

여러 입력 여러 출력 시스템에 알맞도록 너비를 먼저 탐색하는 가장 비슷함 복호 방식

준회원 강 현 구*, 종신회원 송 익 호*, 준회원 안 태 훈*, 종신회원 김 윤 희**

A Maximum Likelihood Decoding Scheme Based on Breadth-First Searching for Multi-Input Multi-Output Systems

Hyun Gu Kang* *Associate Member*, Ickho Song* *Lifelong Member*,
Tae-hun An* *Associate Member*, Yun Hee Kim** *Lifelong Member*

요 약

여러 입력 여러 출력 시스템에서 쓰는 복호 방식인 공 복호기는 깊이를 먼저 탐색하는 가장 비슷함 복호 방식이다. 이 논문에서는 너비를 먼저 탐색하는 방법을 바탕으로, 여러 입력 여러 출력 시스템에 알맞은 새로운 가장 비슷함 복호기를 제안한다. 제안한 복호기는 가장 비슷함 검파기와 비트 오류율이 같으며 공 복호기보다 계산량이 꽤 적다는 것을 모의실험으로 보인다.

Key Words : Multi-Input Multi-Output, Maximum Likelihood Decoding, Sphere Decoder, Breadth-First Searching, Computational Burden

ABSTRACT

The sphere decoder (SD) has recently been proposed to perform maximum likelihood (ML) decoding for multi-input multi-output systems. Employing a 'breadth-first' searching algorithm for closet points in a lattice, we propose a novel ML decoding scheme for multi-input multi-output systems. Simulation results show that the proposed scheme has the same bit error rate performance as the conventional ML decoders while allowing significantly lower computational burden than the SD.

I. 서 론

무선 통신 채널에서 여러 길 감쇄가 일어날 때 보내는 쪽과 받는 쪽에 안테나를 여러 개 쓰면 여러 길 감쇄의 영향을 줄일 뿐 아니라 주파수 효율도 높일 수 있다^[1]. 이러한 여러 입력 여러 출력 (multi-input multi-output) 시스템의 보기로는, 꾸미기 쉽고

열개가 비교적 간단한 브이-블래스트 (vertical-Bell Laboratories Layered Space-Time: V-BLAST) 시스템을 들 수 있다^[2]. 브이-블래스트 시스템에서는 보내려는 데이터를 몇 개로 나누고, 이를 나눈 수만큼의 송신 안테나에서 보낸다. 이때, 이와 같이 송신 안테나를 여러 개 쓰면 보낸 신호끼리 간섭을 일으키기 때문에 효율적인 검파기를 설계하는 것이 무엇

※ 이 논문은 과학기술부의 재원으로 한국과학재단이 선정하여 지원하는 2005년 국가지정연구실 사업 과제 M10500000005-05J0000-00510의 결과 가운데 하나입니다.

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

** 경희대학교 전자정보학부 (yheekim@khu.ac.kr)

논문번호 : KICS2006-09-374, 접수일자 : 2006년 9월 5일, 최종논문접수일자 : 2007년 1월 2일

보다 중요하다.

이제까지 여러 논문에서 여러 입력 여러 출력 시스템에 알맞은 효율적인 검파기들을 연구하였다^{3, 4}. 이론적으로, 가장 비슷함 기준을 따라 신호를 검파하면 비트 오류율 (bit error rate) 성능이 가장 뛰어나다. 이때, 모든 가능성을 다 살펴보는 (full-search: 줄여서, 모가살) 복호기를 써서 신호를 가장 비슷함 검파하면 안테나나 별자리 신호수가 많은 시스템에서는 계산량이 너무 많다. 보기를 들어, 송신·수신 안테나를 4개씩 쓰고 16진 직교진폭변조하여 (quadrature amplitude modulation: QAM) 신호를 보낼 때, 모가살 복호기로 신호를 검파하려면 $4^8 \cdot (8 \times 9) = 4718592$ 번 곱셈해야 한다. 이에, 모가살 복호기보다 간단한 여러 검파기들이 제안되었다.

영 만들기 결정 되먹임 등화 (zero forcing decision feedback equalization: 줄여서, 영결되) 검파기는 모가살 복호기보다 계산량이 적지만 비트 오류율 성능이 많이 떨어진다⁵. 영결되 검파기보다 비트 오류율 성능이 좋은 검파기들이 많이 제안되었지만^{6, 7}, 이들은 여전히 비트 오류율 관점에서 최적이지 않다.

한편, 공 복호기는 (sphere decoder) 가장 비슷함 검파를 초 공간에 숨어있는 격자점들을 찾는 문제로 바꾸어 풀이를 찾는 방식이다⁸⁻¹⁰. 공 복호기는 비트 오류율이 최적이고 모가살 복호기보다 평균 계산량이 적다. 다만, 공 복호기에서는 신호대잡음비를 추정해 처음 문턱값을 얻는데, 처음 얻은 문턱값이 가장 비슷함 풀이의 유클리드 거리보다 아주 크면 계산량이 매우 많이 늘어나고, 처음 얻은 문턱값이 가장 비슷함 풀이의 유클리드 거리보다 작아 복호를 실패하면, 더 큰 문턱값을 써서 처음부터 다시 복호를 시작하거나 지음을 선언한다. 공 복호기가 지음을 선언하면 공 복호기의 비트 오류율은 최적이지 않다.

이 논문에서는, 너비를 먼저 탐색하는 방법으로¹¹ 초 공간에 숨어있는 격자점들을 찾아 가장 비슷함 풀이를 얻는 신호 복호기를 제안한다. 제안한 복호기는 신호대잡음비를 추정하지 않고, 비트 오류율이 최적이며, 공 복호기보다 계산량이 적다. 더욱이, 비트 오류율 성능에서 조금 손해를 보기로 하면, 알고리즘을 조금 고쳐 계산량을 훨씬 더 줄일 수도 있다.

II. 시스템 모형

송신기에서는 데이터를 N_r 개로 나누고 데이터마다 부호를 입힌 다음 직교진폭변조한 신호를 무선

채널로 보내며, 수신 안테나 N_r 개에서는 이들 신호의 조합을 받는다. 이때, 송신기마다 같은 수의 별자리를 써서 직교진폭변조한다고 둔다.

이제, 보낸 복소 신호 벡터 $\tilde{\mathbf{s}}$ 는 아래와 같이 나타낼 수 있다.

$$\tilde{\mathbf{s}} = [\tilde{s}_1, \tilde{s}_2, \dots, \tilde{s}_{N_r}]^T. \quad (1)$$

여기서, 윗 첨자 T 는 벡터 전치를 나타내고 벡터 $\tilde{\mathbf{s}}$ 의 j 째 원소 \tilde{s}_j 는 j 째 송신기에서 직교진폭변조한 신호를 나타낸다. 그러면, 받은 신호 벡터 $\tilde{\mathbf{a}} = [\tilde{a}_1, \tilde{a}_2, \dots, \tilde{a}_{N_r}]^T$ 를 아래와 같이 쓸 수 있다.

$$\tilde{\mathbf{a}} = \tilde{H}\tilde{\mathbf{s}} + \tilde{\mathbf{x}}. \quad (2)$$

여기서, $\tilde{\mathbf{x}} = [\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_{N_r}]^T$ 는 평균이 0이고, 분산이 σ_x^2 이며, 서로 독립이고, 분포가 같은 복소 정규 잡음 N_r 개가 이루는 벡터이고, 벡터 $\tilde{\mathbf{a}}$ 의 j 째 원소 \tilde{a}_j 는 j 째 수신 안테나에서 받은 신호이며,

$$\tilde{H} = \begin{pmatrix} \tilde{h}_{1,1} & \tilde{h}_{1,2} & \dots & \tilde{h}_{1,N_r} \\ \tilde{h}_{2,1} & \tilde{h}_{2,2} & \dots & \tilde{h}_{2,N_r} \\ \vdots & \vdots & \dots & \vdots \\ \tilde{h}_{N_r,1} & \tilde{h}_{N_r,2} & \dots & \tilde{h}_{N_r,N_r} \end{pmatrix} \quad (3)$$

은 크기가 $N_r \times N_r$ 인 채널 전달 행렬이다. 행렬 \tilde{H} 의 원소 $\tilde{h}_{j,i}$ 는 i 째 송신 안테나와 j 째 수신 안테나 사이의 복소 채널 전달 계수를 나타낸다. 이 논문에서 수신기는 \tilde{H} 의 모든 원소를 알고 있다고 둔다.

받은 신호 벡터 (2)는 복소수 풀인데, 이를 실수 풀로 쓰면 아래와 같다.

$$\mathbf{a} = H\mathbf{s} + \mathbf{x}. \quad (4)$$

여기서,

$$\mathbf{a} = [\Re\{\tilde{\mathbf{a}}^T\} \ \Im\{\tilde{\mathbf{a}}^T\}]^T = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_M]^T, \quad (5)$$

$$H = \begin{pmatrix} \Re\{\tilde{H}\} & -\Im\{\tilde{H}\} \\ \Im\{\tilde{H}\} & \Re\{\tilde{H}\} \end{pmatrix}, \quad (6)$$

$$\mathbf{s} = [\Re\{\tilde{\mathbf{s}}^T\} \ \Im\{\tilde{\mathbf{s}}^T\}]^T = [\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_{N_r}]^T, \quad (7)$$

그리고

$$\begin{aligned} \underline{\mathbf{x}} &= [\Re\{\underline{\mathbf{x}}^T\} \Im\{\underline{\mathbf{x}}^T\}]^T \\ &= [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_M]^T \end{aligned} \quad (8)$$

이다. 한편, $\Re\{\cdot\}$ 과 $\Im\{\cdot\}$ 은 각각 실수와 허수 부분을 나타내고, $M=2N_r$, $N=2N_i$, H 는 $M \times N$ 행렬이다. 식 (7)에서, N 차원 벡터 $\underline{\mathbf{s}}$ 는 신호 집합 Ω^N 의 원소인데, 보기를 들어, 4진 직교진폭변조하여 신호를 보낸다면 $\Omega = \{-1, 1\}$ 이고 16진 직교진폭변조하여 신호를 보낸다면 $\Omega = \{-3, -1, 1, 3\}$ 이다.

시스템 모형 (4)에서, $\underline{\mathbf{s}}$ 가 주어졌을 때 $\underline{\mathbf{a}}$ 의 확률 밀도함수 $f_{\underline{\mathbf{a}}|\underline{\mathbf{s}}}(\underline{\mathbf{a}}|\underline{\mathbf{s}})$ 는 아래와 같다.

$$\begin{aligned} f_{\underline{\mathbf{a}}|\underline{\mathbf{s}}}(\underline{\mathbf{a}}|\underline{\mathbf{s}}) &= \frac{1}{(\pi\sigma_x^2)^{N_r}} \\ &\cdot \prod_{j=1}^M \exp\left\{-\left(\frac{\mathbf{a}_j - \sum_{i=1}^N \mathbf{h}_{j,i} \mathbf{s}_i}{\sigma_x^2}\right)^2\right\}, \underline{\mathbf{s}} \in \Omega^N. \end{aligned} \quad (9)$$

여기서, $\mathbf{h}_{j,i}$ 는 채널 행렬 H 의 j 째 행 i 째 열의 원소이다. 한편, 가장 비슷함 기준을 써서 신호를 검파하면, (9)에서

$$\begin{aligned} \Lambda(\underline{\mathbf{s}}) &= \sum_{j=1}^M \left(\mathbf{a}_j - \sum_{i=1}^N \mathbf{h}_{j,i} \mathbf{s}_i\right)^2 \\ &= \|\underline{\mathbf{a}} - H\underline{\mathbf{s}}\|^2 \end{aligned} \quad (10)$$

을 가장 작게 하는 벡터 $\underline{\mathbf{s}}$ 를 찾아야 한다. 여기서, $\|\cdot\|^2$ 은 유클리드 거리이다. 따라서, 시스템 모형 (4)에서 최적 풀이는 아래와 같이 나타낼 수 있다.

$$\hat{\underline{\mathbf{s}}} = \underset{\underline{\mathbf{s}} \in \Omega^N}{\operatorname{argmin}} \|\underline{\mathbf{a}} - H\underline{\mathbf{s}}\|^2. \quad (11)$$

이 논문에서는 $N_i = N_r$ 이라 (곧, $M = N$ 이라) 두지만, 이 논문에서 얻은 결과들을 바탕으로 $M \geq N$ 인 때의 결과도 쉽게 얻을 수 있다.

III. 제안한 복호기

3.1 제안한 복호 알고리즘

검파기의 비트 오류를 성능을 가장 좋게 하려면 (9)-(11)에서 본 바와 같이 가장 비슷함 풀이 $\hat{\underline{\mathbf{s}}} = [\hat{\mathbf{s}}_1, \hat{\mathbf{s}}_2, \dots, \hat{\mathbf{s}}_N]^T$ 를 찾아야 한다. 이를 이제, (11)을 바탕으로 가장 비슷함 풀이 $\hat{\underline{\mathbf{s}}}$ 을 찾으려 할 때 모든

신호 벡터 $\underline{\mathbf{s}} \in \Omega^N$ 에서 (10)을 바로 계산하면 효율이 떨어진다. 이에, 효율적으로 가장 비슷함 풀이 $\hat{\underline{\mathbf{s}}}$ 을 찾으려 (10)의 모양을 바꾸면 아래와 같다.⁸⁾

$$\begin{aligned} \|\underline{\mathbf{a}} - H\underline{\mathbf{s}}\|^2 &= (\underline{\mathbf{a}} - QR\underline{\mathbf{s}})^T (\underline{\mathbf{a}} - QR\underline{\mathbf{s}}) \\ &= \|\underline{\mathbf{y}} - R\underline{\mathbf{s}}\|^2. \end{aligned} \quad (12)$$

여기서, 행렬 Q 와 R 은 H 를 QR 분해하여 얻은 것으로서, $QR = H$ 와 $Q^T Q = I$ 를 만족시킨다. 한편,

$$\begin{aligned} \underline{\mathbf{y}} &= [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_M]^T \\ &= Q^T \underline{\mathbf{a}} \\ &= R\underline{\mathbf{s}} + \underline{\mathbf{v}} \end{aligned} \quad (13)$$

이고

$$\mathbf{y}_j = \sum_{i=1}^N r_{j,i} \mathbf{s}_i + \mathbf{v}_j \quad (14)$$

이다. 식 (13)에서, M 차원 정규 잡음 벡터 $\underline{\mathbf{v}} = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_M]^T$ 의 원소들은 서로 독립이고 분포가 같다.

이제, (12)와 (13)을 바탕으로 (11)을 아래와 같이 다시 쓸 수 있다.

$$\hat{\underline{\mathbf{s}}} = \underset{\underline{\mathbf{s}} \in \Omega^N}{\operatorname{argmin}} \|\underline{\mathbf{y}} - R\underline{\mathbf{s}}\|^2. \quad (15)$$

한편, 가장 비슷함 거리 $\hat{\mathbf{d}}$ 는 아래와 같이 뜻해진다.

$$\begin{aligned} \hat{\mathbf{d}} &= \|\underline{\mathbf{y}} - R\hat{\underline{\mathbf{s}}}\|^2 \\ &= \left(\mathbf{y}_1 - \sum_{i=1}^N r_{1,i} \hat{\mathbf{s}}_i\right)^2 + \left(\mathbf{y}_2 - \sum_{i=2}^N r_{2,i} \hat{\mathbf{s}}_i\right)^2 \\ &\quad + \dots + \left(\mathbf{y}_N - r_{N,N} \hat{\mathbf{s}}_N\right)^2. \end{aligned} \quad (16)$$

여기서, $r_{j,i}$ 는 R 의 j 째 행 i 째 열의 원소이다.

3.2 복호 알고리즘

이제, 성능이 가장 좋으면서 공 복호기보다 계산량이 적은 새로운 방식을 생각해보자. 이해하기 쉽도록, 나무 열개를 써서 신호 벡터를 나타내고 이를 바탕으로 제안한 방식을 설명하고자 한다. 먼저, 뿌리에서 시작하고 N 층으로 이루어진 C 진 나무 열개를 생각하자. 여기서, C 는 집합 Ω 의 원소의 개수이다. 이 나무 열개에서 l 째 층의 k 째 마디가 나타내는 벡터를 $\underline{\mathbf{s}}_i^{(k)} = [\mathbf{s}_{1,i}^{(k)}, \mathbf{s}_{2,i}^{(k)}, \dots, \mathbf{s}_{N-i+1,i}^{(k)}]^T$ 라 하자. 여기서, $1 \leq i \leq N$ 이고 $1 \leq k \leq C^{N-i+1}$ 이다. 그러면, 신호 벡터 $\underline{\mathbf{s}} \in \Omega^N$ 은 이 나무 열개 맨 아래층의 k 째 마디에 해

당하는 벡터 $\mathbf{s}_1^{(k)}$ 로 나타낼 수 있다. 이제, $(N-l+1)$ 차원 벡터 $\mathbf{u}_l = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_{N-l+1}]^T$ 의 마디 거리 $\psi(\mathbf{u}_l)$ 을 아래와 같이 뜻매김하자.

$$\psi(\mathbf{u}_l) = \sum_{j=l}^N \left(\mathbf{y}_j - \sum_{i=j}^N \mathbf{r}_{j,i} \mathbf{u}_{-l+1,i} \right)^2. \quad (17)$$

그러면, l 째 층의 k 째 마디에 해당하는 벡터 $\mathbf{s}_l^{(k)}$ 의 마디 거리는 $\psi(\mathbf{s}_l^{(k)}) = \sum_{j=l}^N \left(\mathbf{y}_j - \sum_{i=j}^N \mathbf{r}_{j,i} \mathbf{s}_{i-l+1,i}^{(k)} \right)^2$ 이고, 이는 (16)의 오른쪽 N 항 가운데에서 가장 오른쪽부터 $N-l+1$ 개 더한 꼴이다.

식 (14)를 살펴보면, \mathbf{y}_N 을 이루는 신호 성분은 신호들 $\{\mathbf{s}_j\}_{j=1}^N$ 가운데 \mathbf{s}_N 뿐이다. 곧, 다른 신호들의 간섭이 없는 \mathbf{y}_N 을 써서 \mathbf{s}_N 을 먼저 복호하는 것이 가장 바람직하다. 이제, N 째 층의 모든 마디 거리를 계산한 다음 d_N 보다 마디 거리가 큰 마디를 모두 버린다. 여기서, d_N 은 $\hat{d} \leq d_N$ 을 만족시키는 수이다. 버린 마디들과 그 마디들에서 뺀 마디들은 더 이상 생각하지 않는다. 다음에, N 째 층에서 살아남은 (버리지 않은) 마디들에서 뺀 $(N-1)$ 째 층의 모든 마디의 마디 거리를 계산하고, d_{N-1} 보다 마디 거리가 큰 마디를 모두 버린다. 여기서, d_{N-1} 은 $\hat{d} \leq d_{N-1} \leq d_N$ 을 만족시키는 수이다. 이런 과정을 되풀이하여 맨 아래층에서 (첫째 층) 마디 거리가 가장 작은 마디를 찾으면 복호 과정이 끝난다. 맨 아래층에서 마디 거리가 가장 작은 마디가 나타내는 벡터가 바로 가장 비슷함 풀이 $\hat{\mathbf{s}}$ 이다.

여기서, $\{d_i\}_{i=N}^2$ 가 작을수록 마디를 더 많이 버릴 수 있으며, 이는 계산량 이득이 더 큼을 뜻한다. 한편, 제안한 방식에서는 $\{d_i\}_{i=N}^2$ 를 얻을 때 QR 분해를 쓰는 잘 알려진 곱파방법들 가운데 계산량이 가장 적은 결정 되먹임 등화 방법을 쓴다. 결정 되먹임 등화 풀이와 결정 되먹임 등화 거리를 각각 $\tilde{\mathbf{s}} = [\tilde{\mathbf{s}}_1, \tilde{\mathbf{s}}_2, \dots, \tilde{\mathbf{s}}_N]^T$ 와 $\tilde{d} = \|\mathbf{y} - R\tilde{\mathbf{s}}\|^2$ 이라 두고, $(l+1)$ 째 층에서 살아남은 마디들에서 뺀 l 째 층의 마디들 가운데 마디 거리가 k 째로 작은 마디에 해당하는 벡터를 $\mathbf{s}_l^{(k)} = [\mathbf{s}_{1,l}^{(k)}, \mathbf{s}_{2,l}^{(k)}, \dots, \mathbf{s}_{N-l+1,l}^{(k)}]^T$ 라 두자. 그러면, $\{d_i\}_{i=N}^2$ 는 아래와 같이 얻을 수 있다.

$$d_i = \min\{d_{i+1}, \|\mathbf{y} - R\mathbf{g}^{(i)}\|^2\}, \quad i = N, N-1, \dots, 2. \quad (18)$$

여기서, $d_{N+1} = \tilde{d}$ 이고 $\mathbf{g}^{(i)} \in \Omega^N$ 은 $\tilde{\mathbf{s}}$ 와 $\mathbf{s}_i^{(1)}$ 을 나란히 놓아 만든 것으로 아래와 같다.

$$\mathbf{g}^{(i)} = [\tilde{\mathbf{s}}_1, \tilde{\mathbf{s}}_2, \dots, \tilde{\mathbf{s}}_{i-1}, \mathbf{s}_{1,i}^{(1)}, \mathbf{s}_{2,i}^{(1)}, \dots, \mathbf{s}_{N-l+1,i}^{(1)}]^T. \quad (19)$$

여기서, \tilde{d} 와 d_i 는 가장 비슷함 거리 \hat{d} 보다 언제나 크거나 같으므로, $\{d_i\}_{i=N}^2$ 는 아래를 만족시킨다는 것을 눈여겨 볼만하다.

$$\hat{d} \leq d_2 \leq d_3 \leq \dots \leq d_N \leq \tilde{d}. \quad (20)$$

제안한 방식에서는 복호 실패가 일어나지 않으며, 버린 마디에는 결코 가장 비슷함 풀이가 들어있지 않음을 (20)에서 알 수 있다.

한편, (18)에서 $\{\|\mathbf{y} - R\mathbf{g}^{(i)}\|^2\}_{i=N}^2$ 를 계산하는 데 드는 계산량도 꽤 많다. 이에, 검사 벡터 $\mathbf{t} = [t_1, t_2, \dots, t_N]^T$ 를 써서 $\{\|\mathbf{y} - R\mathbf{g}^{(i)}\|^2\}_{i=N}^2$ 의 계산량을 줄인다. 처음에는 검사 벡터 \mathbf{t} 를 $\tilde{\mathbf{s}}$ 으로 두고, $d_{i+1} > \|\mathbf{y} - R\mathbf{g}^{(i)}\|^2$ 일 때에는 \mathbf{t} 를 $\mathbf{g}^{(i)}$ 로 바꾼다. 구체적으로는, (18)의 $\|\mathbf{y} - R\mathbf{g}^{(i)}\|^2$ 을 계산하기에 앞서서 $[t_i, t_{i+1}, \dots, t_N]^T$ 와 $\mathbf{s}_i^{(1)}$ 을 견주어 보고 $[t_i, t_{i+1}, \dots, t_N]^T$ 와 $\mathbf{s}_i^{(1)}$ 이 같으면 $d_i = d_{i+1} = \|\mathbf{y} - R\mathbf{g}_i^{(1)}\|^2$ 이므로 $\|\mathbf{y} - R\mathbf{g}^{(i)}\|^2$ 을 계산하지 않는다. 이는 $\{\|\mathbf{y} - R\mathbf{g}^{(i)}\|^2\}_{i=N}^2$ 을 계산하는 데 드는 연산을 줄일 수 있음을 뜻한다. 한편, $[t_i, t_{i+1}, \dots, t_N]^T$ 와 $\mathbf{s}_i^{(1)}$ 이 다르면 $\|\mathbf{y} - R\mathbf{g}^{(i)}\|^2$ 을 계산한 다음 (18)을 써서 d_i 을 얻는다. 표 1은 제안한 방식의 알고리즘을 정리한 것이다.

표 1에 보인 알고리즘에서, 둘째 단계의 2-2와 2-3을 좀 더 자세히 설명하면 다음과 같다. 앞에서 말한 대로 살아남은 마디들만 생각하여, l 째 층에서 살아남은 마디 수를 E_l 이라 하고, $E_{N+1} = 1$ 이라 두자. 한편, 집합 $\{\{\mathbf{s}_{1,N}^{(j)}, \mathbf{s}_{1+1,N}^{(j)}, \dots, \mathbf{s}_{N-l+1,N}^{(j)}\}^T : j=1, 2, \dots, C, p=1, 2, \dots, E_{l+1}\}$ 은 집합 $\{\mathbf{s}_i^{(k)}\}_{k=1}^{C E_{l+1}}$ 과 같다. 보기를 들어, $M=N=4$, $\Omega = \{-1, 1\}$, $E_3 = 2$, $\mathbf{s}_3^{(1)} = [1, -1]^T$, 그리고 $\mathbf{s}_3^{(2)} = [1, 1]^T$ 일 때, 집합 $\{\{\mathbf{s}_{1,4}^{(1)}, \mathbf{s}_{1,3}^{(1)}, \mathbf{s}_{2,3}^{(1)}\}^T, \{\mathbf{s}_{1,4}^{(2)}, \mathbf{s}_{1,3}^{(2)}, \mathbf{s}_{2,3}^{(2)}\}^T\} = \{[-1, 1, -1]^T, [-1, 1, 1]^T, [1, 1, -1]^T, [1, 1, 1]^T\}$ 는 순서만 조금 다를 뿐 집합 $\{\mathbf{s}_2^{(k)}\}_{k=1}^4 = \{[-1, 1, -1]^T, [1, 1, -1]^T, [-1, 1, 1]^T, [1, 1, 1]^T\}$ 와 같다. 그러면, $j=1, 2, \dots, C$ 이고 $p=1, 2, \dots, E_{l+1}$ 일 때

$$\begin{aligned} \Psi(\{\mathbf{s}_{1,N}^{(j)}, \mathbf{s}_{1,i+1}^{[p]}, \dots, \mathbf{s}_{N-i,i+1}^{[p]}\}^T) \\ = \left(\mathbf{y}_i - \mathbf{r}_{i,i} \mathbf{s}_{1,N}^{(j)} - \sum_{\xi=i+1}^N \mathbf{r}_{i,\xi} \mathbf{s}_{\xi-i,i+1}^{[p]} \right)^2 \\ + \sum_{j=i+1}^N \left(\mathbf{y}_j - \sum_{\xi=j}^N \mathbf{r}_{j,\xi} \mathbf{s}_{\xi-i,i+1}^{[p]} \right)^2 \\ = \left(\mathbf{y}_i - \mathbf{r}_{i,i} \mathbf{s}_{1,N}^{(j)} - \theta(\mathbf{s}_{i+1}^{[p]}) \right)^2 + \Psi(\mathbf{s}_{i+1}^{[p]}) \end{aligned} \quad (21)$$

을 계산하면 i 제 층의 마디 거리 $\{\Psi(\mathbf{s}_i^{(k)})\}_{k=1}^{CE_i}$ 을 얻을 수 있다. 여기서, $\theta(\mathbf{s}_{N+1}^{[1]})=0$, $\Psi(\mathbf{s}_{N+1}^{[1]})=0$, 그리고

$$\theta(\mathbf{u}_{i+1}) = \sum_{\xi=1}^{N-i} \mathbf{r}_{i,i+\xi} \mathbf{u}_{\xi} \quad (22)$$

이다. 식 (21)에서, $\Psi(\mathbf{s}_{i+1}^{[p]})$ 는 이미 $(i+1)$ 제 층에서 얻었으므로 i 제 층에서 마디 거리를 얻으려면 $\mathbf{y}_i - \mathbf{r}_{i,i} \mathbf{s}_{1,N}^{(j)}$ 와 $\theta(\mathbf{s}_{i+1}^{[p]})$ 만 계산하면 된다.

한편, 계산량을 더욱 줄이고자 $\theta(\mathbf{s}_{i+1}^{[p]})$ 를 다음과 같이 계산한다. 먼저, (22)를 다시 쓰면 아래와 같다.

$$\begin{aligned} \theta(\mathbf{s}_{i+1}^{[p]}) &= \theta(\mathbf{s}_{i+1}^{[p-1]}) + \theta(\mathbf{s}_{i+1}^{[p]}) - \theta(\mathbf{s}_{i+1}^{[p-1]}) \\ &= \theta(\mathbf{s}_{i+1}^{[p-1]}) + \sum_{\xi=1}^{N-i} \mathbf{r}_{i,i+\xi} (\mathbf{s}_{\xi,i+1}^{[p]} - \mathbf{s}_{\xi,i+1}^{[p-1]}) \end{aligned} \quad (23)$$

여기서, $\theta(\mathbf{s}_{i+1}^{[0]})=0$ 이고 $\mathbf{s}_{i,j}^{[0]}=0$ 이라 둔다. 이제, $p=1$ 일 때부터 $\theta(\mathbf{s}_{i+1}^{[p]})$ 를 얻는데, (23)의 둘째 항을 계산하기에 앞서서 $\mathbf{s}_{i+1}^{[p]}$ 와 $\mathbf{s}_{i+1}^{[p-1]}$ 을 견주어 본다. $\mathbf{s}_{i,i+1}^{[p]} = \mathbf{s}_{i,i+1}^{[p-1]}$ 이면 $\mathbf{r}_{i,i+\xi} (\mathbf{s}_{\xi,i+1}^{[p]} - \mathbf{s}_{\xi,i+1}^{[p-1]})=0$ 이므로 $\mathbf{r}_{i,i+\xi} (\mathbf{s}_{\xi,i+1}^{[p]} - \mathbf{s}_{\xi,i+1}^{[p-1]})$ 을 계산하지 않는다. 곧, $\mathbf{s}_{i,i+1}^{[p]}$ 와 $\mathbf{s}_{i,i+1}^{[p-1]}$ 가 다를 때에만 $\mathbf{r}_{i,i+\xi} (\mathbf{s}_{\xi,i+1}^{[p]} - \mathbf{s}_{\xi,i+1}^{[p-1]})$ 을 계산하여 계산량을 줄일 수 있다. 식 (21)은 공 복호기에서도 쓰지만 (23)을 써서 계산량을 줄이는 방법은 제안한 방식에서만 쓸 수 있음을 새겨두자: 공 복호기에서는 깊이를 먼저 탐색하기 때문에 (23)을 써서 계산량을 줄이기에는 매우 까다롭다. 그림 1은 $C=3$ 이고 $N=4$ 일 때, 제안한 방식을 쓴 복호 과정을 보여준다.

제안한 방식에서는 마디마다 마디 거리를 많아야 한번 계산하고, 마디 거리를 계산할 때에도 이웃 벡터를 (보기를 들면, (23)에서 벡터 $\mathbf{s}_{i+1}^{[p]}$ 과 짝을 이루는 벡터 $\mathbf{s}_{i+1}^{[p-1]}$) 써서 계산량을 줄인다. 한편, 공 복호기에서는 깊이를 먼저 탐색하므로 한 마디를 여러 번 탐색할 때도 있고, 따라서, 살아남은 마디마다 마디 거리를 적어도 한번 계산한다. 공 복호기에서도 마디마다 마디 거리를 한번만 계산할 수 있으나, 그

표 1. 제안한 방식의 알고리즘

<p>첫째 단계 입력 (\mathbf{a}, H, Ω).</p> <p>1-1. 행렬 H를 QR 분해한 뒤 $\mathbf{y} = Q^T \mathbf{a}$를 계산한다. 1-2. 결정 되먹임 등화 방법을 써서 $\check{\mathbf{s}}$을 얻고 $\check{\mathbf{d}} = \ \mathbf{y} - R\check{\mathbf{s}}\ ^2$을 계산한다.</p> <p>출력 ($\check{\mathbf{d}}$, $\check{\mathbf{s}}$, \mathbf{y}, R).</p> <p>둘째 단계 입력 ($\check{\mathbf{d}}$, $\check{\mathbf{s}}$, \mathbf{y}, R, Ω).</p> <p>2-1. 먼저, $i = N$; $\mathbf{d}_{i+1} = \check{\mathbf{d}}$, 그리고 $\mathbf{t} = \check{\mathbf{s}}$으로 둔다. 2-2. N제 층에서 모든 마디들의 마디 거리를 계산하고 $\mathbf{s}_N^{[1]}$을 얻은 다음 2-4로 간다. 2-3. $(i+1)$제 층에서 살아남은 마디들에서 뽑은 i제 층의 모든 마디들의 마디 거리를 계산하고 $\mathbf{s}_i^{[1]}$을 얻는다. $i \neq 1$이면 2-4로 가고, $i = 1$이면 2-7로 간다. 2-4. $[\mathbf{t}_i, \mathbf{t}_{i+1}, \dots, \mathbf{t}_N]^T = \mathbf{s}_i^{[1]}$이면, $\mathbf{d}_i = \mathbf{d}_{i+1}$로 두고 2-6으로 간다. $[\mathbf{t}_i, \mathbf{t}_{i+1}, \dots, \mathbf{t}_N]^T \neq \mathbf{s}_i^{[1]}$이면, 2-5로 간다. 2-5. $\ \mathbf{y} - R\mathbf{q}^{(i)}\ ^2$을 계산한다. $\ \mathbf{y} - R\mathbf{q}^{(i)}\ ^2 \geq \mathbf{d}_{i+1}$이면, $\mathbf{d}_i = \mathbf{d}_{i+1}$로 두고 2-6으로 간다. $\ \mathbf{y} - R\mathbf{q}^{(i)}\ ^2 < \mathbf{d}_{i+1}$이면, $\mathbf{d}_i = \ \mathbf{y} - R\mathbf{q}^{(i)}\ ^2$ 그리고 $\mathbf{t} = \mathbf{q}^{(i)}$로 두고 2-6으로 간다. 2-6. 마디 거리가 \mathbf{d}_i보다 큰 마디들을 모두 버리고, i를 하나 줄인 다음 2-3으로 간다. 2-7. 복호를 끝낸다.</p> <p>출력 ($\hat{\mathbf{s}}$)</p>

렇게 하려면 계산했던 모든 마디 거리를 복호가 끝날 때까지 저장해 두어야 한다.

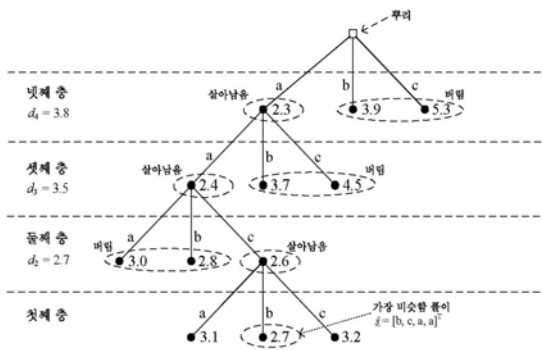


그림 1. 제안한 방식의 복호 과정 ($C=3$, $N=2N_k=4$)

IV. 성능 평가

4.1 계산량 분석

곱셈 횟수는 복호기의 계산량을 견주는 데 많이 쓰인다. 이제, 제안한 방식과 다른 복호기들이 가장 비슷함 풀이를 찾는 데 (복호하는 데) 드는 곱셈 횟수를 견주어 보자.

표 1에 보인 알고리즘의 첫째 단계에서 H 를 QR 분해하는 데 $MN^2 - N^3/3$ 번 flop ^[12], $Q^T \mathbf{a}$ 를 계산하는 데 M^2 번, $\check{\mathbf{a}}$ 을 얻는 데 $N(N+3)/2$ 번 곱셈하고, $\check{\mathbf{s}}$ 을 얻는 데 N 번 나눗셈 한다. 곱셈과 나눗셈의 계산량이 같다고 두면, 첫째 단계에서는 아래만큼 곱셈한다.

$$MN^2 - \frac{N^3}{3} + M^2 + \frac{N(N+3)}{2}. \quad (24)$$

이제, 둘째 단계에서 $\{\theta(\mathbf{g}_{i+1}^{[k]})\}_{k=1}^{E_i}$ 과 \mathbf{d}_i 을 계산하는 데 드는 곱셈 횟수를 각각 U_i 과 D_i 로 두면, $U_N = 0$ 이고 $D_1 = 0$ 이며, i 째 층에서 계산을 하는 데 드는 곱셈 횟수를 아래와 같이 쓸 수 있다.

$$\frac{C}{2} + CE_{i+1} + U_i + D_i \quad (25)$$

따라서, 둘째 단계에서는 아래만큼 곱셈한다.

$$\sum_{i=1}^N \left(\frac{C}{2} + CE_{i+1} + U_i + D_i \right). \quad (26)$$

식 (26)에서, $\{E_i\}_{i=1}^N$, $\{U_i\}_{i=1}^N$, 그리고 $\{D_i\}_{i=2}^N$ 은 신호대잡음비 (signal to noise ratio: SNR), 신호 별자리 크기, 그리고 송신/수신 안테나 수의 영향을 받으며 아래를 만족시킨다.

$$1 \leq E_i \leq C^{N-i+1}, \quad (27)$$

$$N-i \leq U_i \leq (N-i)C^{N-i}, \quad (28)$$

그리고

$$0 \leq D_i \leq N + \sum_{i=1}^N i. \quad (29)$$

식 (26)-(29)을 바탕으로 둘째 단계에서 가장 많이 계산할 때와 가장 적게 계산할 때 드는 곱셈 횟수를 얻으면 각각 아래와 같다.

$$\frac{C^N \{C^2 + C(N-2) - N\} - C^2 + 2C}{(C-1)^2} + \frac{N(2N^2 + 9N + 3C + 1)}{6}, \quad (30)$$

$$\frac{N(N+3C-1)}{2}. \quad (31)$$

곧, 제안한 방식은 가장 비슷함 풀이를 찾을 때, 많으면 (24)와 (30)을 더한 만큼, 적으면 (24)와 (31)를 더한 만큼 곱셈한다.

4.2 모의실험

이제, 몬테카를로 방법으로 10^6 번 모의실험하여 얻은 평균 곱셈 횟수를 바탕으로 제안한 방식과 공 복호기의 계산량을 견주어 보자. 또한, 제안한 방식, 모가살 복호기, 그리고 공 복호기의 비트 오류율 성능도 견주어 보자. 모의실험에서, 신호대잡음비는 (전송신호전력)/ σ_s^2 으로 두었으며, 모든 결과는 10^6 번 거듭 실험하여 얻었다. 한편, 공 복호기에서, 처음 얻은 문턱값이 가장 비슷함 거리보다 작아 복호할 수 없을 때에는 문턱값을 두 배로 하여 다시 복호를 시작하였다.

그림 2는 제안한 방식과 공 복호기가 복호 과정을 10^6 번 되풀이하는 동안 얻은 곱셈 횟수들을 오름차순으로 세운 것이다. 이 그림에서 점 (x, y) 는 x 째로 적은 곱셈 횟수가 y 임을 나타내며, 결정-공 복호기는 공 복호기의 처음 문턱값을 결정 되먹임 등화 방법을 써서 얻었음을 나타낸다. 이 그림에서 제안한 방식은 결정-공 복호기보다 더 적게 계산할 것임을 알 수 있다.

그림 3은 제안한 방식과 결정-공 복호기의 곱셈 횟수를 가장 나쁠 때와 가장 좋을 때로 나누어 보여 준다. 결정-공 복호기에서 이론적으로 가장 나쁠 때의 곱셈 횟수를 얻기는 매우 어렵지만, 가장 좋을 때의 곱셈 횟수는 이론적으로 $MN^2 - N^3/3 + M^2 + N(3N+35)/2$ 쯤 이다. 이론과 모의실험에서 얻은 가장 좋을 때의 곱셈 횟수와 가장 나쁠 때의 곱셈 횟수 모두 제안한 방식이 결정-공 복호기보다 적다.

그림 4는 제안한 방식, 결정-공 복호기, 그리고 추정-공 복호기의 평균 곱셈 횟수를 보여 준다. 여기서, 추정-공 복호기는 신호대잡음비를 추정하여 처음 문턱값을 얻은 공 복호기를 뜻한다. 이때, 수신기가 신호대잡음비를 제대로 추정하였다고 두었고, 가장 비슷함 거리가 문턱값보다 작을 확률이 99.9%가 되도록

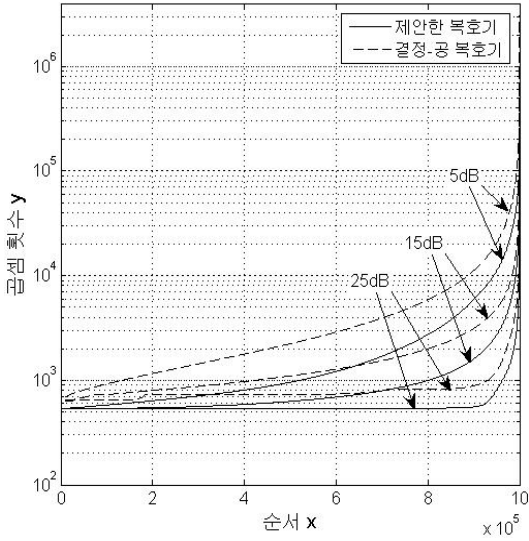


그림 2. 제안한 방식과 결정-공 복호기의 곱셈 횟수 분포 ($N_t = N_r = 4$, 16진 직교진폭변조)

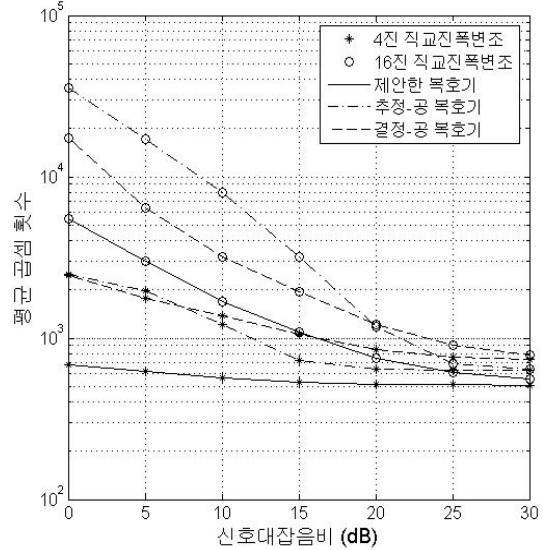


그림 4. 제안한 방식과 공 복호기의 평균 곱셈 횟수 ($N_t = N_r = 4$)

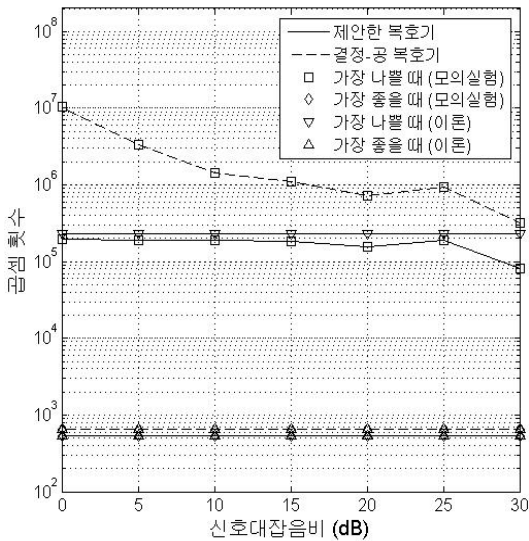


그림 3. 제안한 방식과 결정-공 복호기의 이론적인 곱셈 횟수 모의실험에서 얻은 곱셈 횟수 ($N_t = N_r = 4$, 16진 직교진폭변조)

록 처음 문턱값을 잡았다. 이 그림에서 제안한 방식은 공 복호기보다 평균적으로 적게 계산함을 알 수 있다. 게다가, 추정-공 복호기에서 신호대잡음비를 추정하는 데 드는 곱셈 횟수는 생각하지 않았음에도, 제안한 방식의 평균 곱셈 횟수가 추정-공 복호기의 평균 곱셈 횟수의 16%~86%밖에 되지 않음을 새겨 두자. 한편, 신호대잡음비가 낮아질수록 제안한 방식과 공 복호기의 평균 곱셈 횟수 차가 점점 더 커진

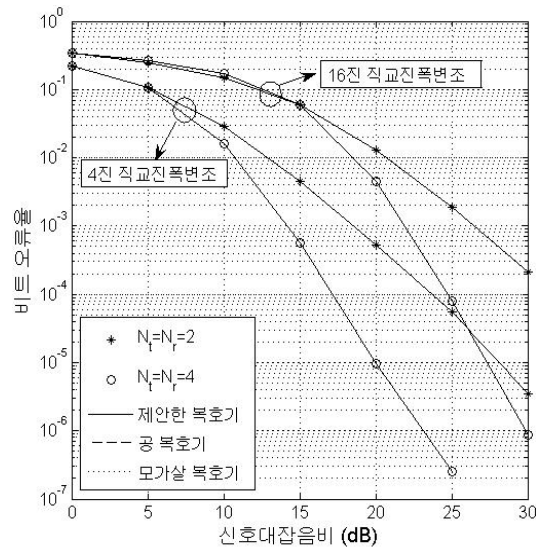


그림 5. 제안한 방식, 모가살 복호기, 공 복호기의 성능 특성

다. 그림 5는 제안한 방식, 모가살 복호기, 그리고 공 복호기의 비트 오류율 성능을 보여 준다. 복호기 모두 가장 비슷함 기준을 따라 풀이를 찾으므로, 비트 오류율 성능이 같음을 알 수 있다.

V. 맺음말

이 논문에서는 여러 입력 여러 출력 통신 시스템에서 비트 오류율은 최적이고 공 복호기보다 계산량

이 적은 새로운 복호기를 제안하였다. 제안한 방식은 너비를 먼저 탐색하여 얻는 정보를 바탕으로 복호하는 데 드는 계산량을 줄였을 뿐만 아니라 신호대잡음비도 추정하지 않는다. 모의실험으로 제안한 방식은 비트 오류율 성능이 최적이고 공 복호기보다 더 적게 계산함을 보였다. 아울러, 신호대잡음비가 낮아질수록 제안한 방식의 계산량 이득이 더 높아짐을 보였다.

참 고 문 헌

[1] G. J. Foschini and M. J. Gans, "On limits of wireless communications in a fading environment when using multiple antennas," *Wireless Pers. Comm.*, vol. 6, no. 3, pp. 311-335, Mar. 1998.

[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. 14-15, Jan. 1999.

[3] G. J. Foschini, G. D. Golden, R. A. Valenzuela, and P. W. Wolniansky, "Simplified processing for high spectral efficiency wireless communication employing multi-element arrays," *IEEE J. Sel. Area. Comm.*, vol. 17, no. 11, pp. 1841-1852, Nov. 1999.

[4] 황현철, 신승훈, 이철진, 광경섭, "시공간블록부호화를 적용한 공간다중화 시스템 수신기: 복잡도 감소 방안," *한국통신학회 논문지*, 29권, 11호, 1244-1252쪽, 2004년 11월.

[5] C. Shen, H. Zhuang, L. Dai, and S. Zhou, "Detection algorithm improving V-BLAST performance over error propagation," *Electron. Lett.*, vol. 39, no. 13, pp. 1007-1008, June 2003.

[6] S. W. Kim and K. P. Kim, "Log-likelihood-ratio-based detection ordering in V-BLAST," *IEEE Tr. Comm.*, vol. 54, no. 2, pp. 302-307, Feb. 2006.

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

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

[9] O. Damen, A. Chkeif, and J. C. Belfiore, "Lattice code decoder for space-time codes," *IEEE Comm.*

Lett., vol. 4, no. 5, pp. 161-163, May 2000.

[10] B. Hochwald and S. ten Brink, "Achieving near capacity on a multiple antenna channel," *IEEE Tr. Comm.*, vol. 51, no. 3, pp. 389-399, Mar. 2003.

[11] D. W. Patterson, *Introduction to Artificial Intelligence and Expert Systems*, Prentice Hall, 1990.

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

강 현 구 (Hyun Gu Kang)

준회원



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

검파와 추정

송 익 호 (Ickho Song)

중신회원



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

연구소 연구원

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

안 태 훈 (Taehun An)

준회원



2006년 2월 성균관대학교 정보통신공학부 공학사
2006년 3월~현재 한국과학기술원 전자전산학과 석사과정
<관심분야> 이동통신, 검파와 추정

김 윤 희 (Yun Hee Kim)

중신회원



1995년 2월 한국과학기술원 전
기및전자공학과 공학사

1997년 2월 한국과학기술원 전
기및전자공학과 공학석사

2000년 1월~2000년 4월 UCSD
방문 연구원

2000년 8월 한국과학기술원 전

자전산학과 박사

2000년 9월~2004년 8월 한국전자통신연구원 선임연
구원

2004년 9월~현재 경희대학교 전자정보학부 조교수

2006년 1월~현재 한국통신학회 논문지 편집위원

IEEE 준석학회원

<관심분야> 이동/무선 통신, 통계학적 신호처리와 통신
이론, 부호화 변조, 적응 전송