

상향링크 다중사용자 MIMO/FDD 시스템을 위한 낮은 복잡도의 안테나 선택 기반 MIMO 스케줄링 기법

정희원 김요한*, 준회원 조성윤*, 이택주*, 정희원 김동구*

Low Complexity Antenna Selection based MIMO Scheduling Algorithms for Uplink Multiuser MIMO/FDD System

Yohan Kim* *Regular Member*, Sungyoon Cho*, Taekju Lee* *Associate Members*,
Dongku Kim* *Regular Member*

요약

본 논문에서는 상향링크 다중사용자 MIMO/FDD 시스템에서 최적 성능을 보이면서 낮은 복잡도를 가지는 안테나 선택 기반 스케줄링 방식을 제안한다. 가장 복잡하면서 가장 우수한 성능을 보이는 스케줄링 알고리즘인 Brute-Force 스케줄러(BFS)의 성능에 근접하면서 복잡도가 현저히 낮은 세 가지 스케줄링 방식을 제안하고 채널 용량, 복잡도, SER 관점에서의 성능 비교를 보인다. 실험 결과 제안한 스케줄링 방식 중 CSS(Capacity-Stacking Scheduler)와 SOAS(Semi-orthogonal Antenna Scheduler) 방식은 낮은 복잡도를 유지하면서 BFS와 같은 성능을 보였으며, 좀 더 복잡도를 줄인 RC-SOAS(Reduced Complexity-SOAS)의 경우 안테나 수가 적은 경우에는 BFS와의 성능차가 거의 없음을 확인하였다. 또한, 스케줄링을 하기 위한 안테나 채널의 집합을 형성하는 방식으로 Full-Set(FS) 방식과 Sub-Set(SS) 방식을 비교하여, SS 방식을 통해 안테나 수가 적은 경우 성능 열화 없이 추가적으로 복잡도의 이득을 얻을 수 있음을 보였다.

Key Words : Multiuser MIMO, Uplink, FDD, Antenna selection, Scheduler, Complexity

ABSTRACT

Antenna selection based MIMO schedulers are proposed to achieve the optimal performance with low complexity in uplink multiuser MIMO/FDD system. In this paper, three heuristic schedulers are proposed to approach the optimal performance which is achieved by the optimal Brute-Force Scheduler. Two search methods called sub-set and full-set way are also discussed to set up the antenna channels to be the candidates of the scheduler. Simulation results show that the sum rate and BER performance of the proposed CSS and SOAS schemes are about the same to that of the brute-force scheduler with affordable complexity, while RC-SOAS with further reduced complexity achieves almost the optimal performance in the case of small number of antennas. Moreover, the complexity can be additionally reduced by the sub-set search method when the number of transmit and receive antennas are 2 respectively, which is applicable in the realistic systems.

※ 본 연구는 삼성전자 SKYPASS 4G 프로젝트의 지원으로 수행되었음.

* 연세대학교 전기전자공학과 이동통신 연구실(john5958@yonsei.ac.kr, acecho@yonsei.ac.kr, leetj@lge.com, dkkim@yonsei.ac.kr)
논문번호 : KICS2007-08-356, 접수일자 : 2007년 8월 8일 최종논문접수일자 : 2007년 11월 15일

I. 서 론

다중 사용자 MIMO 시스템은 최근 몇 년간 매우 활발하게 연구되어 왔다. 특히 최근의 다수의 연구자들에 의해 상향링크 모두 채널 용량 지역(channel capacity region)이 밝혀졌고, 이 채널 용량에 도달하기 위한 많은 연구들이 이루어졌다^{[1][2][3][4]}. 다중 사용자 MIMO 시스템에서의 하향링크는 보통 방송채널(BC: Broadcast Channel)로 불리며 더티 페이퍼 부호화(DPC: Dirty Paper Coding)에 의해 합 채널 용량(sum capacity)와 채널 용량 지역 모두 도달 가능성이 알려져 있으며^{[1][4]}, 상향링크의 경우 다중 접속 채널(MAC: Multiple Access Channel)로 부르고 이 경우의 합 채널 용량과 채널 용량 지역은 BC의 경우에 비해 다루기 쉽고, 그 최적 값 또한 비교적 오래 전부터 알려져 있었다^{[3][8]}.

본 논문은 MIMO MAC 시스템을 기반으로 한다. MIMO MAC의 경우, 반복적 워터 필링(IWF: Iterative Waterfilling) 알고리즘을 이용하여 모든 사용자들의 송신 공분산 행렬(Transmit Covariance Matrix) 계산하는 방식이 합 채널 용량에 도달하는 가장 효과적인 방식이라는 것과, 이때 수신단의 구조는 MMSE-SIC (Minimum Mean Squared Error-Successive Interference Cancellation) 방식이 최적임이 알려져 있다^{[3][7][8]}. 하지만 이 방식은 이론적으로는 최적의 성능을 보이지만, 계산의 복잡도와 채널정보 및 송신 공분산 행렬 등의 교환량이 매우 크기 때문에 실제적으로는 구현하기 어려운 기법이다.

실제 구현 관점에서는 알고리즘의 복잡도와 더불어 궤환(Feedback) 정보량을 줄이는 문제가 중요한 이슈가 된다. 특히 본 논문에서 가정된 FDD(Frequency Division Duplexing) 시스템의 경우, 위에서 언급한 IWF 방식을 사용하기 위해서는 상향링크 채널 탐색 신호(Channel probing signal)를 통해 추정된 채널로 기지국에서 송신 공분산 행렬을 계산한 후, 각 송신 공분산 행렬을 하향링크 궤환 채널을 통해 해당 사용자들에게 전송해야 한다. 이 경우 궤환 정보량은 사용자 수 만큼의 복소 MIMO 채널값을 포함해야 하므로, 시스템에 큰 부담이 된다. 또한, 채널 추정 오류나 궤환을 위한 양자화(Quantization) 오류 등에 의해 큰 성능 저하를 가져올 수 있다. 따라서 보다 현실적인 기법을 위해서 궤환 정보량이 제한된 시스템을 고려해야 할 필요가 있다.

궤환 정보량이 제한된 상황에서 단일 사용자 MIMO 시스템에서의 성능 이득을 위한 기법으로 연구된 안테나 선택(AS: Antenna Selection) 방식은 MIMO MAC 시스템에서도 응용가능하다. 각 사용자와 기지국이 다중 안테나를 가지고 있는 경우, 전체 안테나 채널을 고려한 공간 자유도(Spatial Degrees of Freedom)가 제한되어있기 때문에, 동시에 전송 가능한 총 데이터 스트림의 수 역시 제한된다. 따라서, AS의 적용이 단순히 한 사용자의 안테나를 선택하는데 그치지 않고, 전송 가능한 사용자까지 선택하는 사용자 스케줄러의 형태로 확장된다. 이때, 다음의 두 가지 요구조건이 필요하다.

- * 주어진 환경에서 최대의 합 채널 용량을 달성할 수 있을 것

- * 최적 성능을 보장하면서 복잡도를 낮출 것

본 논문에서는 상향링크 FDD 시스템에서 위의 두 가지 요구조건을 만족시키기 위한 세 가지의 발전적(heuristic) 안테나 선택 기반 MIMO 다중 사용자 스케줄러를 제안한다. 지금까지 제한된 사용자수, 제한된 송수신 안테나 환경에서의 최대의 합 채널 용량에 도달하기 위해 선택된 안테나 채널의 필요 충분 조건은 알려져 있지 않다. 이때, 가능한 최적의 스케줄링 방식은 Brute-Force 스케줄링 알고리즘(BFS: Brute-Force Scheduling)으로, 스케줄링의 대상이 되는 모든 사용자의 모든 안테나로 가능한 모든 안테나 조합을 생성한 후 최적의 합 채널 용량을 가지는 채널을 선택하는 알고리즘이다. 하지만, 안테나와 사용자의 수가 증가할수록 지수적인 복잡도의 증가를 보이기 때문에, BFS의 성능을 보존하면서 낮은 복잡도를 가지는 스케줄링 알고리즘의 설계가 요구된다. 본 논문은 이러한 목적을 달성하기 위한 스케줄러를 제안하고, 그 성능을 채널 용량, 복잡도, SER(Symbol Error Rate)의 관점에서 분석한다. 또한 스케줄링 집합(Scheduling Set)을 정의하여, 성능 저하 없이 복잡도를 추가적으로 줄일 수 있는 상황을 살펴본다. 또 다른 비교 대상으로 가장 복잡도가 낮은 스케줄링 방식 중 하나인 Max-SNR 스케줄러와의 성능 비교를 통해 제안한 스케줄러들의 성능 이득을 확인한다.

본 논문의 구성은 다음과 같다. II장에서 시스템 모델과 공간 자유도 문제를 제시하고, III장에서 기존의 스케줄러와 제안한 방식들의 구조를 보이며, IV장에서 성능 분석을 통해 제안한 기법들의 효율성을 확인한다. V장에서는 결론을 짓는다.

II. 시스템 모델

2.1 시스템 모델

기지국은 N_R 개의 수신 안테나를 가지고 있고, 총 K 명의 사용자는 각각 N_T 개의 송신안테나를 가지고 있다.

이때, 사용자 k 의 채널 행렬은 $\mathbf{H}_k = [\mathbf{h}_{k,1} \mathbf{h}_{k,2} \cdots \mathbf{h}_{k,N_T}]$ 로 표현되고, 이때 $\mathbf{h}_{k,l}, l=1, \dots, N_T$ 는 $N_R \times 1$ 인 평균 $\mathbf{0}$ 이고 분산 1인 복소 가우시안 채널을 나타낸다. 본 논문에서는 모든 사용자의 기지국에서의 수신 SNR은 동일하다고 가정한다.

앞서 언급했듯이, 본 논문에서는 FDD 방식이 가정되었다. 이 경우 각 사용자는 모든 안테나를 통해서 기지국으로 스케줄링용 채널 탐색 신호(channel probing signal)를 전송한다. 이때, 모든 사용자의 모든 안테나의 채널 탐색 신호를 기지국에서 정확히 수신할 수 있다고 가정한다. 따라서, 기지국은 시스템 내의 모든 안테나 채널을 추정할 수 있지만, 각 사용자들은 채널 정보를 가지고 있지 않다. 이후, 기지국은 스케줄러에 의해서 전송에 필요한 적절한 안테나를 선택하게 되고, 해당하는 안테나의 인덱스를 별도의 하향링크 채널을 통해서 사용자들에게 전송하게 된다. 그림 1은 시스템의 기본 구조를 보여준다.

각 사용자들의 채널이 주어질 때, 스케줄러를 고려하지 않고 모든 사용자가 동시에 전송한다고 가정했을 때 다중 사용자 MIMO 상향링크 시스템의 합 채널 용량은 다음과 같은 상향 한계(Upper bound)를 가진다.

$$R_s \leq \log_2 \det \left(\mathbf{I}_{N_R} + \sum_{k=1}^K \mathbf{H}_k \mathbf{Q}_k \mathbf{H}_k^* \right) \quad (1)$$

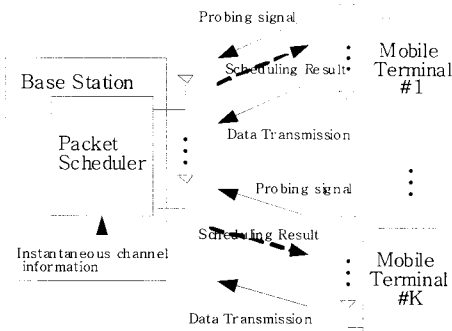


그림 1. 상향링크 다중사용자 MIMO/FDD 시스템의 개괄도

이때, \mathbf{Q}_k 는 사용자 k 의 송신 공분산 행렬이다. 임의의 AS 스케줄러를 q 라고 하고, 스케줄러가 각 사용자들에게서 하나의 안테나만 선택한다고 가정하면 식 (1)은 다음과 같이 바뀐다. 이때 사용자 선택은 고려되지 않는다.

$$R_s \leq \log_2 \det \left(\mathbf{I}_{N_R} + \sum_{k=1}^K \mathbf{H}_k \mathbf{Q}_k \mathbf{H}_k^* \right) \quad (2)$$

$$= \log_2 \det \left(\mathbf{I}_{N_R} + SNR \sum_{k=1}^K \mathbf{H}_k \mathbf{e}_{k,q(k)} \mathbf{e}_{k,q(k)}^* \mathbf{H}_k^* \right)$$

여기서, $\mathbf{e}_{k,q(k)} = [0 \cdots 1 \cdots 0]^T$ 는 안테나 선택을 나타내는 벡터로, 오직 $q(k)$ 번째 위치에서만 1의 값을 가지고 나머지 위치에서는 0의 값을 가진다. $q(k)$ 는 1에서 N_T 사이의 값을 가지며, 스케줄러 q 에 의해서 선택된 사용자 k 의 안테나 인덱스를 나타낸다. 각 사용자들이 하나의 안테나로만 전송한다고 가정되었기 때문에, 이 경우에서의 송신 공분산 행렬은 $SNR \cdot \mathbf{e}_{k,q(k)} \mathbf{e}_{k,q(k)}^*$ 의 형태로 나타난다. 결과적으로 $\mathbf{H}_k \mathbf{e}_{k,q(k)} = \mathbf{h}_{k,q(k)}$ 이고, 이는 채널 행렬 \mathbf{H}_k 의 $q(k)$ 번째 안테나 채널 성분을 나타내므로, 식 (2)는 다음과 같이 다시 쓸 수 있다.

$$R_s \leq \log_2 \det \left(\mathbf{I}_{N_R} + SNR \sum_{k=1}^K \mathbf{h}_{k,q(k)} \mathbf{h}_{k,q(k)}^* \right) \quad (3)$$

· 식 (3)의 합 채널 용량은 MMSE-SIC 수신기로 도달 가능하다^{[3][7]}.

2.2 공간 자유도 문제

식 (3)은 모든 사용자에게 적용된 임의의 AS 스케줄러 q 에 의해서 선택된 안테나들이 동시에 데이터를 전송할 때의 합 채널 용량을 나타내는데, 본 장에서는 동시에 전송되는 데이터 스트림의 수를 결정하는 문제를 살펴볼 것이다.

식 (3)은 다음과 같이 다시 쓸 수 있다.

$$R_s \leq \log_2 \det \left(\mathbf{I}_{N_R} + SNR \sum_{k=1}^K \mathbf{h}_{k,q(k)} \mathbf{h}_{k,q(k)}^* \right) \quad (4)$$

$$= \log_2 \det \left(\mathbf{I}_{N_R} + SNR \mathbf{H}_q \mathbf{H}_q^* \right)$$

여기서, $\mathbf{H}_q = [\mathbf{h}_{1,q(1)} \mathbf{h}_{2,q(2)} \cdots \mathbf{h}_{K,q(K)}]$ 는 $N_R \times K$ 행렬로, 스케줄러 q 에 의해 선택된 각 사용자들의 안테나 채널의 집합이다. 따라서, 다음 식을 얻을 수 있다.

$$\begin{aligned}
 R_s &\leq \log_2 \det(\mathbf{I}_{N_R} + SNR \mathbf{H}_q \mathbf{H}_q^*) & (5) \\
 &= \log_2 \det(\mathbf{I}_{N_R} + SNR \mathbf{U}_q \mathbf{A}_q \mathbf{U}_q^*) \\
 &= \log_2 \det(\mathbf{I}_{N_R} + SNR \mathbf{A}_q)
 \end{aligned}$$

이때, \mathbf{A}_q 는 대각 성분이 $\mathbf{H}_q \mathbf{H}_q^*$ 의 고유값(eigenvalue)들인 대각 행렬이다. 간단한 선형대수를 통해서 $\mathbf{H}_q \mathbf{H}_q^*$ 의 고유값의 최대 개수는 $\min(N_R, K)$ 임을 쉽게 알 수 있다. 따라서, 이는 구분할 수 있는 최대의 공간 차원이 $\min(N_R, K)$ 이고, 시스템이 채널 \mathbf{H}_q 를 통해 동시에 지원할 수 있는 스트림의 최대 개수 또한 $\min(N_R, K)$ 임을 의미한다. 이때 $\min(N_R, K)$ 를 공간 자유도(spatial degrees of freedom)라 한다. 이러한 공간 자유도의 문제는 단일 사용자 MIMO 시스템에서의 논의와 매우 유사하며^[6], 공간 자유도를 초과한 개수의 스트림의 용량이 시스템의 합 채널 용량에서 차지하는 비중은 미미함이 알려져 있다^[7]. 본 논문에서의 다루는 상황은 $K \gg N_R$ 이므로, 스케줄러 설계시 최대 N_R 개의 스트림, 즉 N_R 개의 안테나를 선택하도록 가정하게 된다. 따라서, 지금까지 가정한 스케줄러 q 의 목적이 $N_R \times K$ 차원의 \mathbf{H}_q 이 아닌, 모든 가능한 안테나 채널의 집합에서 $N_R \times N_R$ 차원의 \mathbf{H}_q 를 골라내는 것으로 바뀌게 된다.

따라서, N_R 의 공간자유도에 의한 합 채널 용량은 다음과 같이 쓸 수 있다.

$$\begin{aligned}
 R_s &\leq \log_2 \det\left(\mathbf{I}_{N_R} + SNR \sum_{j=1}^{N_R} \mathbf{h}_{q(j)} \mathbf{h}_{q(j)}^*\right) & (6) \\
 &= \log_2 \det(\mathbf{I}_{N_R} + SNR \mathbf{H}_q \mathbf{H}_q^*)
 \end{aligned}$$

여기서 $q(j) = (k, l)$ 은 스케줄링 알고리즘 q 에 의해 선택된 사용자 k 의 안테나 인덱스 l 를 나타내고, $\mathbf{H}_q = [\mathbf{h}_{q(1)}, \mathbf{h}_{q(2)}, \dots, \mathbf{h}_{q(N_R)}]$ 이다. 식 (6)의 합 채널 용량 성능은 \mathbf{H}_q 의 특성에 의해 결정되고, 이는 스케줄러 q 의 목적 및 성질에 따라 달라진다.

III. 스케줄링 알고리즘

스케줄링 알고리즘을 설계하는데 있어서 서론에서 언급했듯이 다음의 두 가지 목적이 중요한 조건이 된다. 첫 번째는 합 채널 용량을 최대화 하는 것이고, 두 번째는 스케줄링 복잡도를 낮추는 것이다. 따라서, 최적의 스케줄러는 최대의 합 채널 용량 성능을 보이면서 낮은 복잡도가 유지되는 스케

줄러이다. 주어진 채널 행렬 \mathbf{H}_k 들로부터 최대의 합 채널 용량을 가지는 안테나 채널 조합의 성질에 대한 이론적 필요 충분 조건은 아직 알려져 있지 않다. 따라서, 본 논문에서는 세 가지의 제안된 스케줄러를 포함하는 총 다섯 가지의 스케줄링 방식들을 합 채널 용량, SER, 복잡도 성능을 기준으로 비교하도록 한다.

스케줄링 알고리즘을 논의하기 전에, 스케줄러의 대상이 되는 안테나 채널의 집합을 형성하는 다음의 두 가지 방법을 정의한다.

Full-Set(FS) 방법 : N_T 개의 모든 송신 안테나가 스케줄링의 개별 후보로 간주된다. 따라서, 한 사용자에게서 2개 이상의 안테나가 선택될 수도 있으며, 이때에는 해당 사용자의 총 송신 전력이 선택된 안테나의 개수로 나뉘어져야 한다.

Sub-Set(SS) 방법 : FS 방법이 원칙적으로는 최적의 안테나 집합을 찾아내는 방식이다. 하지만, 사용자 수가 많고, 송신 안테나 수가 늘어나면 FS 방법은 스케줄링 복잡도를 상당히 증가시킬 수 있다. 따라서, SS 방법에서는 전체의 송신 안테나를 대상으로 하지 않고, 각 사용자의 안테나 중 채널 놈(Norm)이 가장 큰 안테나 하나씩만 골라서 스케줄링의 대상으로 간주한다. 따라서, 사용자가 K 명 일 때, SS 방식으로 형성된 안테나 채널 집합은 총 K 개의 후보 안테나 채널을 포함하게 된다. 이 경우는 선택된 송신 안테나 마다 해당 사용자의 총 송신 전력을 사용하게 된다.

위의 FS 또는 SS 방법에 의해 형성된 후보 안테나들의 집합을 ‘스케줄링 집합(Scheduling Set)’이라고 하고 본 논문에서는 S 로 표기하도록 한다. 3.1과 3.2는 기존의 스케줄러를, 3.3에서 3.5는 제안된 스케줄러를 보여준다.

3.1 Brute-Force 스케줄러 (BFS: Brute-Force Scheduler)

BFS는 스케줄링 집합에 있는 안테나들의 모든 가능한 조합을 테스트하여 최대의 합 채널 용량을 가지는 최적 조합을 찾아낸다. BFS가 최적의 알고리즘으로 알려져 있지만, 그 복잡도가 스케줄링 집합에서의 조합의 수에 의존하기 때문에 사용자가 많거나 안테나가 많은 시스템에서는 복잡도가 지수적으로 증가하게 된다.

BFS의 간략한 과정은 다음과 같다.

a. 초기화

FS 나 SS 방식으로 스케줄링 집합 S 를

생성한다.

b. 스케줄링

i) 조합 생성 과정을 G 라고 하고 모든 가능한 채널 조합들, 즉

$\mathbf{H}_G = [\mathbf{h}_{G(1)} \ \mathbf{h}_{G(2)} \ \dots \ \mathbf{h}_{G(N_R)}]$ 의 집합을 생성한다. 여기서 $G(i), i=1, \dots, N_R$ 은 스케줄링 집합 S 에 속한 안테나 채널 중 G 에 의해 골라진 안테나의 인덱스이다.

ii) 생성된 모든 채널들 \mathbf{H}_G 들 중, 최대의 $\log_2 \det(\mathbf{I}_{N_r} + SNR \mathbf{H}_G \mathbf{H}_G^*)$ 를 가지는 채널 $\mathbf{H}_{G,MAX}$ 를 찾는다.

3.2 Max-SNR 스케줄러 (MSS: Max-SNR Scheduler)

MSS는 가장 간단한 형태의 스케줄러이다. MSS는 스케줄러 집합의 후보 안테나 채널 중에 가장 채널의 놈(Norm)이 큰 N_r 개의 안테나를 선택한다^[5]. 따라서, MSS의 복잡도는 놈의 계산에 필요한 계산량과 최대값을 찾기 위한 정렬 과정에 의존하게 된다. 하지만, MSS는 수신단의 구조나 합 채널 용량 등의 성능 기준을 고려하지 않았기 때문에 성능 이득이 보장되지는 않는다. MSS의 과정은 다음과 같다.

a. 초기화

FS 나 SS 방식으로 스케줄링 집합 S 를 생성한다.

b. 스케줄링

i) 스케줄링 집합 S 에 있는 모든 안테나 채널의 놈 $\|\mathbf{h}\|^2$ 를 구한다.

ii) 놈들을 내림차순으로 정렬한다.

iii) 가장 놈이 큰 N_r 개를 선택한다.

3.3 Capacity-Stacking 스케줄러 (CSS: Max-SNR Scheduler)

CSS는 최대 합 채널 용량을 얻을 수 있는 채널을 선택하기 위해 순차적인 방식으로 안테나를 선택한다. 즉, 각 단계에서 최대의 채널 용량을 보이는 안테나를 선택하고, 선택한 안테나 채널을 앞서 선택된 채널에 쌓는다(Stack). 다음 단계에서는 쌓여 있는 채널에 아직 후보로 남아있는 안테나 채널을 각각 쌓은 후 채널 용량을 비교하여 가장 채널 용량이 큰 안테나를 선택한다(Capacity-Stacking). 따라서, CSS의 복잡도는 정렬 과정과 채널용량을 계산하는 부분에 의존하게 되고, 이는 BFS 보다 월등

히 낮은 복잡도를 보이게 된다. CSS의 간략한 과정은 다음과 같다.

a. 초기화

스케줄링 집합 S 를 FS나 SS 방식으로 생성한다.

S 에 속해있는 안테나 채널 중, 놈이 가장 큰 안테나 채널 $\mathbf{h}_{CSS(1)}$ 를 선택하고 초기 행렬 $\mathbf{H}_{CSS,1} = [\mathbf{h}_{CSS(1)}]$ 를 생성한다. 이때, $CSS(i), i=1, \dots, N_r$ 은 CSS를 통해 i 번째 단계에서 선택된 안테나 채널의 인덱스를 나타낸다. $S = S/\mathbf{h}_{CSS(1)}$

b. 스케줄링

$2 \leq i \leq N_r$ 동안

i) S 에 남아있는 모든 안테나 채널 $\mathbf{h}_k, k=1, \dots, |S|$ 를 행렬 $\mathbf{H}_{CSS,i-1}$ 하나씩 쌓아서 (Stack) $|S|$ 개의 임시 행렬 $\mathbf{H}_{CSS,i,k} = [\mathbf{H}_{CSS,i-1} \ \mathbf{h}_k], k=1, \dots, |S|$ 를 생성한다.

ii) 각 임시 행렬에 대해서, 다음을 계산한다

$$R_{s,i,k} = \log_2 \det(\mathbf{I}_{N_r} + SNR \mathbf{H}_{CSS,i,k} \mathbf{H}_{CSS,i,k}^*),$$

$k = 1, \dots, |S|$

iii) 다음을 연산한다.

$$\mathbf{H}_{CSS,i} = \arg \max_{\mathbf{H}_{CSS,i,k}} R_{s,i,k}$$

이 결과로 선택된 $\mathbf{H}_{CSS,i}$ 에 해당하는 안테나 채널 \mathbf{h}_k 를 $\mathbf{h}_{CSS(i)}$ 로 표시한다.

$$S = S/\mathbf{h}_{CSS(i)}.$$

3.4 Semi-orthogonal 안테나 스케줄러 (SOAS: Semi-orthogonal Antenna Scheduler)

식 (6)에 나타난 AS 기반 시스템에서 스케줄러 q 에 의한 합 채널 용량을 다시 살펴보면 다음과 같은 등식이 성립한다.

$$R_s = \log_2 \det(\mathbf{I}_{N_r} + SNR \mathbf{H}_q \mathbf{H}_q^*) = \log_2 \det(\mathbf{I}_{N_r} + SNR \mathbf{H}_q^* \mathbf{H}_q) \quad (7)$$

이때, 행렬 $\mathbf{H}_q^* \mathbf{H}_q$ 의 대각 성분은 선택된 안테나 채널의 놈이고, 나머지 성분들은 서로 다른 두 안테나 채널의 내적 값이 된다. 식 (7)에 임의의 음의 값이 아닌(non-negative definite) 행렬 \mathbf{X} 에 대해 성립하는 하다마드 부등식 $\det(\mathbf{X}) \leq \prod_i \mathbf{X}_{i,i}$ 을 적용할 수 있다. 여기서 $\mathbf{X}_{i,i}$ 는 행렬 \mathbf{X} 의 (i,i) 번째 대각 성분이며 \mathbf{X} 가 대각 행렬일 때에 등식이 만족한다. 즉

$\mathbf{H}_q^* \mathbf{H}_q$ 의 대각 이외의 성분이 최대한 0에 가깝도록 안테나를 선택하면 채널 용량을 최대화 시킬 수 있고, 따라서 \mathbf{H}_q 의 안테나 채널들을 최대한 직교성을 가지도록 (semi-orthogonal) 선택하면 채널 용량이 최대화 될 수 있다. 따라서 이 스케줄러는 semi-orthogonal 안테나 스케줄러라고 불린다.

SOAS의 기본 동작은 간단한 그람-슈미트 (Gram-Schmidt) 직교화 방식에 기반한다.

a. 초기화

FS 나 SS 방식으로 스케줄링 집합 S 를 생성한다. 선택된 안테나의 집합 T 를 생성한다. 그람-슈미트 과정에서 형성되는 기저 벡터의 집합 G 를 생성한다.

b. 스케줄링

$|T| < N_R$ 일 때 다음 과정을 수행한다.

모든 안테나 채널 $\mathbf{h}_k \in S, k = 1, \dots, |S|$ 에 대해, 다음을 계산한다.

$|T|=0$ 일 때, $\mathbf{g}_k = \mathbf{h}_k$

$|T| \geq 1$ 일 때,

$$\mathbf{g}_k = \mathbf{h}_k - \sum_{\mathbf{g}_j \in G} \mathbf{g}_j^* \mathbf{h}_k \mathbf{g}_j / \|\mathbf{g}_j\|^2$$

계산한 \mathbf{g}_k 중 최대의 $\|\mathbf{g}_k\|^2$ 를 가지는 \mathbf{g}_{\max} 와 그에 해당하는 안테나 \mathbf{h}_{\max} 를 찾는다.

$$T = T \cup \mathbf{h}_{\max}, G = G \cup \mathbf{g}_{\max}, S = S / \mathbf{h}_{\max}$$

3.5 낮은 복잡도의 Semi-orthogonal 안테나 스케줄러 (RC-SOAS: Reduced Complexity Semi-orthogonal Antenna Scheduler)

3.4에서 제안한 SOAS는 그람-슈미트 방식을 사용해서 스케줄링 복잡도를 줄이는 방식이다. 하지만, 이 또한 N_R 개의 안테나가 선택될 때까지 스케줄링 집합 S 에 포함된 모든 안테나 채널에 대해 그람-슈미트 과정을 반복적으로 사용해야 하는 복잡도가 존재한다. 따라서, 이러한 복잡도의 추가적인 감소를 위해서 제안된 RC-SOAS는 먼저 채널 높이 가장 큰 N_R 개의 안테나를 먼저 선택하고 이렇게 선택된 안테나 채널에 대해서만 그람 슈미트 방식을 사용하여 서로 직교인 N_R 개의 기저 벡터를 생성한다. 이후 다시 스케줄링 집합 S 에 포함된 모든 안테나 채널에 대해서 생성된 기저 벡터와 가장 큰 상관도를 가진 N_R 개의 안테나 채널을 선택한다. 따라서, 그람-슈미트 방식은 초기화 단계에서만 사용되고, 이후 과정은 간단한 계산을 통해 스케줄링하게 된다. 다

음은 RC-SOAS 방식의 과정을 보여준다.

a. 초기화

FS 나 SS 방식으로 스케줄링 집합 S 를 생성한다.

스케줄링 집합 S 에서 가장 높이가 큰 N_R 개의 채널 벡터들인

$\{\mathbf{h}_{1,\max}, \mathbf{h}_{2,\max}, \dots, \mathbf{h}_{N_R,\max}\}$ 을 고른 뒤, 그람-슈미트 방식을 사용해서 N_R 개의 기저 벡터 $\{\mathbf{g}_{1,\max}, \mathbf{g}_{2,\max}, \dots, \mathbf{g}_{N_R,\max}\}$ 를 구한다.

선택되는 안테나 채널의 집합이 될 공집합 T 를 생성한다.

b. 스케줄링

$1 \leq i \leq N_R$ 에 대해서

스케줄링 집합 S 에 속한 모든 벡터 $\mathbf{h}_k, k = 1, \dots, |S|$ 에 대해서, $\mathbf{g}_k^* \mathbf{h}_k$ 를 구한 다음 그 값이 가장 큰 $\mathbf{h}_{i,\max}$ 를 구한다.

$$T = T \cup \mathbf{h}_{i,\max}, S = S / \mathbf{h}_{i,\max}$$

IV. 성능 비교

본 논문에서 스케줄링 알고리즘의 성능 비교는 세 가지 기준, 즉 복잡도, 채널 용량 성능, SER 링크 성능을 기반으로 이루어진다.

4.1 복잡도 비교

스케줄링 알고리즘의 복잡도를 판단하는 기준은 상당히 다양하게 생각할 수 있는데, 본 논문에서의 복잡도 성능이 포함하는 범위는 실제 구현 관점이 아닌 알고리즘 계산에 관련된 연산의 복잡도로 한정한다. 즉, 행렬 곱셈, 행렬식 연산, 조합생성, 정렬의 복잡도로 한정하며 절대적인 복잡도가 아닌 연산 복잡도의 차수를 이용하여 상대적 성능을 비교한다. 또한, FS 방식과 SS 방식의 복잡도 비교를 통해 스케줄링 집합의 크기를 줄이는 것이 어떤 상황에 이득을 주는 지에 대한 고찰을 할 것이다.

먼저 BFS의 복잡도는 주어진 채널들을 이용해서 가능한 모든 안테나 조합을 생성해야 하므로, 조합 생성 복잡도가 가장 큰 영향을 미치게 된다. 위의 복잡도 기준을 적용한 BFS의 복잡도는 FS 방식의 경우 $O(N_R^3 (KN_T)^{N_R})$ 로, SS 방식의 경우 $O(N_R^3 (K)^{N_R})$ 로 주어진다. 따라서, 수신 안테나의 개수가 증가함에 따라, 즉 동시에 전송하는 사용자가 많아질 수록 스케줄링 복잡도는 지수적으로 증가하게 되고, 스케줄링의 대상이 되는 사용자의 수가 많아지는 경우

도 복잡도가 크게 증가하게 된다.

MSS의 복잡도는 3.2장에서 살펴본 바와 같이 채널의 놈 계산과 정렬의 복잡도에만 관련있다. 가장 복잡도가 낮은 정렬 방식을 사용하는 경우 MSS의 전체 복잡도는 FS의 경우 $\alpha(KN_T)$, SS의 경우 $\alpha(K)$ 로 매우 낮음을 볼 수 있다.

CSS의 복잡도는 스케줄링 과정에서 생성된 임시 행렬들의 채널 용량 계산과 정렬의 복잡도가 결정한다. 이 경우 FS의 복잡도는 $\alpha(N_R^3(KN_T-1))$ 가 되고, SS인 경우는 $\alpha(N_R^3(K-1))$ 의 복잡도를 가진다.

SOAS의 경우 그람-슈미트 연산에 필요한 벡터 곱과 정렬에 관련된 연산이 복잡도를 결정한다. 이를 바탕으로 계산한 복잡도는 FS의 경우 $\alpha(N_R(KN_T-1)^2)$ 이 되고, SS는 $\alpha(N_R(K-1)^2)$ 의 복잡도를 가진다.

RC-SOAS의 경우 최초의 N_R 개 만큼의 벡터를 고르는 정렬복잡도 및, 그람-슈미트 이후의 내적 계산 및 정렬의 복잡도가 전체 복잡도를 결정한다. 하지만, 이중 그람-슈미트 연산의 복잡도가 전체 연산 복잡도의 가장 큰 부분을 차지한다. 따라서, 스케줄링 집합의 크기와 상관없이 N_R 개의 벡터를 이용해서 그람-슈미트 연산을 하므로 FS와 SS 모두 $\alpha(N_R^3)$ 의 복잡도를 가진다.

표 1에서 지금까지 언급된 스케줄링 방식들의 복잡도를 정리하였다. BFS의 경우 4.2장에서 다시 언급하겠지만 채널 용량이나 SER 성능에서 가장 우수한 반면 나머지 네 가지 스케줄러에 비해 송수신 안테나와 사용자의 수가 증가할 때 월등히 높은 복잡도를 보인다. MSS는 채널 용량과 SER 성능이 가장 안 좋은 반면 사용자 수나 송신안테나 수에 대해 복잡도가 선형적으로 증가하며, 수신 안테나 수이자 동시에 전송되는 사용자 수인 N_R 과는 무관한 복잡도를 보임을 알 수 있다.

표 1. 본 논문의 스케줄러의 복잡도 비교

	FS 방법	SS 방법
BFS	$\alpha(N_R^3(KN_T)^{N_R})$	$\alpha(N_R^3(K)^{N_R})$
MSS	$\alpha(KN_T)$	$\alpha(K)$
CSS	$\alpha(N_R^3(KN_T-1))$	$\alpha(N_R^3(K-1))$
SOAS	$\alpha(N_R(KN_T-1)^2)$	$\alpha(N_R(K-1)^2)$
RC-SOAS	$\alpha(N_R^3)$	$\alpha(N_R^3)$

제안된 스케줄러 중 가장 낮은 복잡도를 보이는 방식은 RC-SOAS 이다. 각 사용자의 송신안테나 수나 사용자의 수는 복잡도에 큰 영향을 주지 못한다. 수신 안테나 수에 복잡도가 비례하지만 지수가 3으로 고정되어 있기 때문에 BFS같은 급격한 증가는 보이지 않는다. 주의할 점은 SS의 경우 분명히 FS 보다는 적은 복잡도를 보인다는 점이다. 다만, 주요 과정의 복잡도만 FS와 SS에서 $\alpha(N_R^3)$ 로 동일하게 유지되기 때문에 표 1에서는 동일한 복잡도로 간주된 것이다. 또한, 연산 복잡도의 차수만으로 RC-SOAS가 가장 낮은 복잡도의 방식이라고 절대적으로 단정할 수는 없다.

CSS와 SOAS를 비교해보면, CSS의 경우 수신 안테나 수에 대해서는 상대적으로 높은 복잡도를 보이고, SOAS는 사용자 수나 송신 안테나 수가 많아질 때의 복잡도가 높음을 알 수 있다. 절대적으로 SOAS와 CSS의 복잡도 비교를 할 수는 없지만, 사용자나 송신 안테나 수가 많은 경우 CSS가, 수신 안테나 수가 많은 경우 SOAS가 상대적으로 복잡도 면에서 이득임을 알 수 있다. 두 방식 모두 BFS보다는 현저히 낮은 복잡도를 가진다.

표 1에 의해 RC-SOAS 를 제외한 모든 스케줄링 방식이 FS에 비해서 SS 방식이 낮은 복잡도를 가짐을 알 수 있다. 특히 CSS의 경우 SS에 의해 많은 복잡도의 향상을 가져오지는 않지만, 같은 성능을 유지하면서 추가적으로 복잡도가 줄어든다면 시스템 구현 측면에서는 의미가 있다고 볼 수 있다. SS 방식이 알맞은 경우는 채널 용량과 SER 성능을 분석하면서 다시 언급할 것이다.

4.2 채널 용량 및 SER 비교

그림 2는 평균 수신 SNR이 5dB인 경우의 각 스케줄러 기법의 채널 용량 CDF를 보여준다. 기지국과 각 사용자는 각각 2개의 안테나를 가지고 있으며, 총 10명의 사용자가 존재한다. 이 그림에서 가장 먼저 주목할 점은 FS와 SS가 같은 성능을 보인다는 점이다. 가장 왼쪽의 겹쳐있는 두 선은 가장 복잡도가 낮은 스케줄러인 MSS의 FS와 SS 방식의 성능을 나타낸다. MSS가 복잡도는 가장 낮지만, 성능 또한 다른 방식에 비해 떨어짐을 알 수 있다. 오른쪽에는 8개의 선이 같은 위치에 겹쳐있는데, MSS를 제외한 나머지 기법들의 성능이 FS나 SS방식 모두 BFS-FS 방식과 같은 성능을 보임을 알 수 있다. 따라서, 제안한 세 가지 기법 모두 낮은 복잡도로 최적의 스케줄러인 BFS-FS와 같은 성능을 보

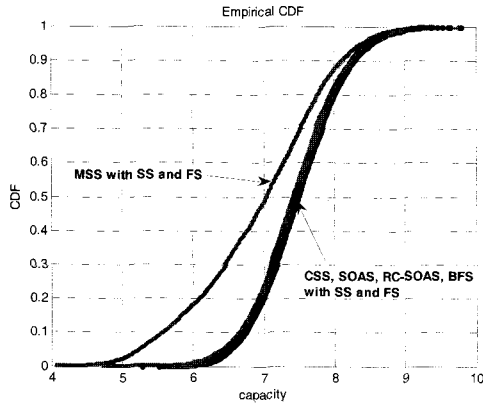


그림 2. 송수신 안테나가 2개, 수신 SNR이 5dB이고 10명의 사용자들이 존재할 때 스케줄러들의 채널 용량 CDF

인다는 점을 알 수 있다. 특히 제안된 기법 중 RC-SOAS 방식이 상대적으로 효과적임을 알 수 있다. 또한 CSS 나 SOAS의 경우는 SS 방식을 쓰더라도 성능 열화가 없다.

그림 3은 FS와 SS 방식의 성능차이를 확인하기 위한 그림이다. 평균 수신 SNR은 역시 5dB로 가정되었고, 안테나 수와 사용자 수는 변수로 가정되어 있다. 먼저 사용자 수에 따른 성능차이를 보기 위해 가장 왼쪽의 두 선을 보자. 두 선중 왼쪽의 선은 8개의 선이 겹쳐있는 것으로, MSS를 제외한 4가지 기법이 송수신 안테나가 2개이고 10명의 사용자가 있을 때, FS와 SS 인 경우 동일한 성능을 보임을 나타낸다. 오른쪽 선은 안테나 수는 고정되어있고, 사용자의 수만 20명으로 증가시킨 경우인데, 모든 기법들이 여전히 동일한 성능을 보임을 알 수 있다. 이를 BFS 기법을 적용할 때 FS와 SS 방식에서 고

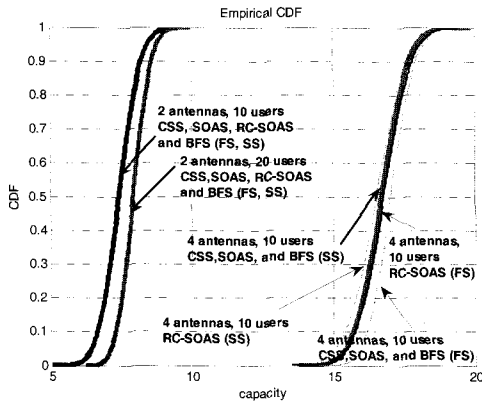


그림 3. 송수신 안테나가 2개, 수신 SNR이 5dB이고 10명의 사용자들이 존재할 때 스케줄러들의 채널 용량 CDF

려해야하는 안테나 조합의 수로 비교하면, 사용자가 10명일 때 SS 방식에서의 조합 수와 FS 방식에서의 조합 수의 비를 구하면 $C_2^{10}/C_2^{20} = 45/190 \approx 23.7\%$ 가 되고, 사용자가 20명일 때는 그 비가 $C_2^{20}/C_2^{40} = 190/780 \approx 24.6\%$ 가 되어서 그 비율이 거의 비슷하다. 이때, C_a^b 는 a 개에서 b 개를 고르는 조합의 수를 나타낸다. 따라서, 사용자가 10명일 때의 경향이 20명일 때의 성능 경향과 크게 다르지 않음을 예상할 수 있다. 오른쪽의 선들은 사용자의 수는 10명으로 고정시킨 상태에서 안테나의 수를 송수신 단 모두 4개로 증가시켰을 때의 결과를 나타낸다.

RC-SOAS를 제외한 CSS, SOAS, BFS의 경우는 일단 SS 인 경우와 FS인 경우 각각에서는 같은 성능을 보인다. 하지만, 안테나가 2개인 경우에는 다르게 FS 방식이 SS 방식보다 우수한 성능을 보인다. 이를 앞서 본 조합 수의 비교를 통해 살펴보면, SS와 FS의 조합의 비가 $C_4^{10}/C_4^{40} = 210/91390 \approx 0.23\%$ 로 안테나가 2개인 경우에 비해서 그 비가 현저히 낮다. 이는 SS 방식으로는 전체 가능한 안테나 조합의 극소수만 검토한다는 뜻이고, 따라서 FS 방식이 우위를 보이게 된다. 이를 바탕으로 FS와 SS는 사용자 수의 변화에 대해서는 크게 차이가 나타나지 않지만, 안테나의 수가 많아지면 FS의 성능이 우위에 있다고 볼 수 있다. 따라서, SS 방식을 통한 추가적인 복잡도의 감소는 안테나의 수가 적은 시스템에 알맞다. 다중 안테나의 구현 가능성이 안테나 수가 적을 때에 용이하다고 가정한다면 알맞은 SS 방식은 실제 시스템에서의 추가적인 복잡도의 감소에 기여할 수 있다. RC-SOAS 기법의 경우는 안테나의 수가 4개로 증가했을 때, CSS나 SOAS에 비해서 성능 열화가 있고, 특히 RC-SOAS-FS의 경우는 CSS-SS, SOAS-SS의 경우와 거의 비슷한 성능을 보인다. 이는 RC-SOAS가 처음 결정하는 기저 벡터가 전체 안테나를 고려해서 결정되지 않고, 가장 높은 N_R 개의 안테나만을 사용해서 계산하기 때문이다. 반면 유사한 알고리즘을 사용하는 SOAS의 경우 매번 전체 안테나를 대상으로 가장 직교성이 큰 안테나를 찾아 기저 벡터를 계산하기 때문에, 안테나의 조합의 수가 많아질 수록 SOAS가 RC-SOAS보다 우수한 성능을 보이게 된다.

그림 4에서 6가지는 제안된 기법들의 SER 성능을 보여준다. 수신기는 MMSE-SIC 구조를 적용하기 위해 MMSE-VBLAST 기법을 사용하였다. 그림 4는 송수신 안테나가 2개씩 있고, 사용자가 6명,

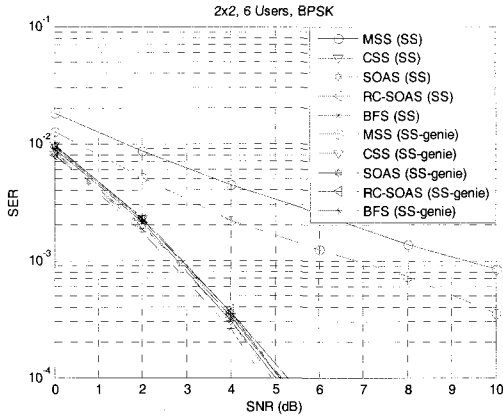


그림 4. 송수신 안테나가 2개, 6명의 사용자가 존재하고 스케줄링 집합이 SS 방식으로 형성되었을 때 BPSK 변조를 이용한 MMSE-VBLAST SER 성능

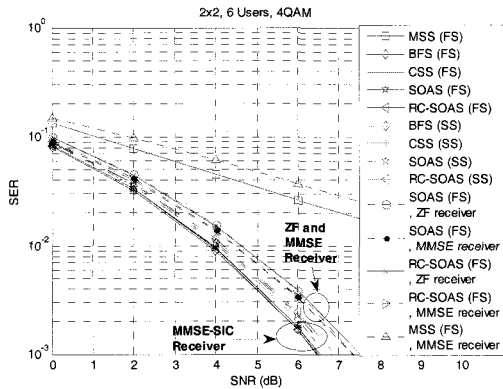


그림 5. 송수신 안테나가 2개, 6명의 사용자가 존재하고 QPSK(4QAM) 변조를 이용한 MMSE-VBLAST 및 ZF, MMSE 수신기의 SER 성능

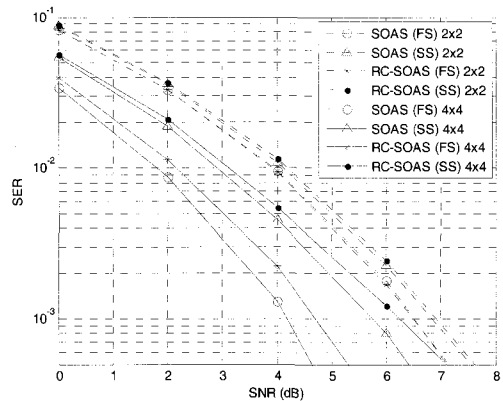


그림 6. 송수신 안테나가 2개 또는 4개, 6명의 사용자가 존재하고 QPSK 변조를 이용한 SOAS와 RC-SOAS의 MMSE-VBLAST SER 성능

BPSK 변조가 사용되었을 때의 SER 성능이다. 스케줄링 집합은 SS를 사용한 것으로 가정했다. MSS의 경우는 채널 높이 큰 안테나만 선택했기 때문에, 각각의 안테나의 채널 상황은 좋지만, 선택된 안테나끼리의 상관도는 커지게 된다. 따라서 MMSE-VBLAST의 간섭 제거 과정에서 오류 전파 (error propagation)의 영향을 받게 된다. 그림 4에서 'genie' 라고 표시되어있는 선들은 오류 전파가 존재하지 않는 이상적인 경우인데, MSS의 경우 genie인 경우와의 성능차가 2dB 가량 되어 오류 전파의 영향을 많이 받는다는 것을 알 수 있다. 반면 CSS와 SOAS, RC-SOAS의 경우는 기본적으로 BFS와 거의 차이가 없는 우수한 성능을 보이며, genie의 경우와도 거의 일치하는 성능을 보여 오류 전파의 영향이 거의 없음을 알 수 있다. 이는 높은 채널 용량을 보이는 채널 조합의 특성이 안테나 채널끼리의 간섭이 적음을 간접적으로 보여주는 것으로 이는 SOAS 기법의 목적과도 부합한다.

그림 5는 그림 4와 같은 조건하에 QPSK 변조를 적용한 경우의 성능이다. 이 그림에서는 FS와 SS 기법의 성능을 같이 도시하였다. 그림 4와 유사하게 MSS의 경우 FS와 SS의 성능이 약 1.5dB 정도 차이를 보임을 알 수 있다. 하지만 나머지 기법들의 경우 FS와 SS의 성능차이가 0.5 dB 이내임을 알 수 있고, 모두 BFS의 성능에 거의 근접함을 볼 수 있다. 이 그림에서는 서로 유사한 알고리즘을 가지는 SOAS와 RC-SOAS에 대해서 MMSE-VBLAST 수신기 뿐 아니라 선형 수신기인 ZF(Zero Forcing)과 MMSE 수신기의 성능도 도시하였다. ZF과 MMSE 수신기를 적용한 경우도 MMSE-VBLAST 수신기의 성능에 비해 약 1dB정도만 열화 됨을 볼 수 있는데, 이는 안테나 채널의 직교성으로 인해 각 스트림간의 간섭이 줄어들고, 따라서 SIC 수신단의 간섭제거의 효과가 크지 않음을 보여준다. 즉, 수신기의 구현 복잡도 측면에서도 제한된 스케줄러를 사용하면 비선형 SIC 수신기 대신 간단한 선형 수신기를 적용해도 성능이 크게 열화되지 않음을 알 수 있다.

그림 6은 그림 4, 5와 다르게 안테나의 수를 4개로 증가시킨 성능을 함께 도시하였다. 스케줄러는 SOAS와 RC-SOAS로 한정하였고, QPSK 변조가 사용되었다. 그림 3에서 살펴보았듯이, 송수신 안테나가 2개씩인 경우, FS와 SS 그리고 SOAS와 RC-SOAS의 성능차가 크지 않다. 하지만 안테나의 수가 증가하면서 FS와 SS의 SER성능이 1dB이상

차이점을 볼 수 있으며, 특히 RC-SOAS의 성능이 SOAS에 비해 열화됨을 볼 수 있다. 이는 그림 3의 채널 용량 성능처럼 안테나의 수가 증가하는 경우 SS 방식과 RC-SOAS 기법의 성능이 SER 측면에서도 열화됨을 보여준다. 따라서, SER 관점에서 안테나의 수가 증가하게 되면 SS 보다는 FS 방식을 사용하는 것이 이득이고, 또한 RC-SOAS보다는 SOAS나 CSS 기법이 효과적임을 알 수 있다.

V. 결론

본 논문에서는 안테나 선택 기반의 상향링크 다중사용자 MIMO/FDD 시스템에서 낮은 복잡도로 최적의 성능을 보이는 세 가지 발견적(heuristic) 스케줄러를 제안하고 복잡도, 채널 용량, SER 관점에서 성능을 비교하였다. 제안된 세 가지 스케줄러 중, CSS와 SOAS 기법은 낮은 복잡도로도 가장 복잡하면서도 성능이 우수한 BFS의 성능과 일치함을 보였고, RC-SOAS 기법은 안테나의 수가 작은 시스템에서 BFS의 성능과 일치함을 확인하였다. 또한 SS 방식으로 스케줄링의 후보가 되는 안테나 집합인 스케줄링 집합을 형성함으로써 송수신단 안테나가 적은 상황에서 성능의 손실 없이 추가적인 복잡도의 감소 효과를 얻을 수 있다. 실험 결과, 안테나 수가 2개로 적은 상황에서는 RC-SOAS 방식이, 안테나 수가 늘어나는 경우는 SOAS 또는 CSS 방식이 상대적으로 효율적임을 보였다.

참고 문헌

[1] S. Vishwanath, "Multiple-user Multiple Antenna Systems: Theoretical Limits and Practical Strategies," Ph. D. dissertation, Stanford Univ, 2003.

[2] P. Viswanath, D. N. Tse, "Sum capacity of the multiple antenna Gaussian broadcast channel," in Proceedings of Int. Symp. Inform. Theory, p.497, June 2002.

[3] W. Yu, W. Rhee, S. Boyd, and J. M. Cioffi, "Iterative Water-Filling for Gaussian Vector Multiple-Access Channels," IEEE Tran. on Inform. Theory, Vol. 50, No. 1, January 2004.

[4] H. Weingarten, Y. Steinberg, and S. Shamai, "The capacity region of the Gaussian MIMO broadcast channel," ISIT 2004, 2004.

[5] S. Serbetli, and A. Yener, "Time-slotted Multiuser MIMO systems: Beamforming and Scheduling Strategies," EURASIP Jour. on Wire. Comm. and Net., vol 2, pp.286-296, 2004.

[6] M. Sanchez-Fernandez, M. L. Pablo-Gonzalez, and A. Lozano, "Exploiting Multiuser Diversity through Uplink Scheduling," VTC spring, 2005.

[7] D.N. Tse, and P. Viswanath, "Fundamentals of Wireless Communication," Cambridge Press, 2005.

[8] E. Telatar, "Capacity of multi-antenna Gaussian channels," European Transactions on Telecommunication, Vol. 10, pp.585-596, Nov. 1999.

김요한 (Yohan Kim)

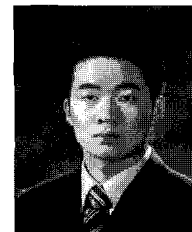
정회원



2002년 2월 연세대학교 기계전자공학부 졸업
 2004년 2월 연세대학교 전기전자공학과(석사)
 2004년 3월~현재 연세대학교 전기전자공학과(박사과정)
 <관심분야> 정보이론, 다중사용자 MIMO 시스템

조성윤 (Sungyoon Cho)

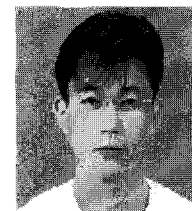
준회원



2006년 2월 연세대학교 전기전자공학과 졸업
 2006년 3월~현재 연세대학교 전기전자공학과 석사과정
 <관심분야> 이동통신, 다중사용자 MIMO 시스템 및 스케줄링

이택주 (Taekju Lee)

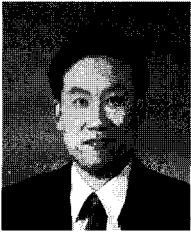
준회원



2000년 2월 한양대학교 전기전자공학과 졸업
 2000년 3월~현재 LG 전자 MC 연구소 선임연구원
 2006년 3월~현재 연세대학교 전기전자공학과 석사과정
 <관심분야> 이동통신, MIMO 송수신기 설계

김 동 구 (Dongku Kim)

정회원



1983년 2월 한국항공대학교 통신
공학과 졸업

1985년 U.S.C. Dept. of Electrical
Engineering(석사)

1992년 U.S.C. Dept. of Electrical
Engineering(박사)

1999년~현재 연세대학교 공과대

학 전기전자공학과 교수

<관심분야> CDMA 이동통신, 다중접속 기술, 변조 및
채널 부호화 기술, 스케줄링/MAC기술, 순방향링크
빔성형 기술 및 MIMO기술, UWB, Binary CDMA