

Gradient Guided 탐색을 이용한 고속 CDMA 다중사용자 검출

Fast Multiuser Detection in CDMA Systems Using Gradient Guided Search

최 양 호*
Choi, Yang-Ho

Abstract

We present a fast algorithm for CDMA (code division multiple access) multiuser detection using the gradient guided search. The fast algorithm calculates the maximum likelihood (ML) metric so efficiently that it needs only $O(K)$ additions in the presence of K users once some initialization is completed. The computational advantages of the fast algorithm over the conventional method are more noticeable as more iterations are required to obtain a suboptimal solution as in the initialization with matched filters.

키워드 : 다중사용자 검출, CDMA, maximum likelihood
Keywords : multiuser detection, CDMA, maximum likelihood

1. 서론

지난 10여 년 동안 이동통신서비스는 예상을 훨씬 뛰어넘는 폭발적인 확산을 거듭하였다. 이와 함께 이동통신관련 기술로 급속도로 발전하였다. 특히 CDMA(code division multiple access) 기술은 시스템 용량, 페이딩에 대한 대응력, 유연성, 비화 등에서의 우수성으로 높은 관심을 불러왔다. 이와 같은 우수성과 함께 영상 등 무선멀티미디어 서비스 제공에 효율적이어서 CDMA 방식은 제3세대 이동통신시스템인 IMT-2000에서 핵심 다원접속 방식으로 자리 매김을 하고 있다[1-3].

다중사용자 검출(multiuser detection)[4-6]은 기본적으로 타 사용자로부터의 간섭을 배제할 수 있어, 시스템 용량을 획기적으로 증가 시킬 수 있고,

CDMA 사용자의 원근(near-far)에 따라 정밀한 전력제어를 해야 하는 까다로운 문제를 피할 수 있다. 관찰된 샘플이 주어졌을 때 조건부 확률을 최대로 하는 ML(maximum likelihood) 원리에 기초하여 다중사용자 검출을 할 수 있다[4]. 이 방식은 최적 성능을 얻을 수 있지만 사용자 수에 따라 계산량이 지수적으로 증가하여 사용자가 적은 경우 외에는 이 방법이 적용이 곤란하다. 최근에 이러한 계산상의 복잡함을 극복하기 위해 이산 공간(discrete space)에서 gradient 탐색에 기초한 방법이 소개되었다[5]. 이 방법에서는 특정 반경을 갖는 Hamming sphere 내에서 ML metric 기준에 따라 가장 좋은 벡터를 찾고, 찾은 벡터를 중심으로 하는 Hamming sphere를 다시 정의하여 벡터를 찾는다. 이와 같은 반복계산(iteration)을 통해 준 최적 해(suboptimal solution)를 구한다.

본 논문에서는 gradient 탐색을 이용한 효율적인 다중사용자 알고리즘을 제안한다. 제안된 알고리즘은 기존의 방법보다 ML metric 계산량을 크

* 강원대학교 전기전자정보통신공학부 부교수, 공학박사

게 줄일 수 있다. Gradient 탐색에서는 초기 추정
이 필요하다. 상대적으로 정확성은 떨어지지만 구
현이 간단한 정합필터(matched filter)에 의한 추정
치를 이용해도, decorrelator를 이용하여 복잡하지
만 정확성이 좋은 추정치를 얻은 경우와 BER(bit
error rate) 성능이 동일함을 시뮬레이션 결과는 보
여준다. 정확성이 떨어지는 추정치를 이용하면 더
많은 반복계산을 해야 하지만 제한한 알고리즘에
서 이에 따른 계산량의 증가는 미미하다.

2. 시스템 모델

K 명의 사용자가 이용하고 있는 CDMA 시스템
을 생각한다. 송신 측에서 k 번째 사용자의 데이터
심벌 $b_k(t)$ 는 2진 의사잡음(pseudo-noise) 시퀀스
(binary pseudo-noise sequence)에 의해 확산된다.
 k 번째 사용자의 j 번째 데이터 비트에 대한 확산 시
퀀스(spreading sequence)는

$$s_k^{(j)}(t) = \sum_{n=0}^{N-1} c_k^{(j)}(n) p_{T_c}(t - nT_c) \quad (1)$$

와 같이 표현된다. 여기서 N 은 처리이득
(processing gain), $c_k^{(j)}(n) \in \{-1, 1\}$ 은 k 번째
사용자의 j 번째 데이터 비트에 대한 n 번째 칩, T_c
는 칩의 주기, $p_T(t)$ 는 길이가 T 크기가 1인 구형
파 펄스를 나타낸다. k 번째 사용자의 j 번째 데이터
비트에 대한 채널의 임펄스 응답 $h_k^{(j)}(t)$ 는

$$h_k^{(j)}(t) = \beta_k^{(j)} \delta(t - \tau_k) \quad (2)$$

와 같이 쓸 수 있다. 여기서 $\beta_k^{(j)}$, τ_k 은 각각 k 번
째 사용자의 j 번째 데이터 비트에 대한 채널의 이
득, 경로지연이다. 경로지연은 전체 관찰 구간동안
일정하다고 가정하였다.

수신된 기저대역 등가 신호(received equivalent
baseband signal) $r(t)$ 는 다음처럼 나타낼 수 있
다.

$$\begin{aligned} r(t) &= \sum_{j=-\infty}^{\infty} \sum_{k=1}^K a_k b_k^{(j)} s_k^{(j)}(t) * h_k^{(j)}(t) + n(t) \\ &= \sum_{j=-\infty}^{\infty} \sum_{k=1}^K \sum_{n=1}^{N-1} [a_k \beta_k^{(j)} b_k^{(j)} c_k^{(j)}(n) \\ &\quad p_{T_c}(t - nT_c - \tau_k)] + n(t) \end{aligned} \quad (3)$$

여기서 a_k 는 k 번째 사용자의 진폭,
 $b_k^{(j)} \in \{-1, 1\}$ 는 j 번째 데이터 비트이고 T_s 는
데이터 심벌의 주기이다. $n(t)$ 는 백색 가우시안
잡음(white Gaussian noise)으로 전력 밀도가
 $N_0/2$ 이다.

$\tau_1 (=0) \leq \tau_1 \leq \dots \tau_K \leq T_s$ 와 같이 가정한다.
이와 같이 가정해도 일반성은 유지된다. k 번째 사
용자의 j 번째 데이터 심벌 구간에서 정합필터 표본
화 출력은 아래와 같이 주어진다.

$$y_k^{(j)} = \frac{1}{T_s} \int_{jT_s + \tau_k}^{(j+1)T_s + \tau_k} s_k^{(j)}(t - \tau_k) r(t) dt \quad (4)$$

상관행렬 (correlation matrix)을 다음처럼 정의한
다.

$$\begin{aligned} [R^{(j)}(l)]_{k_1, k_2} &= \frac{1}{T_s} \int_{-\infty}^{\infty} s_{k_1}^{(j)}(t - \tau_{k_1}) s_{k_2}^{(j+l)}(t - lT_s - \tau_{k_2}) dt \end{aligned} \quad (5)$$

여기서 $[R^{(j)}(l)]_{k_1, k_2}$ 는 $K \times K$ 상관행렬

$R^{(j)}(l)$ 의 k_1 번째 행, k_2 요소, 즉 $(k_1,$
 $k_2)$ 요소를 나타낸다. 식 (3) - (5)를 이용하여,
 K 사용자 모두에 대한 정합필터 출력을 벡터 형태
로 표현하면

$$\begin{aligned} \mathbf{y}^{(j)} &= [y_1^{(j)}, \dots, y_K^{(j)}]^T \\ &= R^{(j)}(-1) \mathbf{A}^{(j-1)} \mathbf{b}^{(j-1)} \\ &\quad + R^{(j)}(0) \mathbf{A}^{(j)} \mathbf{b}^{(j)} \\ &\quad + R^{(j)}(1) \mathbf{A}^{(j+1)} \mathbf{b}^{(j+1)} + \mathbf{n}^{(j)} \end{aligned} \quad (6)$$

와 같고, 위첨자 T 는 전치(transpose)를 나타내며,

$$\mathbf{A}^{(j)} = \text{diag}(A_1^{(j)}, A_2^{(j)}, \dots, A_K^{(j)}) \quad (7)$$

$$A_k^{(j)} = \frac{\alpha_k}{\beta_k^{(j)}} \quad (8)$$

$$\mathbf{b}^{(j)} = [b_1^{(j)}, \dots, b_K^{(j)}]^T \quad (9)$$

$$\mathbf{n}^{(j)} = [n_1^{(j)}, \dots, n_K^{(j)}]^T \quad (10)$$

$$n_k^{(j)} = \frac{1}{T_s} \int_{jT_s + \tau_k}^{(j+1)T_s + \tau_k} s_k^{(j)}(t - \tau_k) n(t) dt \quad (11)$$

이다. 본 논문에서 편의상 $A_k^{(j)}$ 는 실수로 가정한
다. 길이 P 인 패킷 데이터에 대해 생각하여, 식
(6)의 벡터를 P 개를 연결하면

$$\begin{aligned} \tilde{\mathbf{y}} &= [\mathbf{y}^{(0)T}, \dots, \mathbf{y}^{(P-1)T}]^T \\ &= \tilde{\mathbf{R}} \tilde{\mathbf{A}} \tilde{\mathbf{b}} + \tilde{\mathbf{n}} \end{aligned} \quad (12)$$

와 같이 쓸 수 있고, 여기서

$$\tilde{\mathbf{A}} = \text{diag}[\mathbf{A}^{(0)}, \dots, \mathbf{A}^{(P-1)}] \quad (13)$$

$$\tilde{\mathbf{b}} = [\mathbf{b}^{(0)T}, \dots, \mathbf{b}^{(P-1)T}]^T \quad (14)$$

$$\tilde{\mathbf{n}} = [\mathbf{n}^{(0)T}, \dots, \mathbf{n}^{(P-1)T}]^T \quad (15)$$

$$\tilde{\mathbf{R}} = \begin{pmatrix} \mathbf{R}^{(0)}(0) & \mathbf{R}^{(0)}(1) & 0 & 0 & \cdots & 0 \\ \mathbf{R}^{(1)}(-1) & \mathbf{R}^{(1)}(0) & \mathbf{R}^{(1)}(1) & 0 & \cdots & 0 \\ 0 & \mathbf{R}^{(2)}(-1) & \mathbf{R}^{(2)}(0) & \mathbf{R}^{(2)}(1) & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \mathbf{R}^{(L-2)}(1) \\ 0 & 0 & 0 & 0 & \mathbf{R}^{(L-1)}(-1) & \mathbf{R}^{(L-1)}(0) \end{pmatrix} \quad (16)$$

이고 $\tilde{\mathbf{R}}$ 에 대해서는 다음 페이지 식 (16)에 기술되어 있다.

비동기 CDMA에서는 식 (6)에서 보듯이, 현재의 비트는 앞, 뒤의 비트(예로 j 번째인 경우, $(j-1)$ 번째와 $(j+1)$ 번째)와 관련되어 있어 최적성능을 얻기 위해서는 전체 패킷 모두를 고려하여 검출하는 것이 필요하다. 그러나 동기전송인 경우 패킷 전체를 고려하지 않고 검출하고자 하는 비트 구간만 고려하여도 최적 성능을 얻을 수 있다. 본 논문에서 제안하는 알고리즘은 동기식 뿐만 아니라 비동기 CDMA에도 적용할 수 있다. 알고리즘 원리를 쉽게 설명하고 관련 이론의 간편한 전개를 위해 동기식에 대해 유도한다. 동기전송에서 $\tau_1 = \tau_1 = \cdots \tau_K = 0$ 와 같이 놓을 수 있다. 이때 0 번째 비트를 검출한다고 하면

$$\mathbf{y}' = \mathbf{R}'\mathbf{A}\mathbf{b} + \mathbf{n} \quad (17)$$

같이 나타낼 수 있다. 여기서 $\mathbf{y}' = \mathbf{y}^{(0)}$, $\mathbf{R}' = \mathbf{R}^{(0)}(0)$, $\mathbf{A} = \mathbf{A}^{(0)}$, $\mathbf{b} = \mathbf{b}^{(0)}$, $\mathbf{n} = \mathbf{n}^{(0)}$ 이다. 동기 CDMA 수식에서 생략으로 인해 혼란이 없으면 위첨자 (j)를 표시하지 않고 사용한다.

3. Maximum Likelihood에 기초한 데이터 벡터 검출

입력 잡음 $n(t)$ 는 Gaussian 랜덤 프로세스이므로, 식 (17)에서의 정합필터 출력 잡음벡터 \mathbf{n} 은 Gaussian 확률밀도 함수를 가진다. 입력 잡음 $n(t)$ 는 시간상에서 상관되어있지 않으나 정합필터를 통과한 출력 잡음벡터는 식 (11)에서 보듯이 확산 부호와 입력 잡음간 상관을 취해 얻어져서 \mathbf{n} 의 각 성분은 서로 상관되게 된다. 데이터 벡터 \mathbf{b} 가 전송되었다는 가정 하에서, \mathbf{n} 에 대한 조건부 확률밀도 함수는

$$p(\mathbf{y}|\mathbf{b}) = \frac{1}{\sqrt{(2\pi)^K \sigma_{n_o}^2 |\mathbf{R}|}} \cdot \exp\left(-\frac{(\mathbf{y} - \mathbf{R}\mathbf{A}\mathbf{b})^T \mathbf{R}^{-1}(\mathbf{y} - \mathbf{R}\mathbf{A}\mathbf{b})}{2\sigma_{n_o}^2}\right)$$

와 같이 주어진다. 여기서 $\sigma_{n_o}^2 = N_0/2T_s$ 이다. ML(maximum likelihood) 방식에서는 log likelihood 함수 $\lambda(\mathbf{b})$ 를 계산하는 것이 필요하다.

$$\lambda(\mathbf{b}) = 2 \mathbf{b}^T \mathbf{y} - \mathbf{b}^T \mathbf{R} \mathbf{b} \quad (19)$$

여기서

$$\mathbf{y} = \mathbf{A} \mathbf{y}' \quad (20)$$

$$\mathbf{R} = \mathbf{A} \mathbf{R}' \mathbf{A} \quad (21)$$

이다. 최적 ML 수신기는 주어진 \mathbf{y} 에 대해 likelihood 함수가 최대가 되도록 데이터 비트를 추정한다. 즉, 추정된 데이터 벡터 $\hat{\mathbf{b}}$ 는

$$\hat{\mathbf{b}} = \arg \max_{\mathbf{b}} \lambda(\mathbf{b}) \quad (22)$$

와 같이 주어진다. 이와 같이 추정된 비트는 주어진 \mathbf{y} 에 대해 확률밀도 함수를 최대로 한다. 식 (22)는 조합 2차(combinational quadratic) 최대화 문제이다. 이 문제의 정확한 해는 2^K 의 \mathbf{b} 에 대한 모든 가능한 조합에 대해 계산하는 것이 요구된다. 사용자 수 K 가 커지면 계산이 매우 복잡하여 정확한 해를 구하는 것은 불가능하다.

3.1. Gradient guided search

Gradient guided search(GS) 알고리즘은 최적 해 대신에 준 최적 해(suboptimal solution)를 구함으로써 복잡한 계산을 크게 경감하고 있다[5]. 이 알고리즘에서는 반복계산(iteration)을 통해 구하고자 하는 해에 접근해 간다. l 번째 반복단계에서 추정된 데이터 벡터는

$$\hat{\mathbf{b}}^l = \arg \max_{H(\mathbf{b} - \hat{\mathbf{b}}^{l-1}) \leq q} \lambda(\mathbf{b}) \quad (23)$$

와 같고, $H(\cdot)$ 인수에 대한 Hamming weight를 나타낸다. 구의 중심은 전 단계에서 구한 추정 벡터 $\hat{\mathbf{b}}^{l-1}$ 이고 반경이 q 인 Hamming sphere에서 $\lambda(\mathbf{b})$ 를 최대로 하는 벡터가 $\hat{\mathbf{b}}^l$ 이다. 초기 추정 벡터 $\hat{\mathbf{b}}^0$ 를 구한 후, 각 단계에서 $(2^q - 1)$ 조합

에 대해 metric 계산이 필요하다. 만약 $q=K$ 이면 모든 조합에 대해 탐색하는 경우로 최적 해를 구할 수 있지만 앞에서 설명한 대로 계산이 매우 복잡해진다. $q=1$ 일 때가 복잡도가 가장 적다.

$q=1$ 이라 생각한다. l 번째 반복단계에서는 전 단계에서 구한

$$\hat{\mathbf{b}}_i^{l-1} = [\mathbf{b}_1^{l-1}, \dots, \mathbf{b}_K^{l-1}]^T \quad (24)$$

를 가지고 $\hat{\mathbf{b}}^l$ 을 구한다. i 번째 탐색 벡터 \mathbf{b}_i^l 은

$$\mathbf{b}_i^l = [\mathbf{b}_1^{l-1}, \dots, \mathbf{b}_{i-1}^{l-1}, -\mathbf{b}_i^{l-1}, \mathbf{b}_{i+1}^{l-1}, \dots, \mathbf{b}_K^{l-1}]^T \quad (25)$$

와 같으며, 다음처럼 likelihood 차에 비례하는 값 ξ_i^l 을 정의한다.

$$\xi_i^l = \frac{1}{4} [\lambda(\mathbf{b}_i^l) - \lambda(\hat{\mathbf{b}}_i^{l-1})] \quad (26)$$

식 (25)를 식 (26)에 대입하면

$$\xi_i^l = -\hat{\mathbf{b}}_i^{l-1} (y_i - \sum_{j \neq i} r_{ij} \mathbf{b}_j^{l-1}) \quad (27)$$

와 같이 구해지며, r_{ij} 은 \mathbf{R} 의 (i, j) 요소를 나타낸다. $\hat{\mathbf{b}}_i^{l-1}$ 이 실제 전송된 비트와 같다면 ξ_i^l 의 평균은 0보다 크게 된다[6]. 각 ξ_i^l ($i = 1, \dots, K$)를 계산하여 이 중 가장 큰 값이 ξ_m^l 이고 $\xi_m^l > 0$ 이면 $\hat{\mathbf{b}}^l$ 의 m 번째 요소 $\hat{\mathbf{b}}_m^l$ 은 $\hat{\mathbf{b}}_m^{l-1} = -\hat{\mathbf{b}}_m^{l-1}$ 이고 나머지 요소는 $\hat{\mathbf{b}}^{l-1}$ 의 대응하는 요소와 같다. 기존의 방법[5]에서는 ξ_i^l 를 식 (27)에 따라 계산되며, 이 방법은 각 반복단계에서 $K(K-1)$ 번의 덧셈을 필요로 한다. 이보다 크게 계산량을 줄일 수 있는 효율적인 방법을 제안한다. 본 논문에서 기존의 방법을 CGS (conventional GS)라 부른다.

3.2. 빠른 알고리즘

식 (25)는

$$\mathbf{b}_i^l = \mathbf{b}^{l-1} + \mathbf{c}_i^l \quad (28)$$

와 같이 나타낼 수 있다. 여기서

$$\mathbf{c}_i^l = [0, \dots, 0, -2\mathbf{b}_i^{l-1}, 0, \dots, 0]^T \quad (29)$$

이다. 식 (26), (28)로부터 ξ_i^l 은

$$\xi_i^l = -\frac{1}{2} \mathbf{c}_i^{lT} \mathbf{g}^{l-1} - r_{ii} \quad (30)$$

처럼 나타낼 수 있고, 여기서

$$\mathbf{g}^{l-1} = -\mathbf{y} + \mathbf{R} \hat{\mathbf{b}}^{l-1} \quad (31)$$

이다. $\hat{\mathbf{b}}^{l-1}$ 과 $\hat{\mathbf{b}}^{l-2}$ 는 한 비트만 다르며, 다른 비트를 d_{l-1} 번째 비트라 하자. $\hat{\mathbf{b}}^{l-1}$ 과 $\hat{\mathbf{b}}^{l-2}$ 은

$$\hat{\mathbf{b}}^{l-1} = \hat{\mathbf{b}}^{l-2} + \mathbf{e}^{l-1}, \quad l \geq 2 \quad (32)$$

과 같은 관계를 가지며, \mathbf{e}^{l-1} 은 d_{l-1} 번째 비트가 $2\mathbf{b}_{d_{l-1}}^{l-1}$ 이고 나머지 요소는 모두 0인 벡터이다. 식 (32)를 식 (31)에 대입하면

$$\mathbf{g}^{l-1} = [\mathbf{g}_1^{l-1}, \mathbf{g}_2^{l-1}, \dots, \mathbf{g}_K^{l-1}]^T = \mathbf{g}^{l-2} + 2\mathbf{b}_{d_{l-1}}^{l-1} \mathbf{r}_{d_{l-1}} \quad (33)$$

와 같이 표현할 수 있고, \mathbf{r}_n 은 \mathbf{R} 의 n 번째 열을 나타낸다. 식 (30)은

$$\xi_i^l = \mathbf{b}_i^{l-1} \mathbf{g}_i^{l-1} - r_{ii} \quad (34)$$

와 같이 표현된다. 식 (34)와 같이 ξ_i^l 에 대한 간결한 표현식을 유도하였고, 식 (33)에 의해 \mathbf{g}^{l-1} 을 효과적으로 구함에 따라 계산량이 경감된다. 수 있다. $\mathbf{b}_{d_{l-1}}^{l-1} = \hat{\mathbf{b}}^{l-2}$ 이므로 $i = d_{l-1}$ 에 대한 ξ_i^l 는 구할 필요가 없다. 또한 $\mathbf{b}_i^{l-1} \mathbf{g}_i^{l-1} \leq 0$ 이면 $r_{ii} > 0$ 이므로 이 경우 역시 ξ_i^l 을 구할 필요가 없다. 어떤 단계 m 에서 $i = 1, \dots, K$ 모두에 대해 0보다 큰 ξ_m^l 이 없다면 m 번째 단계에서 알고리즘은 종료되고 구하는 해 $\hat{\mathbf{b}}$ 은 $\hat{\mathbf{b}} = \hat{\mathbf{b}}^{m-1}$ 이다. 이 경우 $\hat{\mathbf{b}}^m$ 은 $\hat{\mathbf{b}}^{m-1}$ 과 같다. 초기 추정치 $\hat{\mathbf{b}}^0$ 이 주어졌을 때, 제안한 FGS (fast GS) 알고리즘은 첫 단계를 제외한 각 단계에서 많아야 $2K-1$ 번의 덧셈을 요구한다. 첫 단계에서는 \mathbf{g}^0 의 계산을 포함해서 $K(K+1)$ 번 이하의 덧셈이 필요하다. FGS는 매 단계마다 $K(K-1)$ 번의 덧셈을 요구하는 CGS보다 계산량이 매우 적다. 특히 K 가 크거나 여러 번의 반복계산을 통해 해를 얻는 경우 FGS의 상대적 이점이 크다.

4. 컴퓨터 시뮬레이션

FGS, CGS의 성능을 계산 복잡도 면에서 비교하기 위해 시뮬레이션을 하였다. 계산 방식은 다르지만 FGS, CGS 모두 식 (23) ($q=1$)에 따라 데이터 벡터가 추정되므로 BER 성능은 동일하다. 정합 필터와 decorrelator의 두 방식을 이용하여 초기 추정을 하였다. 정합필터 방식에서

$$\mathbf{b}_i^0 = \begin{cases} 1 & \text{if } y_i > 0 \\ -1 & \text{if } y_i < 0 \end{cases}$$

(35)

와 같이 초가치가 추정되며, y_i 는 \mathbf{y} 의 i 번째 요소이다. Decorrelator에서는

$$\mathbf{z} = \mathbf{R}^{-1} \mathbf{y}'$$

을 계산하여

$$b_i^0 = \begin{cases} 1 & \text{if } z_i > 0 \\ -1 & \text{if } z_i < 0 \end{cases}$$

(36)

와 같이 초가치가 추정되며, z_i 는 \mathbf{z} 의 i 번째 요소이다. 각 사용자의 수신 전력의 세기는 같다. 확산 부호간의 정규화된 상호상관(normalized cross correlation)은 모두 0.8로 똑같이 놓았다. 10^6 번의 독립적인 시뮬레이션을 하여 BER(bit error rate)과 덧셈의 평균 횟수를 구하였다.

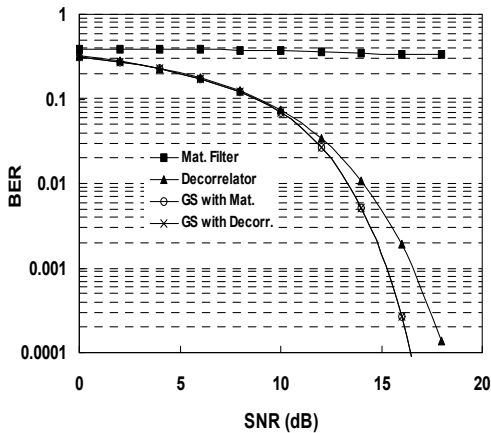


그림 1 SNR에 대한 BER, $K=20$

그림 1은 사용자 수 $K = 20$ 일 때 SNR (signal-to-noise ratio)에 대한 BER 성능을 보여준다. 이들 예에서 초기추정 방식에 관계없이 GS의 BER 성능은 같음을 보여준다. 정합필터, decorrelator에 대한 BER 성능도 포함하였다. 정합필터에서는 SNR이 증가하여도 확산부호들간 큰 상호관련으로 간섭도 같이 커져서 BER이 작아지지 않고 거의 일정하나, decorrelator에서는 간섭을 제거할 수 있어 SNR이 커짐에 따라 BER이 작아진다. 시뮬레이션 결과는 GS 방식을 도입함에 따라 BER 성능이 개선됨을 보여준다. Decorrelator의 성능과 비교하여 특히 SNR이 클 때 성능개선이 크다. 표 1에 FGS, CGS의 평균 덧셈횟수를 나타내었다. CGS에서 매 반복계산마다 $K(K-1)$ 번의 덧셈을 하므로 CGS의 평균 덧셈 수를 380로 나누면 평균 반복횟수를 구할 수 있다. 예상한 대로 decorrelator로 초기 추정을 하면 정합필터보다 정

확한 추정치를 얻을 수 있어 GS에서 계산량이 작아짐을 보인다. 그림 1에서 본 바와 같이 SNR이 증가하여도 정합필터에 의한 초기 추정은 거의 개선되지 않아 계산량의 감소가 미미하다. 반면에 decorrelator의 경우, SNR이 증가하면 할수록 더 정확하게 초기 추정치를 얻을 수 있어 CGS에서 계산량이 감소함을 나타낸다. 그러나 decorrelator에 의한 초기치를 사용하더라도 FGS의 경우 SNR 증가에 대해 계산량 감소는 매우 적다. 그 이유는 다음처럼 설명할 수 있다. decorrelator에 의한 초기치는 비교적 정확하여 반복횟수가 작다. 이처럼 반복횟수가 작을 때 FGS에서는 \mathbf{g}^0 계산에 최고 $K(K+1)$ 번의 덧셈이 필요하여 \mathbf{g}^0 의 계산에 소모 계산량의 대부분을 차지하기 때문이다. 초기 추정에 정합필터를 사용했을 때 여러 번의 반복계산이 필요하여, 예상한 대로, 이 경우 FGS의 계산상 이점이 상대적으로 더 크다. 또한 같은 이유로 SNR이 작을 때 FGS의 상대적 이점이 크다.

표 1 평균 덧셈횟수, $K = 20$

SNR (dB)	정합필터			Decorrelator		
	FGS의 덧셈 횟수	CGS의 덧셈 횟수	반복 횟수	FGS의 덧셈 횟수	CGS의 덧셈 횟수	반복 횟수
0	675	3199	8.42	453	959	2.52
4	683	3262	8.58	447	868	2.28
8	680	3226	8.49	438	747	1.97
12	671	3122	8.22	424	557	1.47
16	658	2980	7.84	411	394	1.04

그림 2는 $K = 30$ 인 경우이다. 이외 시뮬레이션 조건은 그림 1의 경우와 비슷하다. BER 성능은 그림 1과 비슷한 경향을 나타낸다. 표 2는 평균 덧셈 수를 보여준다. 표 1과 비교하면, 사용자 수가 많아짐에 따라 반복횟수가 커지고 따라서 계산량도 증가함을 나타낸다. 8 dB 이하 SNR에 대해 FGS에서 정합필터로 초기화해도 decorrelator로 초기화한 CGS보다 적은 계산량이 필요하다.

표 1과 2를 통해 FGS 방법은 식 (33), (34)와 같이 metric을 효율적으로 계산하여 CGS보다 적은 계산량으로 해를 구함을 확인할 수 있다.

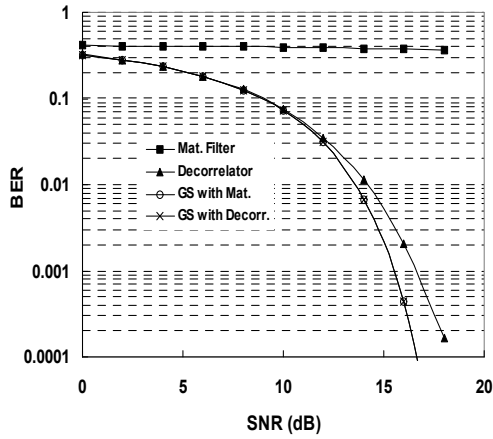


그림 2 SNR에 대한 BER, $K=30$
표 2 평균 덧셈횟수, $K = 30$

SNR (dB)	정합필터			Decorrelator		
	FGS의 덧셈 횟수	CGS의 덧셈 횟수	반복 횟수	FGS의 덧셈 횟수	CGS의 덧셈 횟수	반복 횟수
0	1,547	11,146	12.81	995	2,516	2.89
4	1,560	11,331	13.02	984	2,260	2.60
8	1,555	11,230	12.91	969	1,928	2.22
12	1,538	10,941	12.58	943	1,423	1.64
16	1,514	10,541	12.12	918	922	1.06

5. 결론

제안한 FGS 알고리즘은 ML metric을 효율적으로 계산한다. 이 방법에서는 첫 단계를 제외한 각 반복단계에서 단지 $O(K)$ 덧셈만을 필요로 한다. 기존의 방법인 CGS와 비교하여, SNR이 작거나 초기 추정에 정합필터를 사용했을 때 FGS에서 계산량의 경감이 상대적으로 크다. 계산 면에서 보면 정합필터의 사용이 바람직하다. Decorrelator를 이용하면 좀 더 정확한 추정을 할 수 있지만 역행렬 (inverse matrix)을 구하는 것이 필요하여 계산이 매우 복잡해진다. 크기가 K 인 역행렬을 구하는데 일반적으로 $O(K^3)$ 의 곱셈이 필요하다. 정합필터로 초기치를 추정하여도 BER 성능은 decorrelator를 이용한 것과 똑같다. 부정확한 초기치는 반복횟수를 증가시키지만 이에 따른 FGS에서의 계산량의 증가는 매우 작다. 식 (19)에서 행렬 R 대각요소(diagonal elements)의 metric에 대한 기여치는 b 의 함수가 아니므로 대각요소를 0으로 놓을 수 있다. 대각요소를 0으로 한 이 행렬을 이용하면 계

산량을 더욱 감소시킬 수 있다.

최근에 저자는 heuristics에 기초하여 계산 복잡도를 줄이는 알고리즘이 소개되었음을 알게 되었다[7]. 이 알고리즘은 본 논문에서 제안한 알고리즘과 접근방법을 달리 하지만 계산량은 $O(K)$ 로 제안방식과 비슷하다.

참고 문헌

- [1] M. Zeng, A. Annamalai, and V. K. Bhargava, "Recent advances in cellular wireless communications," *IEEE Commun. Mag.*, vol. 37, no. 9, pp. 128-138, Sept. 1999.
- [2] F. Adachi, M. Sawahashi, and H. Suda, "Wideband DS-CDMA for next generation mobile communications systems," *IEEE Commun. Mag.*, vol. 36, no. 9, pp. 56-69, Sept. 1998.
- [3] R. Steele and C.-C. Lee, GSM, cdmaOne and 3G systems. New York: John Wiley & Sons, Wiley, 2001.
- [4] S. Verdu, "Minimum probability of error for asynchronous Gaussian multiple-access channel," *IEEE Trans. Inform. Theory*, vol. 32, no. 1, pp. 1228-1239, Jan. 1986.
- [5] J. Hu and R. S. Blum, "A gradient guided search algorithm for multiuser detection," *IEEE Commun. Lett.*, vol. 4, no. 11, pp. 337-339, Nov. 2000.
- [6] Z. Gou and K. B. Letaief, "An effective multiuser receiver for DS/CDMA systems," *IEEE J. Select. Areas Commun.*, vol. 19, no. 6, pp. 1019-1028, June 2001.
- [7] H. S. Lim, and B. Venkatesh, "An efficient local search heuristics for asynchronous multiuser detection," *IEEE Commun. Lett.*, vol. 7, no. 6, pp. 299-301, June 2003.