간섭 제한적인 MIMO 환경에서의 협력적인 제한적 피드백 프리코딩

준회원 윤 정 민*, 종신회원 이 종 호**, 정회원 곽 영 우*, 준회원 최 정 식*, 종신회원 김 성 철*

Cooperative Limited Feedback Precoding in Interference-Limited MIMO Networks

Jung-min Yoon* *Associate Member*, Jong-ho Lee** *Lifelong Member*, Young-woo Kwak* *Regular Member*, Jeong-sik Choi* *Associate Member*, Seong-cheol Kim* *Lifelong Member*

요 약

본 논문에서는 한정된 자원에 여러 개의 연결이 혼재해 있는 상황에서 협력적으로 제한적인 피드백을 사용하여 성능을 개선시킬 수 있는 방법을 제안한다. 제안하는 방법은, 코드북에 있는 각 프리코더의 성능 우선순위에 따라 가중치를 부여하여 각 연결로부터 피드백 받은 가중치 데이터로부터 가중치의 합이 가장 높은 프리코더를 선택하는 방식이다. 시뮬레이션 결과 기존의 비협력적 방식에 비해 전송도 뿐만 아니라 시스템의 안정성에 관여되는 프리코더 선택의 수렴 정도 또한 향상됨을 알 수 있다.

Key Words : MIMO, limited Feedback precoding, Interference, codebook selection

ABSTRACT

In this paper, we propose new cooperative precoder selection technique for interference limited MIMO networks. Our proposed method gives weighting to precoders in the codebook according to each precoder's performance priority. By applying our proposed method to precoder selection sequence, performance of entire system can be improved in terms of sumrate, stability, and feedback rate.

I. 서 론

최근의 통신기술 연구는 폭발적인 통신자원 사용자의 증가를 어떻게 해결할 것인가와 초점을 두고 있다. Multiple-Input Multiple-Output (MIMO) 기술은 그 자원사용적인 측면에 있어서의 효율성과 확장성으로 인해 연구의 대상이 되어왔고, 최근에는 이에 더해 통신 수요자의 증가로 인한 다중 사용자 MIMO 환경에서 발생할 수 있는 상호간섭의 완화에 대한 문제가 새로운 연구주제로서 대두되고 있다[1-3].

한정된 자원안의 여러 사용자가 효율적인 통신을 하기 위해서는 상호간의 더 많은 정보를 필요로 하게 된다. 따라서 이런 다중 사용자 환경에서는 피드백에 의한 성능개선이 더욱 부각되게 되었고, 제한적인 피드백 기법은 적은 피드백 양으로도 효율적으로 성능개선을 추구할 수 있는 방법으로서 주목받고 있다.

대표적인 제한적 피드백의 기법중의 하나로서 Grassmanian precoder가 있다[4,5]. 이는 생성되어 있는 프리코더의 목록으로부터 현재의 채널 이득을 가장 극대화 할 수 있는 프리코더의 인덱스를 피드백 하

※ 이 논문은 2010년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(2010-0000844, 2010-0003017).
* 서울대학교 전기공학부, 뉴미디어통신공동연구소 ({jungminy9, limpbiz, sic722, sckim}@maxwell.snu.ac.kr)
** 공주대학교 전기전자제어공학부 (jongholee@kongju.ac.kr)
논문번호 : KICS2010-12-629, 접수일자 : 2010년 12월 27일, 최종논문접수일자 : 2011년 4월 15일

는 기법으로서, 적은 피드백으로 많은 성능 개선을 할 수 있는 방법으로 널리 알려져 있다.

하지만 이는 사용자간의 간섭이 존재하지 않는 단일 사용자일 때의 한정적인 기법으로, 한정된 자원 안에 여러 개의 연결이 혼재해 있는 상황에서는 각 사용자가 선택한 프리코더에 의한 실제 간섭채널의 변화로 인하여, 분산적인 방식으로는 간섭을 고려한 프리코더 선택에 어려움이 따른다는 단점이 있다[6,7].

본 논문에서는 한정된 자원에 여러 개의 연결이 혼재해 있는 상황에서 협력적으로 제한적인 피드백을 사용하여 성능을 개선시킬 수 있는 방법을 제안한다. 제안하는 방법은 사용하는 코드북에 있는 각 프리코더에 따른 성능에 따라 가중치를 주는 방식으로서, 기존의 비협력적 방식에 비해 전송도 뿐만 아니라 시스템의 안정성에 관여되는 수렴 정도 또한 향상시킬 수 있다. 또한 추가적으로 성능의 표준편차를 이용한 가중치 변수를 도입하여, 제안하는 방식의 피드백 양을 줄이면서도 변화하는 개별채널에 대한 시스템의 프리코더 선택 및 성능에 대한 안정성을 높일 수 있는 방법을 제안한다.

Ⅱ. 시스템 모델

본 논문에서는 제한적인 피드백 프리코딩을 사용하는 간섭 제한적인 다중안테나 시스템을 가정한다.

그림 1은 시스템 모델을 나타낸다. 각 연결의 송신단과 수신단은 P개의 안테나를 사용하여 송수신하며, 송신단에서는 매 시간 마다 N개의 데이터 스트림을 $P\times N$ 코드북을 사용하여 전송한다. 코드북에 존재하는 프리코더의 개수를 L이라 할 때, 사용 가능한 코드북은 $\boldsymbol{V}=\{v(1),v(2),\cdots,v(L)\}$로서 나타낼 수 있다. 시스템 내에 총 K개의 송수신 연결이 있다고 가정할 때, k번째 연결의 $P\times 1$ 수신신호 벡터 $\boldsymbol{y}_k$는 다음과 같이 나타낼 수 있다[8].

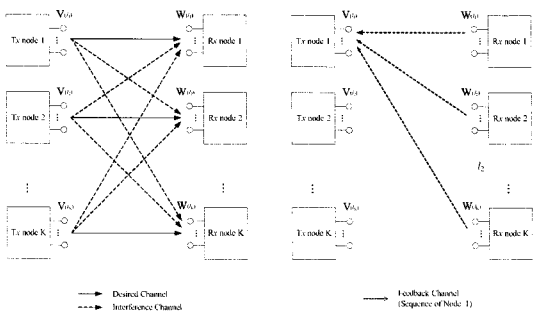


그림 1. 시스템 모델

$$\boldsymbol{y}_k=\boldsymbol{H}_{k,k}v(l_k)\boldsymbol{s}_k+\sum_{\substack{m=1\\ m\neq k}}^{K}\boldsymbol{H}_{k,m}v(l_m)\boldsymbol{s}_m+\boldsymbol{n}_k \quad (1)$$

$\boldsymbol{H}_{k,m}$은 m번째 연결의 송신단으로부터 k번째 연결 수신단으로의 $P\times P$ 채널 행렬로서 $CN(0,1)$의 복소 Gaussian 분포를 가진다. $v(l_m)$은 m번째 연결이 코드북에서 선택한 $P\times N$ 프리코더 행렬이며, l_m은 m번째 연결이 선택한 코드북의 인덱스를 나타내고, 코드북이 L개로 구성되어 있으므로 인덱스는 $1<l_m<L$의 범위를 가진다. $\boldsymbol{s}_m$은 m번째 연결 송신단의 $N\times 1$ 심볼 스트림 벡터이고, $\boldsymbol{n}_k$는 k번째 연결 수신단의 $P\times 1$ 수신 잡음 벡터로서 $CN(0,\sigma^2)$의 복소 Gaussian 분포를 가진다. 이 때 $\boldsymbol{H}_{k,m}v(l_m)$은 실질적인 채널로서 $\overline{\boldsymbol{H}_{k,m}}(l_m)=\boldsymbol{H}_{k,m}v(l_m)$와 같이 표현할 수 있고, 수신 신호 벡터 (1)은 다음과 같이 표현할 수 있다.

$$\boldsymbol{y}_k=\overline{\boldsymbol{H}_{k,k}}(l_k)\boldsymbol{s}_k+\sum_{\substack{m=1\\ m\neq k}}^{K}\overline{\boldsymbol{H}_{k,m}}(l_m)\boldsymbol{s}_m+\boldsymbol{n}_k \quad (2)$$

수신기에는 간섭을 고려한 MMSE 수신기를 사용하였다[9,10].

$$\boldsymbol{w}(l_k)=\overline{\boldsymbol{H}_{k,k}}(l_k)^{\dagger}\left[\sum_{m=1}^{K}\overline{\boldsymbol{H}_{k,m}}(l_m)\overline{\boldsymbol{H}_{k,m}}(l_m)^{\dagger}+N\sigma^2\boldsymbol{I}_P\right]^{-1} \quad (3)$$

$\boldsymbol{w}(l_k)$는 $N\times P$ 수신신호 행렬을 나타내며, $\boldsymbol{I}_P$는 $P\times P$ 단위행렬을 나타낸다. $\dagger$는 행렬의 pseudo inverse를 나타내며, $[\ ]^{-1}$은 역행렬을 나타낸다.

주어진 시스템에서 k번째 연결의 채널용량 R_k는 각 심볼 스트림의 채널 용량의 합으로서, 다음과 같이 나타낼 수 있다.

$$R_k=\sum_{i=1}^{N}\log_2(1+SINR_{i,k,l_k}) \quad (4)$$

여기에서 수신기를 고려했을 때의 k번째 연결 수신단의 i번째 심볼 스트림의 수신 후 $SINR$은 다음과 같다.

$$SINR_{i,k,l_k}=\frac{\left|\boldsymbol{w}(l_k)\overline{\boldsymbol{H}_{k,k}}(l_k)\right|^2_{(i,i)}}{\|\boldsymbol{w}(l_k)\|_i^2N\sigma^2+\boldsymbol{T}_{i,k,m}+\boldsymbol{D}_i} \quad (5)$$

$T_{i,k,m}$, D_i는 각각 m번째 연결로부터 k번째 연결의 i번째 심볼 스트림으로의 간섭, k번째 연결의 자기 수신신호 내에서 다른 심볼로부터 i번째 심볼로의 간섭을 나타내는 항으로서 다음과 같이 표시할 수 있다.

$$T_{i,k,m}=\sum_{m\neq k}\| w(l_k)\overline{H_{k,m}}(l_m)\|_i^2 \quad (6)$$

$$D_i=\sum_{j\neq i}|w(l_k)\overline{H_{k,k}}(l_k)|_{(i,j)}^2 \quad (7)$$

$\| \ \|_i$, $|\ |_{(i,j)}$ 은 각각 행렬의 i행의 norm, (i,j) 번째 원소의 절대값을 의미한다.

Ⅲ. 분산적인 제한적 피드백 프리코딩

여기서는 소단원에 관한 내용을 간단히 살펴본다. 여기서는 소단원에 관한 내용을 간단히 살펴본다.

3.1 최적화된 프리코더의 탐색

모든 연결의 수신단이 각자 모든 연결의 채널 정보를 가지고 있다고 가정하고 소모적인 방법을 통하여 각자 연결의 성능이 아닌 시스템 전체 성능에 최적화된 프리코더의 조합을 찾는다고 할 때, 그 조합과 시스템 전체의 채널용량은 다음과 같이 나타낼 수 있다[8].

$$(l_1^*, l_2^*, \cdots, l_K^*)=\arg max\sum_{k=1}^{K}\sum_{i=1}^{N}\log_2(1+SINR_{i,k,r_k}) \quad (8)$$

$$R_{total}=\sum_{k=1}^{K}\sum_{i=1}^{N}\log_2(1+SINR_{i,k,l_k^*}) \quad (9)$$

위에서 *는 최적화 시켰을 때의 프리코더를 나타내고, r_k는 프리코더 인덱스의 임시적인 변수를 나타낸다. 이런 중앙 집중화된 방식은 매우 비현실적이고 복잡하며 구현에 많은 어려움이 따르며, 제한적인 피드백을 사용하는 의의에도 맞지 않으므로 시스템 내에서 얻어낼 수 있는 성능의 이론적인 상한의 의미로 보는 것이 적당하다.

3.2 분산적인 방식을 이용한 프리코더의 탐색

간섭을 고려하지 않았을 경우, 각 연결은 SNR을 기준으로 코드북에서 프리코더를 선택하게 된다. 하지만 다수의 연결이 혼재하는 환경에서는 식 (7), (8)에서와 같이 실제 채널용량이 간섭에 의해 바뀌게 되므로 이를 고려하여 적절한 프리코더를 선택하여야 한다. 앞서 언급했던 바와 같이, 다른 연결에서 선택한 프리코더에 따른 실제 간섭채널의 변화는 k번째 연결의 성능과 최적화된 코드북의 선택에 영향을 주게 되고, 이런 변화된 간섭을 고려하여 선택한 k번째의 프리코더 또한 다른 연결로의 실제 간섭채널을 변화시키므로, 다른 연결의 최적화된 프리코더 선택에 영향을 주게 되기 때문이다. 따라서 이런 실제 간섭채널의 변화를 최대한 반영해 주기 위해서는 k번째 연결을 제외한 다른 모든 연결의 업데이트 상황을 바탕으로 반복적으로 최적화된 코드북을 탐색하고 업데이트 해주어야 한다.

먼저 각 연결이 SNR을 바탕으로 프리코더를 선택한 다음 현재의 실제 채널 정보를 알려주는 초기화 순서를 마친 후, $m\neq k$인 다른 연결의 직전까지의 업데이트 상황을 바탕으로 변화된 간섭을 고려하여 k번째 연결의 업데이트를 하게 된다. 다수의 연결이 존재할 경우 일반화된 수식은 다음과 같이 나타낼 수 있다[8].

$$l_k^{[x]}=\arg max\sum_{i=1}^{N}\log_2(1+SINR_{i,k,r_k}^{[x]})\Big|_{(1<r_k<L)} \quad (10)$$

$$SINR_{i,k,r_k}=\frac{|w(r_k)\overline{H_{k,k}}(r_k)|_{(i,i)}^2}{\| w(r_k)\|_i^2 N\sigma^2+Q^{[x]}(r_k)+Q^{[x-1]}(r_k)+D_i} \quad (11)$$

$$Q^{[x]}(r_k)=\sum_{m=1}^{k-1}\| w(r_k)\overline{H_{k,m}}(l_m^{[x]})\|_i^2 \quad (12)$$

$$Q^{[x-1]}(r_k)=\sum_{m=k+1}^{K}\| w(r_k)\overline{H_{k,m}}(l_m^{[x-1]})\|_i^2 \quad (13)$$

x는 참여하고 있는 연결이 모두 업데이트를 한 번씩 마쳤을 때의 전체 업데이트의 반복 횟수를 나타낸다. 시스템 내의 모든 연결이 순차적으로 한 번씩 업데이트를 마치면 다시 한 번 같은 식으로 업데이트를 시행하게 된다. $Q^{[x]}(r_k)$와 $Q^{[x-1]}(r_k)$는 업데이트 순서에 도달한 k번째 연결이 바로 직전까지 업데이트 된 정보를 바탕으로 간섭을 고려하여 업데이트함을 의미한다. 이런 순차적인 업데이트를 설정한 수렴회수에 도달할 때까지 반복하게 된다. 하지만 위와 같은 분산적인 방식은 다른 연결이 선택한 프리코더에 의해 k번째 연결이 받는 영향은 반영할 수 있으나, k번째 연결이 선택하는 프리코더가 다른 연결에 주는 영향은 반영할 수 없으며, 또한 이 때문에 시스템 전체

에서 볼 때 선택한 프리코더가 어느 한 프리코더로 수렴하지 못하여 시스템이 불안정해질 수 있다는 단점이 있다.

Ⅳ. 협력적인 제한적 피드백 프리코딩

앞서 언급한 바와 같이 위와 같은 분산적인 업데이트 방식의 문제점은 k번째 연결이 프리코더를 선택할 때 현재의 상황에서 자신의 성능을 극대화 하는 것에 초점을 두기 때문에, 선택한 프리코더에 의해 발생할 수 있는 다른 연결로의 영향을 전혀 고려할 수 없다는 점이다. 따라서 k번째 연결의 프리코더를 선택함에 있어서 다른 연결로의 영향까지 고려하여 선택할 수 있다면, 참여하고 있는 전체 연결의 성능의 합에 관점에서 봤을 때 성능을 향상 시킬 수 있을 뿐만 아니라 시스템의 불안정성 또한 줄일 수 있다.

4.1 성능 우선순위 가중치에 의한 프리코더 선택

식 (12)는 현재 업데이트하고자 하는 연결의 프리코더를 선택할 때 k번째 연결의 성능만을 고려하여 프리코더를 선택함을 나타낸다. 하지만 식 (5) 및 (12)에서 알 수 있듯이, 다른 연결 또한 k번째 연결의 선택에 영향을 받고, 이와 같은 다른 연결의 성능변화를 고려하기 위해서는 비협력적인 방법 보다는 협력적인 방법이 성능개선에 용이함을 알 수 있다. 제안하는 방법은 k번째 연결의 프리코더에 따른 성능뿐만 아니라 k번째 연결에서 선택한 프리코더에 따른 다른 연결의 성능 또한 반영하는 방식이다. 기존의 분산적인 방식과의 차이점은 피드백을 현재 연결 뿐 아니라 다른 연결로부터도 받으며, 현재 연결 및 다른 연결로부터 피드백 받는 데이터들이 분산적인 방식에서와 같이 결정되어진 프리코더를 피드백 하는 것이 아닌 프리코더를 결정하기 위한 자료로서 활용되어 진다는 점이다.

프리코더의 선택은 앞서 언급한 바와 같이 k번째 연결이 선택한 프리코더에 의해 변화하는 다른 연결들의 각자의 성능을 바탕으로 부여하는 우선순위 및 가중치를 반영하게 된다. k번째 사용자가 선택한 프리코더 인덱스 r_k에 j번째 사용자의 성능 우선순위에 따라 부여되는 우선순위를 b_{j,r_k}, 그에 부여하는 가중치를 c_{j,r_k}라고 가정하자. 이 때 우선순위 b_{j,r_k}는 사용하는 코드북의 개수 L 만큼의 범위를 가지며, r_k에 따라 1에서 L 까지의 값을 각각 부여받게 된다. 이는 다음과 같이 나타낼 수 있다.

$$b_{j,r_k} = W\left[\sum_{i=1}^{N}\log_2(1+SINR_{i,j,r_k})\right] \tag{14}$$

여기서 $W[\bullet]$는 앞서 설명한 바와 같이 성능에 따라 우선순위를 부여하는 함수이다. 우선순위 b_{j,r_k} 계산한 후, 모든 연결의 수신단은 우선순위 $\{b_{j,r_k};r_k=1,2,\cdots,L\}$을 k번째 연결의 송신단으로 피드백하게 된다. 이후, k번째 연결의 수신단에서는 각 연결로부터 피드백 받은 각 연결의 성능 우선순위 b_{j,r_k}에 가중치 c_{j,r_k}를 부여한 후, 이 가중치의 합이 가장 큰 프리코더를 적용 프리코더로서 선택하게 된다. 이 프리코더 선택 범주는 다음과 같이 나타낼 수 있다.

$$l_k = \arg max \sum_{j=1}^{N} c_{j,r_k}\Big|_{(1<r_k<L)} \tag{15}$$

부여하는 가중치는 프리코더 선택에 따른 성능 변화의 양상이 선형성을 보임을 이용하여, 우선순위에 따라 $\{c_{j,r_k};b_{j,r_k}=1,2,\cdots,L\}$의 가중치를 가지는 가장 간단한 선형적인 가중치 부여방식을 사용하였다.

4.2 가중치 분포 형태와 피드백 양의 상관관계

k번째 연결에서 선택한 프리코더가 다른 연결의 성능에 주는 영향을 반영하기 위해서는 먼저 주어진 시스템에서 선택한 프리코더에 따른 시스템의 성능 변화를 확인해야 할 필요가 있다. 이를 위하여 다른 연결이 선택한 프리코더가 고정된 상황에서 k번째 연결이 선택한 프리코더에 따른 SINR 변화의 추이를 관찰하였다.

그림 2는 프리코더의 우선순위에 따른 성능을 나타낸 그림이다. 그림에서 선택한 프리코더에 따른 성능

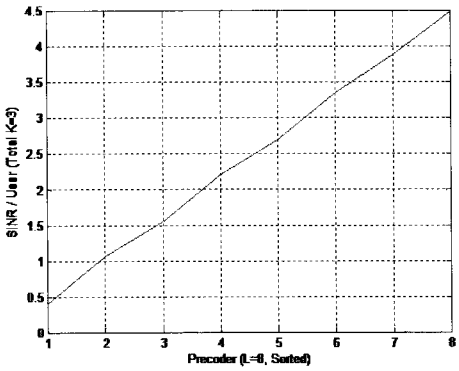


그림 2. 선택한 프리코더에 따른 SINR (오름차순 정렬).

변화는 선형성을 가짐을 알 수 있고, 따라서 선택하는 프리코더의 우선순위를 간단한 선형적 가중치로서 나타낼 수 있음을 알 수 있다. 이하에서는 L 개의 종류를 가지는 코드북을 사용할 경우, 우선순위에 따라 $\{c_{j,r_k};b_{j,r_k}=1,2,\cdots,L\}$의 가중치를 가지는 가장 간단한 선형적인 가중치 부여방식을 사용하였다. 가중치 부여방식을 코드북과 함께 모든 연결의 송수신단에서 알고 있을 때, 피드백은 단순히 프리코더의 인덱스 순서를 피드백 함으로서 적용할 수 있다.

4.3 가중치의 분포 형태와 피드백 양의 상관관계

4.2 절에서 언급한 선형적 가중치 분포를 사용할 경우, k번째 연결이 다른 연결로부터 $B_j=\{b_{j,r_k};r_k=1,2,\cdots,L\}$을 수집하는데 있어서 피드백 데이터가 가질 수 있는 경우의 수 M은 $L!$임을 알 수 있다. 예를 들어 2bit 코드북을 사용하는 경우, $L=2^2=4$로서 4개의 프리코더가 존재하게 되고, 이 때 우선순위 데이터의 경우의 수는 프리코더의 순차조합만큼 존재하므로 $M=4!=24$ 개의 종류를 가지게 된다.

이 때 다른 연결로부터 현재 업데이트 하고자 하는 연결로의 피드백 양 $R_{j,k}$는 $\log_2 M$의 피드백 양을 가지게 되고, k번째 연결의 업데이트는 자신을 포함한 모든 연결의 수신단으로부터 데이터를 수집하게 된다. 또한 통신에 참여하고 있는 모든 연결 K가 X회 순차적인 전체반복을 하여 업데이트를 시행한다고 하면, 이 때 전체 업데이트 반복횟수를 포함한 피드백 정도는 다음과 같이 나타낼 수 있다.

$$R=X\sum_{j=1}^{K}R_k=XK^2\log_2 M \tag{16}$$

반복횟수 X는 사용하는 코드북에 따라 경험적으로 적절하게 설정할 수 있고, 참여하는 연결의 수는 고정되어 있다고 가정할 때, 협력 정도 R은 가중치 테이블의 개수 M에 의존하게 된다. 하지만 이와 같이 코드북에 존재하는 모든 프리코더의 우선순위를 피드백하는 방식을 띄게 된다면 피드백 양이 너무 많아진다는 단점이 있다. 예를 들어 3명의 사용자가 참여하고 있는 상황에서 $L=8$인 코드북을 사용하는 경우 일회의 반복을 위해 $R=3\log_2(8!)=48bit$의 데이터 수집 과정을 거쳐야 한다. 따라서 최상의 성능을 내는 단 하나의 프리코더를 찾는 과정임을 감안할 때, 같은 가중치의 기대값을 가지는면서도 피드백 양은 줄일 수 있는 적절한 가중치 테이블을 설정함으로서 우선순위의 피드백 양을 줄일 수 있다.

표 1. 가중치 부여형태에 따른 피드백 테이블의 가지수 M과 피드백 양 R

순번	C	M	$R_{j,k}=\log_2(M)$
1	[1 2 3 4 5 6 7 8]	8!	16 (15.30)
2	[4 4 4 4 4 4 4 8]	8	3 (3)

위는 $L=8$일 경우 가중치 부여방식의 예시이다. 모든 우선순위를 피드백하는 1번의 가중치 부여방식을 선택하는 경우 협력을 위해 매번 16비트의 데이터를 전송해야 하지만 피드백 양의 감소를 위해 같은 가중치의 기댓값을 가지는 2번의 부여방식을 선택하는 경우에는 단순히 가장 좋은 성능을 가지는 프리코더의 인덱스만을 피드백 함으로서 피드백 정도를 줄일 수 있다. k번째 연결에서 최종적으로 선택하는 프리코더는 수집한 피드백 데이터로부터 송신단이 결정하게 된다.

4.4 표준편차 가중치의 추가적인 도입

지금까지는 k번째 연결이 선택한 프리코더에 따른 다른 연결의 성능 차이의 정도가 k번째 연결의 성능 차이의 정도와 동일하다는 가정에서 가중치의 합을 이용하여 프리코더를 선택하였다. 하지만 선택한 프리코더에 따른 SINR이 모두 선형적인 특성을 가지되, 각 연결에서의 성능의 변화폭 정도는 연결마다 다를 수 있다. 예를 들어 한 연결이 10만큼의 변화폭을 가질 때, 다른 연결은 5만큼의 변화폭을 가질 수도 있는 것이다. 이와 같이 프리코더에 따른 SINR의 변화폭이 적을 때는 그 연결에서의 프리코더 선택에 따른 성능 변화폭의 중요 정도는 떨어진다고 볼 수 있다. 또한 이러한 변화폭의 미반영은 프리코더 선택의 비수렴으로 인한 안정성의 감소로도 이어지게 된다. 따라서 시스템의 불확실성을 줄이고 프리코더 선택의 수렴도를 높이기 위한 방법으로서 그 연결의 성능의 표준편차를 또 하나의 지표로서 활용할 수 있다. k번째 연결의 프리코더 선택에 따른 j번째 연결의 성능의 표준편차를 $d_{j,k}$로 나타낸다고 할 때, 이는 다음과 같이 나타낼 수 있다.

$$d_{j,k}=STD[\sum_{i=1}^{N}\log_2(1+SINR_{i,j,r_k})]\Big|_{(1<r_k<L)} \tag{17}$$

여기서 $STD[\ ]\|_{(1<r_k<L)}$ 는 코드북에 존재하는

모든 프리코더에 따른 sum rate들의 표준편차를 나타낸다. 식 (18)을 도입한 현재 연결의 프리코더 선택의 선택 방법은 식 (16)을 변형하여 다음과 같이 나타낼 수 있다.

$$l_k = \arg max \sum_{j=1}^{K} c_{j,r_k} \times d_{j,k} \Big|_{(1 < r_k < L)} \quad (18)$$

현재 다루고 있는 시스템에 대한 대전재는 기지국 간에는 원활한 협력 또는 피드백이 가능하나 기지국과 사용자, 또는 사용자 사이에서는 CSI의 공유 등에 어려움을 겪어 제한적인 피드백을 사용하고 있는 상황을 가정하고 있다는 것이다. 따라서, 성능의 변화폭에 따른 가중치의 중요도를 반영하기 위해 표준편차를 새로운 지표로서 활용하는 것은 좋으나, 단 한 개의 지표를 사용하더라도 이를 그대로 반영하는 것은 제한적인 피드백을 사용해야 하는 가정에 맞지 않게 된다.

따라서 이 표준편차 가중치를 몇 개의 레벨로서 나누어 반영한다면 알고리즘의 복잡도와 피드백 양을 모두 감소시킬 수 있다. 이와 같은 $d_{j,k}$의 계급적 반영은 실험적으로 얻어진 폭을 반영하였다.

V. 시뮬레이션 결과

Ⅳ장에서 소개한 알고리즘은 표 2와 같이 나타낼 수 있다. 시뮬레이션 결과에서 SNR은 한 연결에서의 송신단의 다중 안테나를 통하여 송신하는 전체파워 대비 수신단의 한 안테나에서의 노이즈 파워를 의미한다. 채널은 앞서 언급한 바와 같이 Rayleigh fading channel을 가정하였으며, 연결의 목표 채널 대비 간섭 채널의 손실은 αdB로서 α는 0에서 8 사이의 균등분포를 가정하였다. 사용한 코드북은 3bit codebook (L=8)을 사용하였다. 모든 실험은 각 parameter당 10000번의 실험을 행한 결과이다.

5.1 성능 우선순위 가중치의 부여 방식에 따른 결과비교

성능 우선순위 가중치의 부여 방식은 설정하기에 따라 각자 다른 피드백 양과 성능을 나타낸다. 이는 피드백 양이 앞서 언급한 바와 같이 설정한 가중치 부여 방식을 나타낼 수 있는 가지수 M에 의해 변화되기 때문이다. 여기서는 '가장 높은 성능을 나타내는 프리코더를 찾는다.'라는 전제를 따르는 몇 가지의 가중치

표 2. 제안하는 알고리즘의 프리코더 업데이트 방법

Initialize $l_k^{[0]}$ randomly with $k = 1,\cdots,K$
iteration begins $x = 1,\cdots,X$
  For k=1 to K
    For j=1 to K
      [Compute weighting factor $c_{j,k,r}$, $r = 1,\cdots,L$ from the priority of each precoder in codebook, eq (15).]
      [Compute STD weighting factor $d_{j,k}$ from the sumrate performance of present link, eq (18).]
      Quantize $d_{j,k}$.
    end
    [Choose appropriate precoder $l_k^{[x]}$ from the stated precoder selection criterion in eq (19).]
  end
end

부여 방식을 설정하였고, 그에 따른 피드백 양과 시뮬레이션 결과를 게재하였다.

표 3은 시뮬레이션에 사용한 가중치 부여 방식들을 나타낸 표이다. C는 가중치 부여방식을 나타낸다. M은 가중치 부여방식에 따라 발생할 수 있는 가지 수를 나타내며, R은 M에 따른 피드백 정도를 나타낸다. 설정한 가중치 부여방식들은 세분화하여 나타내고자 하는 부분은 세분화하여 따로 나타내고 나머지 부분들은 같은 기대값을 가지도록 설정하였다. 1번의 경우 가장 높은 성능을 내는 프리코더의 인덱스만을 피드백하면 되는 경우이고, 6번은 프리코더의 우선순위를 모두 피드백하는 경우이다. 2번과 3번은 가장 1순위와 2순위의 코드북을 피드백하되, 2번은 1,2순위를 비슷한 상위그룹으로 간주하여 피드백 양을 줄인 경우이고, 3번은 그대로 피드백 한 경우이다. 4번과 5번은 2번의 경우를 좀 더 확장한 경우이다. 그룹화가 더 적절한지, 적절하다면 어느 정도의 그룹화가 적정한지를

표 3. 시뮬레이션에 사용한 가중치 부여 방식

순번	C	M	R
1	[4 4 4 4 4 4 4 8]	8	3
2	[3.5 3.5 3.5 3.5 3.5 3.5 7.5 7.5]	$8C2$	5
3	[3.5 3.5 3.5 3.5 3.5 3.5 7 8]	8×7	6
4	[2.5 2.5 2.5 2.5 6.5 6.5 6.5 6.5]	$8C4$	7
5	[2.5 2.5 2.5 2.5 5.5 5.5 7.5 7.5]	$8C2\times(8C2-1)$	10
6	[1 2 3 4 5 6 7 8]	8!	16

알아보기 위해 설정하였다. 당연하게도 6번이 가장 높은 성능을 낼 것으로 기대하였다. 하지만 이는 피드백이 많으므로, 피드백 양을 줄일 수 있으면서도 성능의 저하가 크지 않은 코드북을 찾는 데 주안점을 두었다.

그림 3은 설정한 각 테이블에 따른 시뮬레이션 결과를 나타낸다. 그룹화의 성능을 알아보기 위해 설정한 4, 5번은 피드백 양이 적은 3번의 경우보다도 낮은 성능을 보여주었다. 2번의 경우 역시 피드백 양이 적은 1번의 경우와 비슷한 성능을 보였다. 따라서 성능을 그룹화하는 경우 보다는 상위 그룹을 정확히 피드백 해내는 것이 중요하다는 것을 알 수 있다. 또한 전체적인 관점에서 보았을 때, 피드백 양이 더 많은 3번과 가장 최소의 피드백을 하는 1번의 경우의 성능차가 그리 크지 않았다. 따라서 각 연결에서 최고의 성능을 내는 코드북만을 최소의 피드백양으로 피드백하는 1번의 경우가 가장 적절하다고 판단하였고, 이하에서는 이 1번 가중치 부여방식을 사용하여 제안하는 방식에 대한 시뮬레이션을 행하였다.

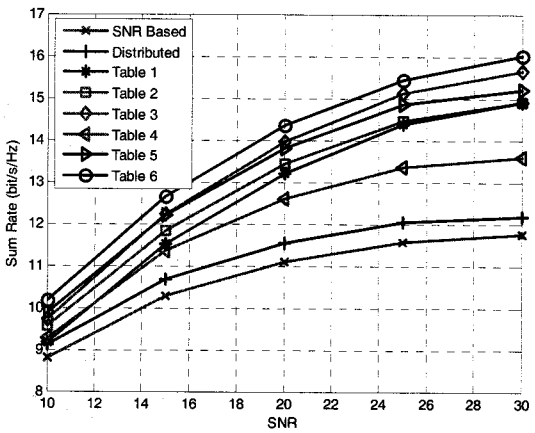


그림 3. 가중치 부여방식에 따른 성능 비교

5.2 가중치 적용시의 결과 비교

그림 4와 5는 각각 K=3, P=2, N=1 과 K=4, P=3, N=1에서 기존의 방식과 이론적인 상한선, 그리고 Ⅳ.2에서 언급한 가장 간단한 가중치 부여방식을 사용하여 제안하는 방식을 적용했을 때의 sum-rate 비교 결과이다. 그림에서 SNR based는 간섭을 고려하지 않았을 경우의 성능, Centralized Optimization은 이론적인 상한선, Distributed는 기존의 분산적인 방식을 나타내고, Cooperative 3bit Feedback은 제안하는 방식 중 성능 우선순위 가중치만을 적용하였을 때의 결과를, Cooperative 3bit Feedback (Ideal)은 이상적인 경우의 표준편차 가중치를 추가적으로 적용하였을 때

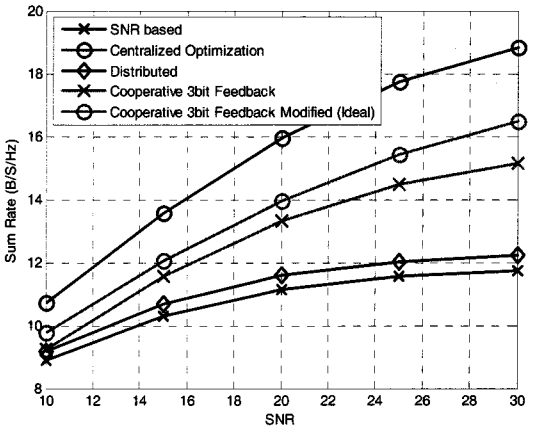


그림 4. 성능우선순위 가중치 적용 후 Sum Rate 비교 결과 (K=3, P=2, N=1)

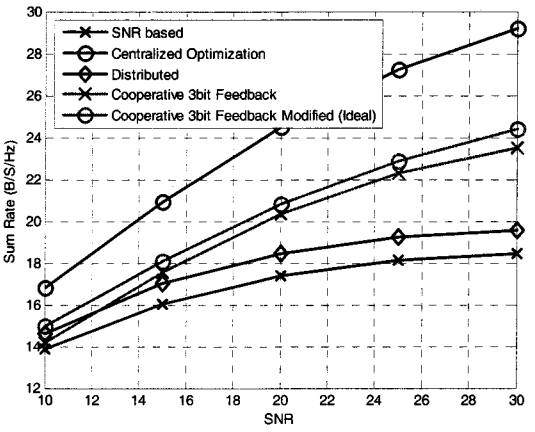


그림 5. 성능우선순위 가중치 적용 후 Sum Rate 비교 결과 (K=4, P=3, N=1)

의 시뮬레이션 결과를 나타낸다. SNR이 낮을 경우, 간섭보다도 잡음에 의해서 성능이 결정되기 때문에 성능 개선 정도가 미미한 양상을 보인다. 하지만 높은 *SNR*의 경우, 분산적인 방식이 4% 정도로 낮은 성능 개선율을 보이는데 비해 성능우선 순위 가중치만을 적용하였을 때는 그림에서 최대로 약 29%, 표준편차 가중치를 추가로 적용하였을 때는 40% 정도의 성능 개선 정도를 보임을 알 수 있다. 또한 분산적 방식이 중앙집중화된 방식에 비해 약 65% 정도의 성능을 내지 못하는 데 비하여 제안한 협력적 방법은 두 방법을 모두 적용하였을 경우 87% 정도의 성능을 낼 수 있음을 알 수 있다. 이는 그림 4와 5에서 나타나는 바와 같이 참여한 연결의 수와 상관없이 성능을 얻어낼 수 있음을 알 수 있다.

그림 6은 프리코더 선택의 비수렴도를 나타낸다. 그림으로부터 성능우선순위 가중치만을 적용하였을

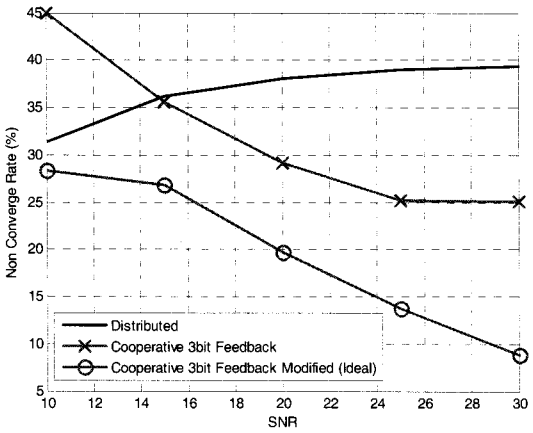


그림 6. 표준편차 가중치의 추가 적용 후 비수렴도 비교 결과 (K=4, P=3, N=1)

때도 분산적인 방식에 비해 수렴도가 향상되었으나, 전체적인 관점에서 보았을 때는 여전히 비수렴도가 25% 이상으로 높음을 알 수 있다. 하지만 표준편차 가중치를 적용할 경우, 높은 SNR에서는 비수렴도가 5%에 근접한 정도로 많이 낮아졌음을 알 수 있다. 또한 SNR이 높아져 전체적인 성능이 프리코더 선택에 따른 간섭의 영향에 의한 성능저하 문제로 옮겨갈수록, 분산적인 방식은 비수렴도가 높아지고, 성능우선순위 가중치만을 적용한 방식 역시 비수렴도가 감소는 하나 일정 이하로 내려가지 못하는 양상을 보이나, 표준편차 가중치를 추가적으로 적용하였을 때는 전체적으로 비수렴도가 낮고 또 SNR이 증가할수록 점차 낮아지는 양상을 보임을 알 수 있다.

그림 7은 프리코더 업데이트가 수렴할 경우 전체 반복회수 X의 평균값을 나타낸다. 반복회수의 경우, 기존의 분산적인 방식보다 어느정도 증가하는 양상을 보인다. 하지만 표준편차 가중치의 적용시 다시 어느 정도 낮아지는 양상을 보임을 알 수 있으며 2회 안쪽으로 근접함을 알 수 있다. 이는 반복회수는 시스템에서 일정하게 고정하고 반복해야 함을 가정하면, 반복횟수 설정에서 분산적인 방식과 같은 회수로 설정하여도 무리가 없음을 알 수 있다. 또 그림 6에서 알 수 있듯이 제안하는 방식의 비수렴도가 기존의 분산적인 방식에 비해 월등히 낮음을 고려하면, 실제 반복회수는 더 적어짐을 알 수 있다.

5.3 양자화된 표준편차 가중치 적용시의 결과비교

앞장에서 언급했던 바와 같이 표준편차 가중치 변수를 그대로 피드백 하는 것은 피드백 하는 과정에서 더 큰 오류가 발생할 수 있고, 시스템의 대 전재에 맞지 않는 양상을 보이게 된다. 따라서 이 변수를 가정에 맞게 양자화하여 전송할 필요성이 있는데, 이 경우 양자화에 의해 이상적인 경우보다 어느 정도 왜곡과 오류가 발생하게 된다. 이 절에서는 표준편차 가중치 변수의 양자화 정도에 따른 성능을 알아본다. 계급화는 시뮬레이션에서 얻어낸 변수의 최대값을 바탕으로 설정하여 행하였다.

그림 8과 9는 계급화 정도에 따른 시뮬레이션 결과이다. 모든 결과는 계급화 정도가 높아질수록 성능이나 수렴 정도에 있어서 표준편차 가중치 변수의 값이 이상적인 피드백이 행해진 경우로 비례적으로 수렴함을 알 수 있다. 이와 같이 계급화된 표준편차 가중치 변수를 적용할 경우, 매 피드백 시 더해지는 피드백의 양은 변수의 계급화 정도를 따르게 된다. 변수의 계급화 정도를 G로서 나타낸다고 할 때, 추가되는 피드백

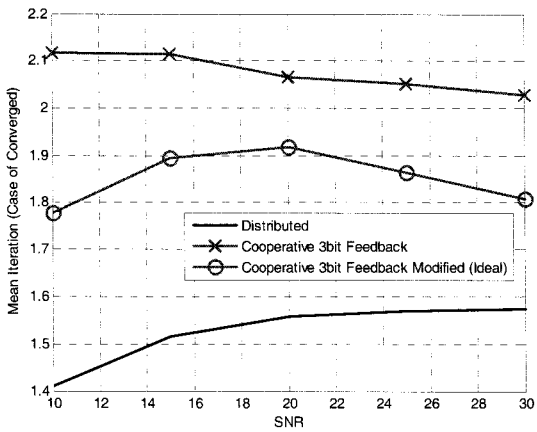


그림 7. 표준편차 가중치의 추가 적용 후 반복횟수 비교 결과 (K=4, P=3, N=1)

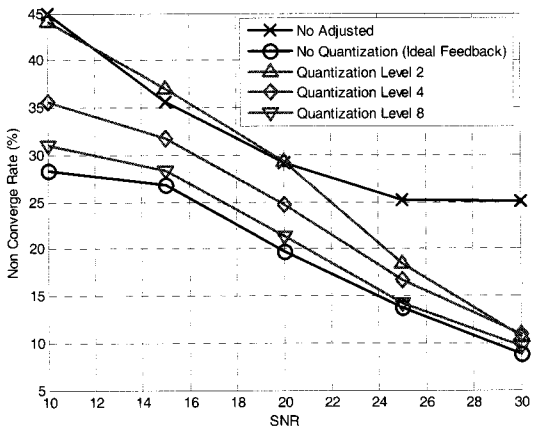


그림 8. 표준편차 가중치의 양자화에 따른 비수렴도 비교 결과 (K=4, P=3, N=1)

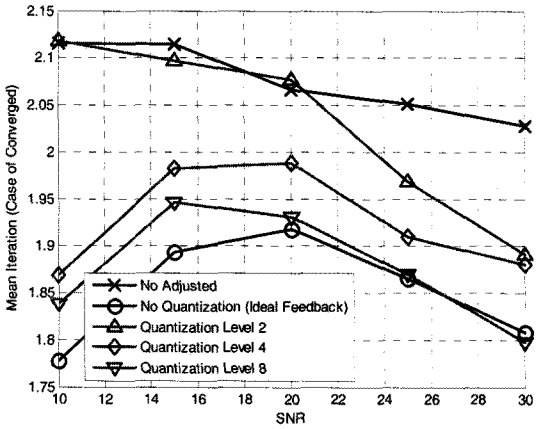


그림 9. 표준편차 가중치의 양자화에 따른 반복 횟수 비교 결과 (K=4, P=3, N=1)

정도는 $\log_2 G$로서 나타낼 수 있다. 이 때 식 (17)에 표준편차 가중치 변수의 피드백 양을 더한 전체 피드백 양은 다음과 같이 나타낼 수 있다.

$$B_k = X\sum_{j=1}^{K} B_k = XK^2(\log_2 M + \log_2 G) \qquad (19)$$

VI. 결 론

본 논문에서는 제한적 피드백 프리코딩을 사용하는 간섭 제한적인 MIMO 시스템에서 제한적인 협력을 통해 시스템 전체의 성능 및 안정성을 향상시킬 수 있는 방법에 대해 논의하였다. 제안하는 방법은 각 연결에서 코드북에 존재하는 각 프리코더에 따른 성능우선순위와 연결 자체의 성능 변화정도에 가중치를 부여하는 방식이다. 시뮬레이션 결과 기존의 분산적인 방식에 비해 약 36% 정도 성능이 개선됨을 알 수 있었고, 이론적인 상한선에 대해 약 87% 정도의 성능을 냄을 알 수 있었다. 시스템의 안정성과 연결되는 프리코더의 수렴도 또한 기존의 분산적인 방식이 수렴도가 낮음에 비해 전체적으로 수렴도가 향상됨을 알 수 있었다.

따라서 제안하는 협력적인 방식은 프리코더 선택 방법에 있어서 피드백에 의한 부담을 줄이면서도 좀 더 효율적이고 안정적인 성능을 내는 방법으로서 쓰일 수 있을 것으로 기대된다.

참 고 문 헌

[1] Telatar E.: "Capacity of multi-antenna Gaussain channels," *European Trans. Telecomm.*, 10, pp. 585-595, 1999.

[2] Caire G., and Shamai S.: "On the achievable throughput of a multiantenna Gaussian broadcast channel," *IEEE Trans. Inf. theory*, 49, pp. 1691-1706, 2003.

[3] Vishwanath S., Jafar S., and Goldsmith A.: "Duality, achievable rates, and sum-rate capacity of Gaussian MIMO broadcast channels," *IEEE Trans. Inf. Theory*, 49, pp. 2658-2668, 2003.

[4] Love D. J., and Heath R. W.: "Grassmannian beamforming for multiple-input multiple-output wireless systems," *IEEE Trans. Inf. Theory*, 49, pp.2735-2747, 2003.

[5] Love D. J., and Heath R. W.: "Limited feedback unitary precoding for spatial multiplexing systems," *IEEE Trans. Inf. Theory*, 51, pp.2967-2976, 2005.

[6] Li C., and Wang X.: "Cooperative multi beamforming in ad hoc networks," *EURASIP Journal on Advances in Signal Processing*, 310247, 2008.

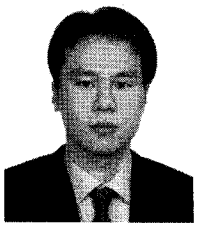
[7] Scurari G., Palomar D. P., and Barbarossa S.: "The MIMO iterative waterfilling algorithm," *IEEE Trans. Signal Process.*, 57, pp. 1917-1935, 2009.

[8] Lee J.-H., and Li G. Y.: "Iterative limited feedback beamforming for MIMO ad-hoc networks," *Proc. IEEE Globecom* 2009, Honolulu, HI, Dec. 2009.

[9] Trivellato M., Boccardi F., and Huang H.: "On transcevier design and channel quantization for downlink multiuser MIMO systems with limited feedback," *IEEE J. Select. Areas Commun.*, 26, pp. 1494-1504. 2008.

[10] Lee H, Lee K, Hochwald B. M, and Lee I, "Regularized channel inversion for multiple-antenna users in multiuser MIMO downlink," *in Commun., 2008. ICC '08. IEEE International Conference on*, pp. 3501-3505, May 2008.

윤 정 민 (Jung-min Yoon) 정회원



2009년 2월 연세대학교 전기전자공학부 졸업
2011년 2월 서울대학교 전기컴퓨터공학부 석사
2011년 2월~현재 삼성전자 DMC 연구소 재직 중
<관심분야> MIMO 시스템, 전파전파, 이동통신

이 종 호 (Jong-ho Lee) 종신회원

1999년 2월 서울대학교 전기공학부 학사
2001년 2월 서울대학교 전기컴퓨터공학부 석사
2006년 2월 서울대학교 전기컴퓨터공학부 박사
2006년 3월~2008년 8월 삼성전자 통신연구소 책임연구원
2008년 9월~2009년 8월 Georgia Institute of Technology 박사 후 연구원
2009년 9월~현재 공주대학교 전기전자제어공학부 조교수
<관심분야> 통신신호처리, 통신이론

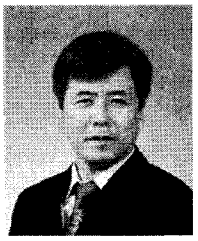
곽 영 우 (Young-woo Kwak) 정회원

2005년 8월 포항공과대학교 전자전기공학과 졸업
2005년 9월~현재 서울대학교 전기컴퓨터공학부 석박사통합과정 재학중
<관심분야> MIMO 시스템, MIMO-Relay, 이동통신

최 정 식 (Jeong-sik Choi) 준회원

2010년 2월 포항공과대학교 전자전기공학과 졸업
2010년 3월~현재 서울대학교 전기컴퓨터공학부 석사과정
<관심분야> MIMO 시스템, 이동통신

김 성 철 (Seong-Cheol Kim) 종신회원



1984년 서울대학교 전기공학과 학사
1987년 서울대학교 전기공학과 석사
1995년 Polytech University 박사
1995년~1999년 AT&T 연구소
1999년~현재 서울대학교 교수
<관심분야> 무선 통신, 이동 통신, 유무선 채널 모델링, MIMO-OFDM, 전력선 통신, 위성 통신, 전파전파, 전파환경