood (ML) scheme for the decoding of multiple input multiple output systems. Based on the metric-first search method and by employing Schnorr-Euchner enumeration and branch length thresholds, the proposed scheme provides reduced computational complexity. The proposed scheme is shown by simulation to have lower computational complexity than other near ML decoders while maintaining the bit error rate very close to the ML performance.

1. 서 론

무선 통신에서 여러 입력 여러 출력 (multiple input multiple output) 시스템은 한 입력 한 출력 (single input single output) 시스템보다 주파수 효율이 높을 뿐 아니라^[1], 대역폭이나 전송 전력을 늘리지 않고도 시스템 용량을 높일 수 있기 때문에 다음 세대 이동통신에서 중요한 기술들 가운데 하나로 꼽

히고 있다. 이제까지 여러 연구에서 여러 입력 여러 출력 시스템에 알맞은 복호기들을 다루었는데^[1-4], 가장 비슷함 (maximum likelihood) 기준을 따라 신호를 복호하면 이론적으로 비트오류율 (bit error rate) 성능이 가장 뛰어나다. 이때, 모든 가능성을 다 살펴보는 복호기를 써서 신호를 복호하면 계산량이 너무 많으므로 계산량이 그보다 적은 공 복호기가 (sphere decoder)^[3] 제안된 바 있다. 최근에는

* 이 논문은 교육과학기술부의 재원으로 한국과학재단이 선정하여 지원하는 국가지정연구실 사업 R0A-2005-000-10005-0과 지식경제부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업 IITA-2009-C1090-0902-0010의 결과 가운데 하나입니다.

* 한국과학기술원 전기및전자공학과 (tahn@Sejong.kaist.ac.kr, khg@Sejong.kaist.ac.kr, jh@Sejong.kaist.ac.kr, i.song@ieee.org)

** 성균관대학교 정보통신공학부 (syoon@skku.edu)

논문번호: KICS2009-03-101, 접수일자: 2009년 3월 10일, 최종논문접수일자: 2009년 6월 3일

공 복호기보다 계산량을 더 줄인 너비 먼저 복호기가 (breadth-first signal decoder)^[4] 연구되었다. 하지만, 가장 비슷함 복호기는 실제 시스템에 쓰기에 아직도 계산량이 많다는 문제가 있다.

이에, 비트오류율 성능은 조금 떨어지더라도 계산량을 많이 줄일 수 있는 준최적 복호기를 여러 사람이 생각하였다^[5-9]. 여러 준최적 복호기들 가운데 슈노르-오히너2 (Schnorr-Euchner2) 방법과^[5] 반지름이 커지는 알고리즘은^[9] 길이 먼저 살펴보기에서 안테나 가 많아질 때 계산량이 지수적으로 늘어나는 것을 막고자 제안되었다. 보기를 들면, 슈노르-오히너2 방법은 슈노르-오히너 늘어놓기를 바탕으로 계산량을 줄이고, 반지름이 커지는 알고리즘은 살펴보기 공간을 확률적으로 잘라서 계산량을 줄인다. 이 두 방법은 계산량이 적고 비트오류율 성능이 좋지만, 슈노르-오히너2 방법에서는 신호대잡음비를 추정해야하고, 반지름이 커지는 알고리즘에서는 쓸모 있는 점을 찾지 못하면 반지름을 더 크게 잡고 처음부터 다시 나무를 살펴봐야 한다.

한편, 너비 먼저 살펴보기를 쓰는 복호기들 가운데 큐알 분해-엠 (QRD-M) 방법인^[10] 엠-알고리즘을^[11] 바탕으로 하여 나무의 뿌리에서 한 층씩 내려가며 나무를 살핀다. 이때, 층마다 마디 길이가 가장 짧은 마디를 몇 개만 남겨 나무를 살피는 데 드는 계산량을 줄인다. 최근에는 큐알 분해-엠 방법의 계산량을 더욱 줄이고자 적응 큐알 분해-엠 방법과^[7] 효율적인 큐알 분해-엠 방법이^[8] 제안되었다. 효율적인 큐알 분해-엠 방법은 층마다 남기는 마디 가운데 마디 길이가 문턱값보다 큰 마디를 버림으로써 계산량을 더 줄인다. 하지만, 문턱값을 얻으려면 부분 결정 피드백 등화 (decision feedback equalization) 풀이와 그 유클리드 거리를 층마다 계산해야 한다.

길이 먼저 살펴보기의 하나인 큐알 분해-더미 (QRD-Stack) 방법인^[6] 더미 알고리즘을^[11] 바탕으로 하며, 마디 길이가 가장 짧은 마디에서만 가지들을 이어 나무를 살핀다. 이 방법은 여러 마디들을 한꺼번에 살피기 때문에 신호대잡음비가 높을 때는 계산량이 꽤 적지만, 신호대잡음비가 낮아지면 나무를 살피는 동안 위층으로 자주 되돌아가므로 계산량이 많아진다.

이 논문에서는 길이 먼저 살펴보기를 바탕으로 슈노르-오히너 늘어놓기와 가지 길이 (branch length) 문턱값을 쓰는 새로운 준최적 복호 방법을 제안한다. 제안한 방법은 첫째 층까지 예상 길이를 어렵잡아 얻은 문턱값을 가지 길이와 견주어 나무를 살피

는 동안 위층으로 되돌아가는 것을 되도록 줄인다. 제안한 방법은 계산량은 다른 준최적 방법들보다 적으며, 비트오류율 성능은 가장 비슷함 성능에 매우 가깝다.

II. 시스템 모형

보내는 안테나를 N_T 개, 받는 안테나를 N_R 개 쓰는 여러 입력 여러 출력 시스템을 생각하자. 송신기에서 입력 데이터를 N_T 개로 나누고, 보내는 안테나를 거쳐 무선 채널로 보내면, 받는 안테나에서는 이들 신호의 조합을 받는다. 이때, 보내는 안테나에서 보내는 신호들은 별자리 수가 같은 직교진폭변조 (quadrature amplitude modulation) 신호들이라고 두자. 그러면, j 째 받는 안테나에서 받은 복소 신호를 \tilde{y}_j 라고 할 때, 받은 신호 벡터 $\tilde{\mathbf{y}} = [\tilde{y}_1, \tilde{y}_2, \dots, \tilde{y}_{N_R}]^T$ 의 이산시간 바탕대역 모형은

$$\tilde{\mathbf{y}} = \tilde{\mathbf{H}}\tilde{\mathbf{s}} + \tilde{\mathbf{v}} \quad (1)$$

처럼 나타낼 수 있다. 여기서, 위첨자 T 는 벡터 전치, $\tilde{\mathbf{H}}$ 는 평균이 0이고 분산이 1이며 독립이고 분포가 같은^[12] 복소 정규 확률변수들이 이루는 $N_R \times N_T$ 채널 행렬, $\tilde{\mathbf{s}} = [\tilde{s}_1, \tilde{s}_2, \dots, \tilde{s}_{N_T}]^T$ 는 보낸 신호 벡터, $\tilde{\mathbf{v}} = [\tilde{v}_1, \tilde{v}_2, \dots, \tilde{v}_{N_R}]^T$ 는 평균이 0이고 분산이 σ^2 이며 독립이고 분포가 같은 복소 정규 확률변수들이 이루는 벡터이다. 한편, 수신기는 채널 행렬을 완전히 알고 있다고 둔다.

이제, 실수 부분을 $\Re(\cdot)$, 허수 부분을 $\Im(\cdot)$ 로 나타내면, 복소 바탕대역 모형 (1)은

$$\begin{aligned} \mathbf{y} &= \begin{pmatrix} \Re(\tilde{\mathbf{y}}) \\ \Im(\tilde{\mathbf{y}}) \end{pmatrix} \\ &= \begin{pmatrix} \Re(\tilde{\mathbf{H}}) - \Im(\tilde{\mathbf{H}}) \\ \Im(\tilde{\mathbf{H}}) \quad \Re(\tilde{\mathbf{H}}) \end{pmatrix} \begin{pmatrix} \Re(\tilde{\mathbf{s}}) \\ \Im(\tilde{\mathbf{s}}) \end{pmatrix} + \begin{pmatrix} \Re(\tilde{\mathbf{v}}) \\ \Im(\tilde{\mathbf{v}}) \end{pmatrix} \\ &= \mathbf{H}\mathbf{s} + \mathbf{v} \end{aligned} \quad (2)$$

처럼 실수 꼴로 바꾸어 쓸 수 있다. 식 (2)에서 $\mathbf{y} = [y_1, y_2, \dots, y_n]^T$ 는 받은 신호 벡터이고, $\mathbf{s} = [s_1, s_2, \dots, s_m]^T$ 는 보낸 신호 벡터이며, $\mathbf{v} = [v_1, v_2, \dots, v_n]^T$ 는 평균이 0이고 분산이 $\sigma^2/2$ 이며 독립이고 분포가 같은 덧셈꼴 정규 잡음들이 이루는 벡터이다. 여기서, $m = 2N_T$ 이고 $n = 2N_R$ 이다. 이 논문에서는 $n = m$ 이라고 둔다.

먼저, 채널 행렬 \mathbf{H} 를

$$H = QR \quad (3)$$

처럼 분해한다고 하자. 여기서, Q 는

$$Q^T Q = I \quad (4)$$

인 $m \times m$ 단위행렬이고 $R = [r_{i,j}]$ 은 $m \times m$ 위쪽세모행렬이다. 이제, (2)의 두 변에 Q^T 를 왼쪽에서 곱하면,

$$r = R\underline{s} + w \quad (5)$$

를 얻는다. 여기서, $\underline{r} = Q^T \underline{y} = [r_1, r_2, \dots, r_m]^T$ 이고 $w = Q^T \underline{v}$ 이다. 한편, (4)를 새기면 (5)에 보인 잡음 성분 w 와 (2)에 보인 잡음 성분 v 의 통계적인 특성이 같음을 알 수 있다.

보내는 안테나에서 신호 별자리

$$\Omega = \left\{ -\frac{\sqrt{L}-1}{2}, -\frac{\sqrt{L}-3}{2}, \dots, \frac{\sqrt{L}-3}{2}, \frac{\sqrt{L}-1}{2} \right\}, \quad L = 4, 16, \dots \quad (6)$$

을 써서 L 진 직교진폭변조한 신호를 보낸다고 하자. 그러면, Ω 를 끝없이 늘린 것을

$$\Omega_\infty = \left\{ a + \frac{1}{2} : a \in Z \right\} \quad (7)$$

이라 하고, 정수 집합을 Z 라 하면, (5)에 보인 벡터 집합 $\{R\underline{s}\}$ 는 R 이 만드는 무한 격자

$$A(R) = \{R\underline{s} : \underline{s} \in \Omega_\infty^m\} \quad (8)$$

의 부분 집합이라고 볼 수 있다^[3]. 그러면, 벡터 \underline{r} 은 정규 잡음 w 를 더한 격자점이라고 생각할 수 있다. 따라서, 벡터 \underline{r} 과 행렬 R 이 주어졌을 때, 최적 풀이 $\hat{\underline{s}}$ 는

$$\begin{aligned} \hat{\underline{s}} &= \underset{\underline{s} \in \Omega_\infty^m}{\operatorname{argmin}} \|\underline{r} - R\underline{s}\|^2 \\ &= \underset{\underline{s} \in \Omega_\infty^m}{\operatorname{argmin}} \sum_{i=1}^m \left(r_i - \sum_{j=1}^m r_{i,j} s_j \right)^2 \end{aligned} \quad (9)$$

처럼 얻을 수 있다. 여기서, $\|\cdot\|$ 는 유클리드 거리를 나타낸다.

III. 제안한 복호 방법

3.1 나무 살펴보기

여러 입력 여러 출력 시스템에서 신호를 복호할 때 나무 열개가 자주 쓰인다. 뿌리에서 시작하고 $(m+1)$ 층인 \sqrt{L} 진 나무를 생각하자. 여기서, 뿌리는 가장 높은 층인 $(m+1)$ 째 층을 가리킨다. 그러면, 이 나무의 $(k+1)$ 째 층과 k 째 층 사이에 있는 가지는 신호 벡터 \underline{s} 의 k 째 원소 s_k 를 나타내고, 나무의 마디는 그 마디와 뿌리를 잇는 가지들이 이루는 벡터를 나타낸다. 이제, k 째 층에 있는 l 째 마디를 $(m-k+1)$ 차원 벡터

$$\underline{s}_k^{(l)} = [s_{k,k}^{(l)}, s_{k+1,k}^{(l)}, \dots, s_{m,k}^{(l)}]^T \quad (10)$$

처럼 나타내자. 이때, $k=1, 2, \dots, m$ 이고 $l=1, 2, \dots, \sqrt{L}^{m-k+1}$ 이며, $\underline{s}_{m+1}^{(1)}$ 은 뿌리를 나타낸다.

마디 $\underline{s}_k^{(l)}$ 과 그 부모마디

$$\begin{aligned} \underline{s}_{k+1}^{(p)} &= [s_{k+1,k+1}^{(p)}, s_{k+2,k+1}^{(p)}, \dots, s_{m,k+1}^{(p)}]^T \\ &= [s_{k+1,k}^{(l)}, s_{k+2,k}^{(l)}, \dots, s_{m,k}^{(l)}]^T \end{aligned} \quad (11)$$

사이에 있는 가지의 길이를

$$\begin{aligned} \phi(\underline{s}_k^{(l)}) &= \left(r_k - \sum_{j=k}^m r_{k,j} s_{j,k}^{(l)} \right)^2 \\ &= e_k^2(\underline{s}_k^{(l)}) \end{aligned} \quad (12)$$

처럼 뜻매김하면, $\underline{s}_k^{(l)}$ 의 마디 길이는

$$\begin{aligned} \Phi(\underline{s}_k^{(l)}) &= \sum_{i=k}^m \phi(\underline{s}_i^{(p_i)}) \\ &= \sum_{i=k}^m \left(r_i - \sum_{j=i}^m r_{i,j} s_{j,i}^{(p_i)} \right)^2 \\ &= \sum_{i=k}^m \left(r_i - \sum_{j=i}^m r_{i,j} s_{j,k}^{(l)} \right)^2 \\ &= \sum_{i=k}^m e_i^2(\underline{s}_k^{(l)}) \end{aligned} \quad (13)$$

처럼 얻을 수 있다. 바꾸어 말하면, 마디의 마디 길이는 그 마디와 뿌리를 잇는 가지들의 길이를 모두 더한 것이다. 식 (13)에서, $p_k = l$ 이고 $p_{m+1} = 1$ 이라 할 때 $\underline{s}_{i+1}^{(p_{i+1})}$ 은 $\underline{s}_i^{(p_i)}$ 의 부모마디이고,

$$e_i(s_k^{(l)}) = r_i - \sum_{j=\max(i,k)}^m r_{i,j} s_{j,k}^{(l)} \quad (14)$$

이다. 한편, (13)의 둘째 줄에서 세째 줄을 얻을 때에는 $i = k, k+1, \dots, m$ 이고 $j = i, i+1, \dots, m$ 일 때 $s_{j,i}^{(p_i)} = s_{j,k}^{(l)}$ 임을 썼다.

이제, (9)에서 보인 최적 풀이 $\hat{s} \in \Omega^m$ 을 찾는 문제는 나무의 첫째 층에 있는 벡터들 $\{s_1^{(1)}, s_1^{(2)}, \dots, s_1^{(\sqrt{L^m})}\}$ 가운데 마디 길이 $\phi(s_1^{(l)})$ 이 가장 짧은 마디 $s_1^{(l)}$ 을 찾는 문제와 같다는 것을 알 수 있다.

제안한 복호 방법을 자세히 다루기 전에 몇 가지를 뜻매김하자.

- 앞마디: 아래층에 있는 어떠한 마디와도 이어지지 않은 마디
- 가장 깊은 마디: 앞마디들 가운데 가장 낮은 층에 있는 마디
- 가장 좋은 마디: 앞마디들 가운데 마디 길이가 가장 짧은 마디
- 가장 좋은 가지: 한 마디에서 아직 생각하지 않은 가지들 가운데 길이가 가장 짧은 가지

나무를 뿌리에서 살펴며 내려오는 동안 가장 깊은 마디는 하나보다 많을 수 있으나, 가장 좋은 가지와 가장 좋은 마디는 확률 1로 하나뿐이라는 것을 알 수 있다.

3.2 길이 먼저 살펴보기와 슈노르-오히너 늘어놓기

길이 먼저 살펴보기에서는 앞마디들을 서로 견주어 가장 좋은 마디를 고르고, 가장 좋은 마디에서 바로 아래층으로 가지들을 이어 새로운 앞마디들을 만든다. 이 과정을 첫째 층에서 가장 좋은 마디를 찾을 때까지 되풀이한다. 일반적으로, 길이 먼저 살펴보기는 신호대잡음비가 높을 때는 계산량이 많지 않고 성능도 좋지만 신호대잡음비가 낮을 때는 자주 위층으로 되돌아가기 때문에 계산량이 많다. 게다가, 가장 좋은 마디를 찾으면, 가장 좋은 마디에서 이을 수 있는 가지들을 모두 잇기 때문에 마디들을 쓸데없이 많이 이어서 생각하게 된다. 따라서, 별자리 신호수와 안테나 개수가 늘어나면 계산량이 매우 많아진다.

가장 좋은 마디에서 생각하는 가지 수는 슈노르-오히너 늘어놓기를 써서 가장 좋은 가지부터 한 번에 하나씩 가지를 이음으로써 줄일 수 있다. 슈노르-오히너 늘어놓기는 나무 살펴보기에서 효율을 높이는 데 자주 쓰인다^[13].

먼저, Ω 의 원소 가운데 x 와 가장 가까운 것을 $Q(x)$ 라고 쓰자. 보기를 들어, $\Omega = \{-1.5, -0.5, 0.5, 1.5\}$ 일 때, $Q(-0.2) = -0.5$ 이고 $Q(2.1) = 1.5$ 이다. 그러면, $s_k^{(l)}$ 의 가장 좋은 가지 $S(s_k^{(l)})$ 을

$$S(s_k^{(l)}) = Q\left(\frac{e_{k-1}(s_k^{(l)})}{r_{k-1,k-1}}\right) \quad (15)$$

처럼 얻을 뒤, 슈노르-오히너 늘어놓기에 따라 $s_k^{(l)}$ 에서 $(k-1)$ 째 층에 있는 마디들로 이을 수 있는 가지들 $\{B_j(s_k^{(l)})\}_{j=1}^{\sqrt{L}}$ 을 가지 길이의 오름차순으로

$$S(s_k^{(l)}) \leq \frac{e_{k-1}(s_k^{(l)})}{r_{k-1,k-1}} \text{ 일 때 } \{S(s_k^{(l)}), S(s_k^{(l)})+1, S(s_k^{(l)})-1, S(s_k^{(l)})+2, S(s_k^{(l)})-2, \dots\} \cap \Omega, \quad (16)$$

$$S(s_k^{(l)}) > \frac{e_{k-1}(s_k^{(l)})}{r_{k-1,k-1}} \text{ 일 때 } \{S(s_k^{(l)}), S(s_k^{(l)})-1, S(s_k^{(l)})+1, S(s_k^{(l)})-2, S(s_k^{(l)})+2, \dots\} \cap \Omega$$

처럼 줄 세울 수 있다. 슈노르-오히너 늘어놓기를 써서 가지들을 다시 줄 세움으로써 가지들을 좀 더 조직적으로 다룰 수 있고, 쓸만한 가지들만 살펴볼 확률을 높일 수 있다.

3.3 제안한 복호 방법

제안한 방법은 길이 먼저 살펴보기와 슈노르-오히너 늘어놓기를 바탕으로 뿌리에서 차례대로 나무를 살펴며 내려온다. 곧, 제안한 방법은 (가) 가장 좋은 마디를 찾고, (나) 가장 좋은 마디를 살펴볼 만하지 결정하고, (다) 가장 좋은 마디의 가장 좋은 가지부터 한 번에 하나씩 가지를 잇는다.

3.3.1 단계 1: 가장 좋은 마디를 고르기

모든 앞마디들의 길이를 견주어서 가장 좋은 마디를 고른다.

3.3.2 단계 2: 가장 좋은 마디가 있는 층 확인하기

단계 1에서 고른 가장 좋은 마디가 가장 깊은 마디이면 단계 2-1을 밟고, 그렇지 않으면, 단계 2-2를 밟는다.

(가) 단계 2-1: 가장 좋은 마디가 첫째 층에 있는 마디인지 확인하기

가장 좋은 마디가 첫째 층에 있는 마디이면, 아직

살피지 않은 마디들의 마디 길이는 가장 좋은 마디의 마디 길이보다 길거나 같으므로, 가장 좋은 마디가 곧 얻으려는 마디이다. 가장 좋은 마디가 첫째 층이 아닌 다른 층에 있으면, 단계 3으로 가서 새로운 앞마디를 만들어 나무를 다시 살핀다.

(나) 단계 2-2: 가장 좋은 마디가 살피볼 만한 것인지 알아보기

가장 좋은 마디가 가장 깊은 마디가 아니면, 첫째 층까지 갈 것을 생각하여 가장 좋은 마디가 살피볼 만한 것인지 알아본다. 곧, 가장 좋은 마디가 $s_k^{(l)}$ 일 때, $s_k^{(l)}$ 과 그 부모마디 사이에 있는 가지의 길이 $\phi(s_k^{(l)})$ 을 문턱값

$$\gamma_k^2 = \left\{ \frac{\Gamma\left(\frac{k}{2} + 1\right)}{\pi^{\frac{k}{2}}} \prod_{i=1}^k |r_{i,i}| \right\}^{\frac{2}{k}} \quad (17)$$

과 견주어 본다. 이때, $\phi(s_k^{(l)}) \leq \gamma_k^2$ 이면 $s_k^{(l)}$ 을 더 살피볼 만하다고 여기어 단계 3으로 가고, $\phi(s_k^{(l)}) > \gamma_k^2$ 이면 $s_k^{(l)}$ 은 더 살피볼 것이 없다고 여겨 $s_k^{(l)}$ 을 버리고 단계 1로 돌아간다. 이러한 과정은 나무를 쓸데없이 살피는 것을 막고 계산량을 줄여준다.

3.3.3 단계 3: 새로운 앞마디들을 더하기

단계 2를 거쳐 온 가장 좋은 마디를 $s_k^{(l)}$ 이라 쓰고 $s_k^{(l)}$ 의 부모마디를 $s_{k+1}^{(p)}$ 라고 하자. 먼저, $s_k^{(l)}$ 의 가장 좋은 가지 $S(s_k^{(l)})$ 을 (15)처럼 고르고, $s_k^{(l)}$ 과 $(k-1)$ 층에 있는 $s_{k-1}^{(c)} = [S(s_k^{(l)}), s_k^{(l)}]^T$ 를 잇는다. 그 다음, 가지 길이

$$\phi(s_{k-1}^{(c)}) = \{e_{k-1}(s_k^{(l)}) - r_{k-1,k-1} s_{k-1,k-1}^{(c)}\}^2 \quad (18)$$

을 계산하여 $s_{k-1}^{(c)}$ 의 마디 길이 $\phi(s_{k-1}^{(c)})$ 를

$$\Phi(s_{k-1}^{(c)}) = \Phi(s_k^{(l)}) + \phi(s_{k-1}^{(c)}) \quad (19)$$

처럼 얻는다. 여기서, $\Phi(s_k^{(l)})$ 은 k 층에서 이미 계산한 것이다.

(가) 단계 3-1: 부모마디에서 이을 수 있는 가지가 없을 때

가장 좋은 마디 $s_k^{(l)}$ 이 뿌리이거나 $s_{k+1}^{(p)}$ 에서 모든 가지들을 벌써 이었다면, 단계 1로 간다.

(나) 단계 3-2: 부모마디에서 이을 수 있는 가지가 적어도 하나 남아있을 때

이제, $s_k^{(l)}$ 이 뿌리가 아니고 $s_{k+1}^{(p)}$ 에서 이을 수 있는 가지가 적어도 하나 남아있으면, $s_{k+1}^{(p)}$ 에서 가장 좋은 가지를 이어 새로운 앞마디를 만들고, 상황에 따라 마디 길이가 가장 긴 마디를 버린다.

먼저, $s_{k+1}^{(p)}$ 의 $(j+1)$ 째 가지 증분을

$$\begin{aligned} \Delta_{j+1}(s_{k+1}^{(p)}) &= \begin{cases} \operatorname{sgn}\left(\frac{e_k(s_{k+1}^{(p)})}{r_{k,k}} - s_k^{(l)}\right), & j=0, \\ -\Delta_j(s_{k+1}^{(p)}) - \operatorname{sgn}(\Delta_j(s_{k+1}^{(p)})), & j=1,2,\dots \end{cases} \quad (20) \end{aligned}$$

처럼 뜻매김하자. 여기서, 함수 $\operatorname{sgn}(x)$ 는 $x \geq 0$ 이면 $\operatorname{sgn}(x) = 1$ 이고 $x < 0$ 이면 $\operatorname{sgn}(x) = -1$ 이다. 그러면, $s_{k+1}^{(p)}$ 의 b 가지 증분들을 이미 계산하였을 때, $s_{k+1}^{(p)}$ 의 가장 좋은 가지는

$$S(s_{k+1}^{(p)}) = \begin{cases} s_{k,k}^{(l)} + \Delta_{b+1}(s_{k+1}^{(p)}), & \\ s_{k,k}^{(l)} + \Delta_{b+1}(s_{k+1}^{(p)}) \in \Omega \text{ 일 때,} \\ s_{k,k}^{(l)} + \Delta_{b+1}(s_{k+1}^{(p)}) + \Delta_{b+2}(s_{k+1}^{(p)}), & \\ s_{k,k}^{(l)} + \Delta_{b+1}(s_{k+1}^{(p)}) \notin \Omega \text{ 일 때} \end{cases} \quad (21)$$

처럼 얻을 수 있다. 여기서, (15)로 $S(s_{k+1}^{(p)})$ 를 얻을 때 $\Delta_1(s_{k+1}^{(p)})$ 를 미리 계산해 두면, (20)에서 나눗셈 $e_k(s_{k+1}^{(p)})/r_{k,k}$ 가 필요하지 않아 계산량을 줄일 수 있다. 이제, $s_{k+1}^{(p)}$ 와 $s_k^{(l')} = [S(s_{k+1}^{(p)}), s_{k+1}^{(p)}]^T$ 를 잇고, 가지 길이 $\phi(s_k^{(l')})$ 를 계산하여 $s_k^{(l')}$ 의 마디 길이 $\Phi(s_k^{(l')})$ 를

$$\begin{aligned} \Phi(s_k^{(l')}) &= \Phi(s_{k+1}^{(p)}) + \phi(s_k^{(l')}) \\ &= \Phi(s_k^{(l)}) - \phi(s_k^{(l)}) + \phi(s_k^{(l')}) \quad (22) \end{aligned}$$

처럼 얻는다. 여기서, $\Phi(s_k^{(l)})$ 과 $\phi(s_k^{(l)})$ 은 $s_k^{(l)}$ 과 $s_{k+1}^{(p)}$ 를 이을 때 이미 계산하여 저장해 둔 것이다. 곧, $\Phi(s_{k+1}^{(p)})$ 를 저장하지 않고 $\Phi(s_k^{(l')})$ 를 계산할 수 있는 것이다. 다음에, 앞마디가 N 개보다 많으면 (곧, $(N+1)$ 개이면), $s_{k-1}^{(c)}$ 를 뺀 앞마디들 가운데 마디 길이가 가장 긴 마디를 하나 버려, 앞마디를 $s_{k-1}^{(c)}$ 를 포함하여 N 개만 남긴다. 그 다음, 단계 1로 돌아간다.

3.3.4 문턱값

여기서, (17)에 보인 문턱값 γ_k^2 을 쓰는 까닭을 살펴보자. 물리적으로 해석하면 문턱값 γ_k^2 은 k 째 층에 있는 가장 좋은 마디의 부모마디에서 첫째 층까지 가는 길의 예상 길이이다. 이때, 예상 길이는 가장 좋은 마디의 부모마디와 첫째 층에 있는 마디를 잇는 모든 길들 가운데 가장 짧은 길을 생각하여 아래와 같이 얻을 수 있다.

가장 좋은 마디 $s_k^{(l)}$ 의 부모마디 $s_{k+1}^{(p)}$ 에서 첫째 층으로 가는 가장 짧은 길을 생각하자. 그 길을 나타내는 마디 가운데에서 첫째 층에 있는 것을

$$\underline{s}_{k+1}^{(f)} = [s_{1,1}^{(f)}, s_{2,1}^{(f)}, \dots, s_{m,1}^{(f)}]^T \quad (23)$$

이라 하자. 여기서, $j = k+1, k+2, \dots, m$ 일 때

$$s_{j,1}^{(f)} = s_{j,k}^{(l)} \quad (24)$$

이다. 식 (12)에서 알 수 있듯이 가지 길이는 늘 0보다 크거나 같기 때문에 $\underline{s}_{k+1}^{(f)}$ 이 가장 짧은 길에 있다면, $s_{k+1}^{(p)}$ 와 $\underline{s}_{k+1}^{(f)}$ 사이 길이 $\mathcal{L}_{k+1,1}$ 은 $s_k^{(l)}$ 과 $s_{k+1}^{(p)}$ 사이 길이 $\phi(s_k^{(l)})$ 보다 길거나 같아야 한다. 곧,

$$\phi(s_k^{(l)}) \leq \mathcal{L}_{k+1,1} \quad (25)$$

이어야 한다. 이제, (24)를 써서 길이 $\mathcal{L}_{k+1,1}$ 을

$$\begin{aligned} \mathcal{L}_{k+1,1} &= \sum_{i=1}^k \left(r_i - \sum_{j=i}^m r_{i,j} s_{j,1}^{(f)} \right)^2 \\ &= \sum_{i=1}^k \left(r_i - \sum_{j=k+1}^m r_{i,j} s_{j,k}^{(l)} - \sum_{j=i}^k r_{i,j} s_{j,1}^{(f)} \right)^2 \\ &= \sum_{i=1}^k \left(e_i(s_{k+1}^{(p)}) - \sum_{j=i}^k r_{i,j} s_{j,1}^{(f)} \right)^2 \\ &= \| \underline{r}'_k - R_k \underline{s}_k \|^2 \end{aligned} \quad (26)$$

처럼 다시 쓰자. 그러면,

$$\underline{s}_k = [s_{1,1}^{(f)}, s_{2,1}^{(f)}, \dots, s_{k,1}^{(f)}]^T \quad (27)$$

이라 할 때, $\mathcal{L}_{k+1,1}$ 은 R 의 부분 행렬

$$R_k = \begin{pmatrix} r_{1,1} & r_{1,2} & \dots & r_{1,k} \\ 0 & r_{2,2} & \dots & r_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & r_{k,k} \end{pmatrix} \quad (28)$$

이 만드는 k 차원 격자

$$\Lambda_k(R_k) = \{ R_k \underline{s}_k : \underline{s}_k \in \Omega^k \} \quad (29)$$

의 격자점 $R_k \underline{s}_k$ 와 고친 신호 벡터

$$\underline{r}'_k = [e_1(s_{k+1}^{(p)}), e_2(s_{k+1}^{(p)}), \dots, e_k(s_{k+1}^{(p)})]^T \quad (30)$$

사이 거리의 제곱으로 생각할 수 있다. 여기서, $s_{k+1}^{(p)}$ 와 첫째 층에 있는 마디들을 잇는 모든 길들 가운데 가장 짧은 길의 길이가 $\mathcal{L}_{k+1,1}$ 이므로 k 차원 격자 $\Lambda_k(R_k)$ 의 모든 격자점들 가운데 $R_k \underline{s}_k$ 가 \underline{r}'_k 에 가장 가까운 격자점이라는 것을 알 수 있다. 곧, \underline{r}'_k 가 격자점 $R_k \underline{s}_k$ 의 보로노이 영역 (Voronoi region)^[13]

$$\begin{aligned} V(\Lambda_k(R_k), \underline{s}_k) &= \{ \underline{r}'_k \in \mathbb{R}^k : \| \underline{r}'_k - R_k \underline{s}_k \|^2 \\ &\leq \| \underline{r}'_k - R_k \tilde{\underline{s}}_k \|^2, \forall R_k \tilde{\underline{s}}_k \in \Lambda_k(R_k) \} \end{aligned} \quad (31)$$

안에 들어있다는 것을 알 수 있다. 여기서, \mathbb{R} 은 실수 집합을 나타낸다. 보로노이 영역 (31)은 $\Lambda_k(R_k)$ 에서 다른 격자점들보다 $R_k \underline{s}_k$ 에 더 가까운 모든 벡터들 \underline{r}'_k 의 집합을 나타낸다.

그런데, 아쉽게도 보로노이 영역의 경계를 정확히 그리는 것은 거의 언제나 매우 어려우므로 k 차원 보로노이 영역 $V(\Lambda_k(R_k), \underline{s}_k)$ 를 부피가 같고 중심이 $R_k \underline{s}_k$ 인 k 차원 공으로 어렵하자. 그뿐만 아니라, 유한 격자에서 보로노이 영역 부피는 격자점의 위치에 따라 바뀌므로 유한 격자 $\Lambda_k(R_k)$ 대신에 무한 격자 $\Lambda(R_k)$ 에서 $R_k \underline{s}_k$ 의 보로노이 영역 부피를 얻도록 하자. 다시 말해서,

$$\text{Vol}(V(\Lambda_k(R_k), \underline{s}_k)) = \text{Vol}(V(\Lambda(R_k), \underline{s}_k)) \quad (32)$$

라고 두자. 이제,

$$\begin{aligned} \text{Vol}(V(\Lambda(R_k), \underline{s}_k)) &= \sqrt{\det(R_k^T R_k)} \\ &= \prod_{i=1}^k |r_{i,i}| \end{aligned} \quad (33)$$

이므로^[14], 반지름이 d 인 k 차원 공의 부피가 $\frac{\pi^{k/2}}{\Gamma(k/2+1)} d^k$ 임을 생각하면,

$$\begin{aligned} d &= \left\{ \frac{\Gamma\left(\frac{k}{2}+1\right)}{\pi^{k/2}} \prod_{i=1}^k |r_{i,i}| \right\}^{1/k} \\ &= \gamma_k \end{aligned} \quad (34)$$

를 얻을 수 있다. 간추리면, r'_k 에서 가장 가까운 격자점의 보로노이 영역을 부피가 같은 k 차원 공으로 어렵해서 얻은 공 반지름의 제곱이 문턱값이다.

이제, k 차원 공으로 어려운 보로노이 영역 안에 있는 점 r'_k 는

$$\|r'_k - R_k s_k\|^2 \leq d^2 \quad (35)$$

를 만족시킨다. 곧, $s_k^{(i)}$ 이 i 번째 층까지 갈 때 그 길이 여유 있으려면

$$\phi(s_k^{(i)}) \leq \gamma_k^2 \quad (36)$$

이어야 함을 (25), (26), (35)에서 알 수 있다. 다시 말해서, (36)을 만족시킨다는 것은 뿌리에서 $s_k^{(i)}$ 까지 길이가 충분히 짧아서 $s_k^{(i)}$ 을 더 살펴볼 만하다는 것을 뜻한다. 한편, $\phi(s_k^{(i)}) > \gamma_k^2$ 이면 $s_k^{(i)}$ 은 짧은 길에서 이미 벗어나 있어서 더 살펴볼 까닭이 없다는 것을 뜻한다.

IV. 성능 평가

4.1 계산량 분석

복호기의 계산량을 견주는 데에 곱셈량이 널리 쓰이므로^[4], 이 논문에서도 제안한 방법과 다른 복호기들의 곱셈량을 얻어 견주어 본다.

복호의 전처리 단계에서 채널 행렬을 분해하는 데 어렵잡아 곱셈이 $2m^3/3$ 번^[15] 들고 $Q^T y$ 를 계산하는 데 곱셈이 m^2 번 든다. 이제, 제안한 방법의 단계마다 드는 곱셈량을 살펴보자. 먼저, 단계 1에서는 곱셈이 필요하지 않다. 다음에, 단계 2에서 문턱값을 얻을 때, $(k-1)$ 번 곱셈하여 k 차원 보로노이 영역의

부피를 계산하고 그 부피에 $\frac{\Gamma(\frac{k}{2}+1)}{\pi^{k/2}}$ 을 곱한다. 여

기서, 상수 $\frac{\Gamma(\frac{k}{2}+1)}{\pi^{k/2}}$, $k=1,2,\dots,m$ 은 수신기에서 이미 알고 있고, (17)에서 k 제곱근을 얻는 것은 $(k-1)$ 번 곱셈하는 것과 계산량이 같다고 두자. 그러면, 문턱값 γ_k^2 을 얻는 데 $(k-1)+1+(k-1)+1=2k$ 번 곱셈해야 한다는 것을 알 수 있다. 한편, 단계 3에서는 가장 좋은 마디와 그 부모마디에서 새로운 잎마디를 만들고 마디 길이를 계산하는 데 각각 $(m-k+4)$ 번

과 2번 곱셈해야 한다.

이제, 제안한 방법을 써서 복호하는 데 드는 최소 곱셈량을 간단히 생각해보자. 먼저, 가장 좋은 때에는 제안한 방법으로 나무를 살피는 동안 위층으로 되돌아가지 않고, 문턱값과 견주지도 않는다. 이때, 제안한 방법은 마디만 $(2m-1)$ 개 이어서 복호하고, 따라서 전처리 단계까지 생각하면,

$$\frac{2}{3}m^3 + \frac{3m^2+9m}{2} - 2 \quad (37)$$

번 곱셈이 필요하다는 것을 알 수 있다.

하지만, 실제 환경에서 나무를 살펴볼 때에는 잡음의 세기에 따라 여러 가지 상황이 일어나기 때문에 최소 곱셈량은 복호기의 계산량을 따지기에 그다지 알맞지 않다.

4.2 모의실험

이제, 전처리 과정에 드는 계산량까지 생각하여 제안한 방법, 큐알 분해-더미 방법, 효율적인 큐알 분해-엠 방법, 슈노르-오히너2 방법, 그리고 반지름이 커지는 알고리즘의 비트오류를 성능과 평균 계산량을 10^6 번 이상 모의실험하여 견주어 보자. 모의실험에서 송신기는 채널 상태 정보를 모르며 모든 심볼들을 같은 에너지 $E_s = E_{tot}/N_T$ 로 보낸다고 두자. 여기서, E_{tot} 는 한 심볼 동안 송신기에서 쓰는 전체 에너지이다. 그러면, 받는 안테나에서 신호대잡음비는 $E_s N_T / \sigma^2$ 라 쓸 수 있다.

그림 1은 여러 복호기의 비트오류율 성능을 안테나 수와 별자리 신호수에 따라 보여준다. 이 그림에서 제안한 방법, 효율적인 큐알 분해-엠 방법, 그리고 큐알 분해-더미 방법에서 L 진 직교진폭변조를 쓸 때 남기는 잎마디 수 N 은 L 로 두어 성능이 최적 성능에 가깝도록 하였다. 이 그림을 살펴보면 여러 준최적 복호기의 비트오류율 성능은 거의 같고, 최적 비트오류율 성능에 매우 가깝다는 것을 알 수 있다.

그림 2.5는 안테나 수와 별자리 신호수가 바뀔 때 여러 복호기의 평균 곱셈량이 어떻게 달라지는지 보여준다. 여기서, 공 복호기의 처음 반지름은 결정된 먹임 등화 방법을 써서 얻었으며, 실선, 파선, 점선은 각각 길이, 너비, 깊이 먼저 살펴보기를 바탕으로 하는 복호기들을 나타낸다. 이 그림들에서 제안한 방법은 다른 준최적 복호기들보다 계산량이 적다는 것을 알 수 있다. 게다가, 제안한 방법의 계산량은 별자리 신호수와 신호대잡음비가 바뀌더라도 크게 영향을

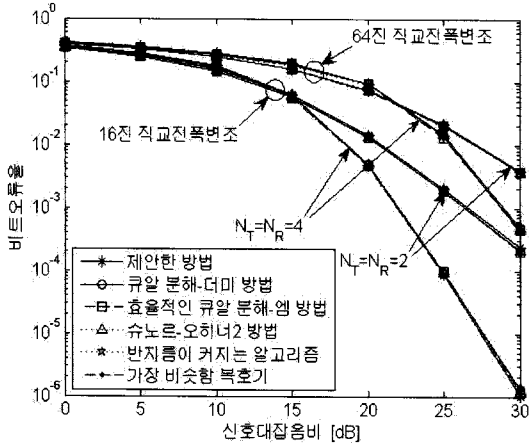


그림 1. 여러 복호기의 비트오류율 성능: $N_T = N_R = 2, 4$ 인 시스템에서 16진, 64진 직교진폭변조하여 신호를 보낼 때

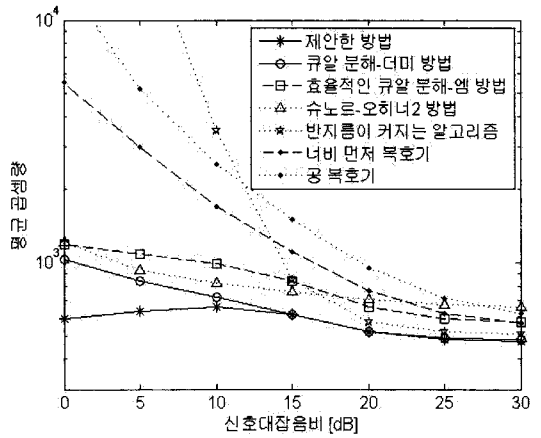


그림 4. 여러 복호기의 평균 복잡량: $N_T = N_R = 4$ 인 시스템에서 16진 직교진폭변조하여 신호를 보낼 때

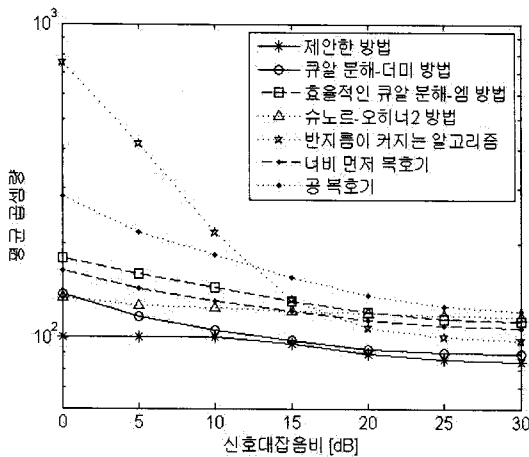


그림 2. 여러 복호기의 평균 복잡량: $N_T = N_R = 2$ 인 시스템에서 16진 직교진폭변조하여 신호를 보낼 때

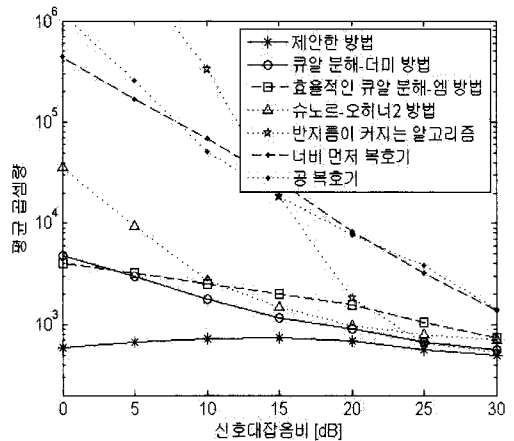


그림 5. 여러 복호기의 평균 복잡량: $N_T = N_R = 4$ 인 시스템에서 64진 직교진폭변조하여 신호를 보낼 때

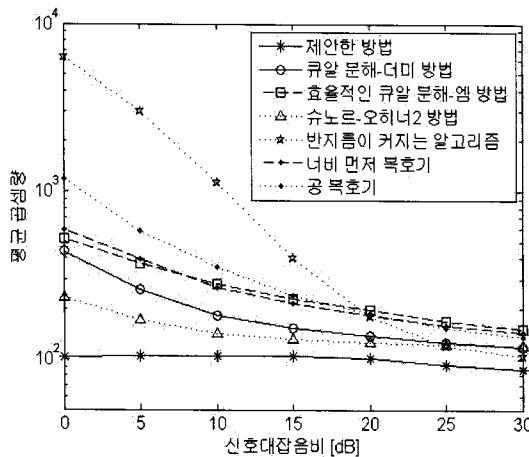


그림 3. 여러 복호기의 평균 복잡량: $N_T = N_R = 2$ 인 시스템에서 64진 직교진폭변조하여 신호를 보낼 때

반지름이 커지는 알고리즘은 복잡량이 줄어들며, 계산량 이득은 신호대잡음비가 낮을 때 더욱 두드러진다. 여기서, 큐알 분해-더미 방법도 길이 먼저 살펴보기를 바탕으로 하지만, 큐알 분해-더미 방법의 계산량은 별자리 신호수가 많아질수록 늘어난다는 것을 알 수 있다.

4.3 직관적 해석

제한한 방법은 신호대잡음비가 낮거나 높을 때 계산량이 적다. 이는 다음과 같이 설명할 수 있다. 첫째, 신호 세기가 일정할 때 신호대잡음비가 낮아질수록 평균 가지 길이는 길어지지만 문턱값 γ_t 은 신호대잡음비의 영향을 받지 않는다. 따라서, 신호대잡음비가 낮아질수록 문턱값보다 길이가 긴 가지가 많아지고, 가지들을 더 많이 버리므로 계산량은 줄어든다. 둘째, 신호대잡음비가 높아질수록 가

장 좋은 마디에서 가장 좋은 가지의 길이는 짧아지고, 가장 좋은 마디의 아이 마디가 새로운 가장 좋은 마디가 될 가능성이 높아진다. 다시 말해서, 신호대잡음비가 높아질수록, 가장 좋은 마디가 나무의 뿌리에서 아래 방향으로만 생길 확률이 높아지고, 따라서, 첫째 층에 빨리 닿아, 계산량이 줄어든다. 이 두 가지 특성 때문에 제안한 방법은 신호대잡음비가 낮거나 높을 때 계산량이 적다.

V. 맺음말

이 논문에서는 비트오류율 성능은 최적 성능에 가까우면서 계산량은 다른 준최적 복호기들보다 적은 새로운 준최적 복호기를 제안하였다. 제안한 복호기는, 길이 먼저 살펴보기와 슈노르-오허너 늘어놓기를 쓰고 첫째 층까지 예상 길이를 어림잡아 얻은 문턱값과 가지 길이를 견중으로써, 나무를 살피는 동안 가지들을 쓸데없이 많이 잊지 않고, 위층으로 되돌아가는 것을 되도록 줄인다. 제안한 복호기가 비트오류율 성능이 가장 비슷함 성능에 매우 가깝고 계산량은 다른 준최적 복호기들보다 더 적음을 모의실험으로 보였다.

참 고 문 헌

[1] 오종호, 안태훈, 송익호, 박주호, 박소령, "감쇄 환경에서 여러 입력 여러 출력 시스템에 알맞은 혼합 검파 방식," 한국통신학회 논문지, 31권, 9호, 897-904쪽, 2006년 9월.

[2] G. D. Golden, G. J. Foschini, R. A. Valenzuela, and P. W. Wolniansky, "Detection algorithm and initial laboratory results using the V-BLAST space-time communication architecture," *Electron. Lett.*, vol. 35, no. 1, pp. 14-16, Jan. 1999.

[3] E. Viterbo and J. Bouros, "A universal lattice code decoder for fading channels," *IEEE Tr. Inform. Theory*, vol. 45, no. 5, pp. 1639-1642, July 1999.

[4] 강현구, 송익호, 안태훈, 김윤희, "여러 입력 여러 출력 시스템에 알맞도록 너비를 먼저 탐색하는 가장 비슷함 복호 방식," 한국통신학회 논문지, 32권, 1호, 34-42쪽, 2007년 1월.

[5] Z. Guo and P. Nilsson, "Reduced complexity Schnorr-Euchner decoding algorithms for MIMO

systems," *IEEE Comm. Lett.*, vol. 8, no. 5, pp. 286-288, May 2004.

[6] W. Chin, "QRD based tree search data detection for MIMO communication systems," *Proc. IEEE Vehic. Techn. Conf.*, Stockholm, Sweden, vol. 3, pp. 1624-1627, May 2005.

[7] H. Kawai, K. Higuchi, N. Maeda, and M. Sawahashi, "Adaptive control of surviving symbol replica candidates in QRD-MLD for OFDM MIMO multiplexing," *IEEE J. Select. Areas Comm.*, vol. 24, no. 6, pp. 1130-1140, June 2006.

[8] K. Jeon, H. Kim, and H. Park, "An efficient QRD-M algorithm using partial decision feedback detection," *Proc. Asilomar Conf. Signals, Syst., Comp.*, Pacific Grove, CA, pp. 1658-1661, Oct. 2006.

[9] R. Gowaikar and B. Hassibi, "Statistical pruning for near-maximum likelihood decoding," *IEEE Tr. Signal Process.*, vol. 55, no. 6, pp. 2661-2675, June 2007.

[10] K. J. Kim and R. A. Iltis, "Joint detection and channel estimation algorithms for QS-CDMA signals over time-varying channels," *IEEE Tr. Comm.*, vol. 50, no. 5, pp. 845-855, May 2002.

[11] J. B. Anderson and S. Mohan, "Sequential coding algorithms: A survey and cost analysis," *IEEE Tr. Comm.*, vol. 32, no. 2, pp. 169-176, Feb. 1984.

[12] 송익호, 김광순, 박소령, 박철훈, *확률과정의 원리*, 교보문고, 2009.

[13] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, "Closest point search in lattices," *IEEE Tr. Inform. Theory*, vol. 48, no. 8, pp. 2201-2214, Aug. 2002.

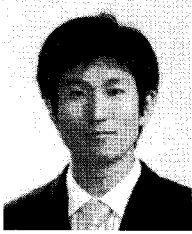
[14] J. H. Conway and N. J. A. Sloane, *Sphere Packings, Lattices, and Groups*, Springer-Verlag, 1999.

[15] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, 1996.

[16] W. Zhao and G. B. Giannakis, "Sphere decoding algorithms with improved radius search," *IEEE Tr. Comm.*, vol. 53, no. 7, pp. 1104-1109, July 2005.

안 태 훈 (Taehun An)

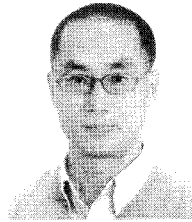
정회원



2006년 2월 성균관대학교 정보통신공학부 공학사
2008년 8월 한국과학기술원 전기및전자공학과 공학석사
2008년 9월~현재 한국과학기술원 전기및전자공학과 박사과정
<관심분야> 이동통신, 검파와 추정

송 의 호 (Iickho Song)

종신회원

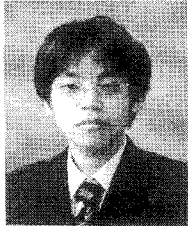


1982년 2월, 1984년 2월 서울대학교 전자공학과 공학사 (준최우등), 공학석사
1985년 8월, 1987년 5월 펜실베이니아대학교 전기공학과 공학석사, 공학박사
1987년 3월~1988년 2월 벨 통

신연구소 연구원
1988년 3월~현재 한국과학기술원 전기및전자공학과 조교수, 부교수, 교수
1995년 1월~현재 한국통신학회 논문지 편집위원, 편집 부위원장
대한전자공학회, 한국음향학회, 한국통신학회 평생회원, IET 석학회원, IEEE 석학회원
<관심분야> 통계학적 신호처리와 통신이론, 신호검파와 추정, 이동통신

강 현 구 (Hyun Gu Kang)

정회원



2004년 8월 고려대학교 전자 및 정보공학부 공학사
2006년 8월 한국과학기술원 전기및전자공학과 공학석사
2006년 9월~현재 한국과학기술원 전기및전자공학과 박사과정
<관심분야> 이동통신, 정보이

론, 검파와 추정

윤 석 호 (Seokho Yoon)

종신회원



1997년 2월 한국과학기술원 전기및전자공학과 학사 (최우등)
1999년 2월 한국과학기술원 전기및전자공학과 석사
2002년 2월 한국과학기술원 전기및전자공학과 박사
2002년 4월~2002년 6월 MIT

박사 후 연구원

2002년 7월~2003년 2월 하버드대학교 박사 후 연구원
2003년 3월~현재 성균관대학교 정보통신공학부 전임강사, 조교수, 부교수
2000년 2월 삼성 휴먼테크 논문대상 동상 받음
2007년 Marquis Who's Who in Asia에 등재
2007년 IEEE 준석학회원
2008년 Marquis Who's Who in the World에 등재
<관심분야> 이동통신, 통계학적 신호처리, 적응 신호처리

오 종 호 (Jongho Oh)

정회원



2004년 2월 한국과학기술원 전기및전자공학과 공학사
2005년 8월 한국과학기술원 전기및전자공학과 공학석사
2005년 9월~현재 한국과학기술원 전기및전자공학과 박사과정
<관심분야> 전자공학, 통신공

학, 검파와 추정