LLL 알고리즘 기반 이중 스피어 MIMO 수신기

*전명운 **이정우
서울대학교 전기컴퓨터공학부, 뉴미디어통신공동연구소
*ifindme@wspl.snu.ac.kr **junglee@snu.ac.kr

LLL Algorithm Aided Double Sphere MIMO Detection

*Jeon, Myeongwoon **Lee, Jungwoo
Seoul National University EECS, INMC

요약

격자 감소 (lattice reduction) 알고리즘은 주어진 기저 벡터를 직교에 가까운 기저 벡터로 바꾸어 준다. 그중 대표적인 알고리즘으로 LLL (Lenstra, Lenstra & Lovasz) 알고리즘이 있다. 격자 감소 알고리즘을 이용하여 다중 안테나 입출력(MIMO) 통신시스템의 선형 수신기(linear detector)의 성능을 향상 시킬 수 있다. 스피어 복호 알고리즘 (sphere decoding algorithm)은 MIMO 통신 시스템에서 사용되는 복호기중 최대 우도 복호기 (Maximum Likelihood Detector)와 비슷한 BER(bit error rate)성능을 가지고 복잡도를 줄일 수 있어서 많이 연구되어 왔다. 이때 스피어의 반지름의 설정이나 트리 검색 구조 방식 등은 복잡도에 큰 영향을 미친다. 본 논문에서는 LLL 알고리즘에 기반하여 스피어의 반지름 설정 및 트리 검색 노드 수를 제한하는 방식으로 스피어 복호 알고리즘의 복잡도를 기존 알고리즘에 비해 크게 낮추면서도 비트 오류률 (BER) 성능 열화를 최소한으로 한 알고리즘을 제안하고 전산 실험을 통해 검증한다.

1. 서론

데이터의 고속 송수신을 위해 다중 입출력 (Multiple Input Multiple Output) 통신 시스템 연구가 활발히 이루어져왔으며, 이에 따라 MIMO 시스템의 수신기에서 사용되는 복호 알고리즘의 연구도 많이 이루어졌다. 복호 알고리즘 중에서는 ML 복호기 (Maximum Likelihood detector)가 가장 좋은 성능을 보이지만 복잡도가 매우 높아서 실제 구현하기가 어렵기 때문에, 이를 대체하기 위하여 ZF (Zero Forcing) 이나 MMSE (Minimum Mean Square Error) 같은 선형 복호기가 사용해 왔다. 하지만 점차 밀도높은 데이터 송수신을 위해 Modulation Order이 상승함에 따라 이들 보다 BER (Bit Error Rate) 성능이 향상된 수신기를 필요로 한다. 이에 MMSE V-BLAST (Vertical Bell Labs Layered Space-Time)같은 비선형 복호기가 개발되었지만 여전히 최적인 ML 복호기의 성능에는 접근하지 못하고 있다. 이에 ML의 성능에 근접하면 복잡도를 줄인 스피어 복호기(Sphere Decoder)가 연구되고 있다. ML이 성좌내의 모든 벡터를 대입하여 그 유클리드 거리가 최소인 해를 찾아내는데 비해, 스피어 복호기는 트리 탐색을 통해 성좌내의 벡터들을 레벨 별로 대입해나가며 일정한 거리 이상, 즉 스피어의 경계를 넘어가는 벡터에 대해서는 검색을 중단함을 통해 상당한 복잡도 감소를 가능하도록 한다[1][2]. 하지만 스피어의 반지름의 설정이나 트리 검색 구조 방식 등은 복잡도에 큰 영향을 미치며, 이를 낮추는 것이 주요 해결되어야할 과제이다. 이 문제를 해결하기 위해서 다른 선형 대수학 관련 알고리즘이 이용 될 수 있다. 격자 감소(lattice reduction) 알고리즘은 주어진 기저 벡터에 대해 가장 작고 직교에 가까운 기저벡터를 찾아준다. 격자 감소 알고리즘에는 LLL (Lenstra, Lenstra & Lovasz) , Minkowski reduction , Gauss reduction, Seysen reduction 알고리즘 등이 알려져 있다[3][4]. 이중 LLL 알고리즘이 적절한 복잡도와 성능으로 가장 널리 이용된다. LLL 알고리즘을 이용하여 수신기에 적용하면, 채널을 거치면서 왜곡된 신호의 기저를 직교에 가깝도록 변환시켜서, ZF, MMSE등의 선형 복호기의 성능을 향상 시킬 수 있다. 격자 감소 알고리즘과 스피어 복호기는 QR decomposition을 기반으로 하는 공통점을 가진다.

본 논문에서는 LLL 알고리즘을 이용해 스피어 복호기의 스피어의 반지름 설정의 기준이 되는 벡터를 설정하고 차원(layer) 별로 서로 다른 트리 검색 노드수로 제한하는 방식으로 스피어 복호 알고리즘의 복잡도를 기존 알고리즘에 비해 크게 낮추면서도 비트 오류률 (BER) 성능 열화를 최소한으로 한 알고리즘을 제안한고, 기존 다른 스피어 복호 알고리즘과 비교하여 비트오류률(bit error rate) 전산실험을 통해 성능 및 복잡도를 측정한다.

본 논문은 아래와 같이 구성되어있다. 2장에서는 격자 감소 알고리즘으로서 LLL알고리즘과 스피어 복호기 관하여 설명하고 3장에서는 LLL 알고리즘에 기반한 차원별 노드 제한 스피어 복호기 알고리즘을 제안한다. 4장에서는 제안된 알고리즘의 BER 성능과 복잡도에 관한 전산실험 결과를 보여주고 마지막으로 5장에서 결론을 맺도록 한다

2. LLL 알고리즘과 스피어 복호 알고리즘

가. 격자 감소 (lattice reduction) 기반 선형 검출기

새로운 기저(basis) 벡터를 찾아내는 격자 감소 알고리즘을 사용할

때, relection, swap, translation 변환은 다음과 같은 unimodular 행렬 T로 나타낼수 있다[4].

$$\mathbf{B}' = \mathbf{BT} \tag{1}$$

이때 B는 주어진 기저 벡터, B′는 새롭게 얻어진 기저 벡터를 의미한다. LLL알고리즘은 B=QR일 때 QR decomposiiton을 이용하여 최적의 기저벡터로 변환시키는데, 그 과정은 다음과 같고, 출력으로 T을 얻어내서 이용 할 수 있다.

LLL Algorithm

$Initialization: Q, R, P(=\mathrm{I_m})$

1) $\tilde{Q} = Q, \tilde{R} = R, T = P, k = 2$

2) while $k \le m$

3) for $l = k-1, \ldots, 1$

4) $\mu = \left\lceil \dfrac{\tilde{R}(l,k)}{\tilde{R}(l,l)} \right\rfloor$

5) if $\mu \ne 0$

6) $\tilde{R}(1:l,k) = \tilde{R}(1:l,k) - \mu\tilde{R}(1:l,l)$

7) $T(:,k) = T(:,k) - \mu T(:,l)$

8) end

9) end

10) if $\delta\tilde{R}(k-1,k-1)^2 > \tilde{R}(k,k)^2 + \tilde{R}(k-1,k)^2$

11) Swap columns k-1 and k in R and T

12) Calculate givens rotation matrix Θ such that element $R(k,k\text{-}1)$ becomes zero

$$\Theta = \begin{bmatrix} \alpha & \beta \\ \beta & \alpha \end{bmatrix},\ \alpha = \frac{\tilde{R}(k-1,k-1)}{\left\|\tilde{R}(k-1:k,k-1)\right\|},\ \beta = \frac{\tilde{R}(k,k-1)}{\left\|\tilde{R}(k-1:k,k-1)\right\|}$$

13) $R(k\text{-}1:k, k\text{-}1:m) = \Theta R(k\text{-}1:k, k\text{-}1:m)$

14) $Q(:,k\text{-}1:k) = Q(:,l\text{-}1:k)\Theta^T$

15) $k = \max\{k-1, 2\}$

16) else

17) $k = k+1$

18) end

19)end

위의 LLL 알고리즘을 이용하여 MIMO에서 사용되는 선형 복호기 (linear detector)의 성능을 향상 시킬수 있다[4].

나. 스피어 복호 알고리즘

y는 수신신호, H는 N×M채널 행렬, x는 송신신호, n은 잡음벡터라 하면 다음과 같은 식이 성립한다.

$$\mathbf{y} = \mathbf{Hx} + \mathbf{n} \tag{2}$$

이때. Maximum likelihood의 식은 아래와 같다.

$$x_{ML} = \arg\min_{x\in\Omega} \left\|\mathbf{y} - \mathbf{H}\hat{\mathbf{x}}\right\|^2 \tag{3}$$

스피어 디코딩은 수신신호와 격자벡터사이의 길이가 반지름보다 작아야 하므로,

$$\left\|\mathbf{y} - \mathbf{H}\hat{\mathbf{x}}\right\|^2 < r^2 \tag{4}$$

채널행렬 H를 QR decomposition을 사용하여 분해한 후 이를 적용하면,

$$\mathbf{H} = \mathbf{Q}\begin{bmatrix} \mathbf{R} \\ 0_{(N-M)\times M} \end{bmatrix} \tag{5}$$

$$\left\|\mathbf{y} - [\mathbf{Q_1} \quad \mathbf{Q_2}]\begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix}\hat{\mathbf{x}}\right\|^2 < r^2 \tag{6}$$

$$\left\|\begin{bmatrix} \mathbf{Q_1^H} \\ \mathbf{Q_2^H} \end{bmatrix}\mathbf{y} - \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix}\hat{\mathbf{x}}\right\|^2 < r^2 \tag{7}$$

$$\left\|\mathbf{Q_1^H y} - \mathbf{R}\hat{\mathbf{x}}\right\|^2 + \left\|\mathbf{Q_2^H y}\right\|^2 < r^2 \tag{8}$$

두번째 항은 수신신호와 채널 행렬에 의해 고정된 값이므로 이를 정리하면

$$r'^2 \equiv r^2 - \left\|\mathbf{Q_2^H y}\right\| \tag{9}$$

$$\mathbf{y}' \equiv \mathbf{Q_1^H y} = [y_1', y_2', \cdots, y_m']^T \tag{10}$$

$$\begin{aligned} &\left\|\mathbf{y}' - \mathbf{R}\hat{\mathbf{x}}\right\| \\ &= (y_M' - r_{M,M}\hat{x}_M)^2 + (y_{M-1}' - r_{M-1,M}\hat{x}_M - r_{M-1,M-1}\hat{x}_{M-1})^2 \\ &+ \cdots + (y_1' - r_{1,M}\hat{x}_M, - r_{1,M-1}\hat{x}_{M-1} \cdots - r_{1,1}\hat{x}_1)^2 \le r'^2 \end{aligned} \tag{11}$$

이 식을 만족하는 x벡터를 찾아내려면 위 항의 첫번째 항이 반지름을 넘는지 검사하고 만약 괜찮으면 그 다음 항까지 누적하여 더하여 반지름을 넘는지 조사하는 방식으로 구현할 수 있다. 이때 각 항이 각 차원을 얘기하며, 특정 차원에서 반지름을 넘어간다면 중단하고 다음 벡터를 이용하여 검색을 계속 해 나간다. 이 검색 구조를 트리 구조로 변화시켜 시각화 할 수 있다.

3. LLL 알고리즘 기반 이중 스피어 MIMO 수신기

가. LLL 알고리즘 기반 스피어 반지름 설정

LLL 알고리즘에 기반하여 스피어 복호기의 초기 반지름 설정 및 트리 탐색 노드 제한에 이용할수 있는 추정 값을 얻을수 있다. LLL 알고리즘 기반의 선형 복호기를 이용하는데, ZF, MMSE등을 이용할 수 있다. 먼저 아래 식과 같이 채널 H를 변환한다. 이때 사용되는 T는 2

장에서 설명한 LLL 알고리즘을 통해 얻어낸다.

$$\begin{aligned} \mathbf{y} &= \mathbf{Hx}+\mathbf{n} \\ &= \mathbf{HTT}^{-1}\mathbf{x}+\mathbf{n} \\ &= \tilde{\mathbf{H}}\mathbf{z}+\mathbf{n} \end{aligned} \tag{12}$$

만약 ZF 알고리즘을 이용한다고 가정하면

$$\begin{aligned} \tilde{\mathbf{z}}_{LR-ZF} &= (\tilde{\mathbf{H}}^T\tilde{\mathbf{H}})^{-1}\tilde{\mathbf{H}}^T\mathbf{y} \\ &= \mathbf{z}+\tilde{\mathbf{H}}^{+}\mathbf{n} \\ &= \mathbf{T}^{-1}\tilde{\mathbf{x}}_{ZF} \end{aligned} \tag{13}$$

와 같이 변환된 채널을 통해 벡터를 구할 수 있다. 추정된 신호 벡터는 다음과 같다.

$$\hat{\mathbf{x}}_{LR-ZF} = \mathbf{T}Q(\tilde{\mathbf{z}}_{LR-ZF}) \tag{14}$$

위에서 Q 함수는 modulation order에 따른 slice 기능을 한다. 위와 같은 과정을 통해서, 왜곡된 채널 H를 거친 신호에 선형 복호기가 적용될때의 왜곡이 그대로 남게 되는 것을 방지하고, 직교에 가까운 기저 벡터를 가지도록 만들어진 새 채널을 적용하여, 보다 원 신호에 가까운 벡터를 얻어낸다. 이렇게 얻어낸 벡터 값을 통해 먼저 스피어 복호기의 초기 반지름 설정에 이용할 수 있다. 이때 초기 반지름은 다음과 같이 설정한다.

$$C = \left\| \mathbf{y} - \mathbf{H}\hat{\mathbf{x}}_{LR} \right\| \tag{15}$$

스피어 복호기에서 Schnorr-Euchner 알고리즘을 사용할 경우 반지름은 트리 검색을 하면서 parital Euclidean distance (PED)의 비교에 따라 업데이트 된다[5]. LLL 알고리즘과 스피어복호기는 모두 QR decomposition 기반의 알고리즘 이므로 결합시 많은 복잡도 이득을 얻을 수 있다.

나. LLL 기반 차원별 노드수 제한 스피어 복호기

Sphere Decoding은 기본적으로 차원별로 나누어 계산하지만, 탐색 벡터의 제외 여부는 누적된 Distance로 판별한다. 이와 별개로 각 차원별로 4x4 MIMO를 가정한 아래 그림과 같이 가능성 높은 초기 벡터값 주변에 2차원적인 원을 그려서 그 안의 lattice point (노드) 에 대해서만 Sphere Decoding을 적용한다면, 방문하는 노드수가 급격하게 줄어들게 된다. 각 차원(layer)별로 제한되는 노드수는 별도로 정해질 수 있다. 이때 초기벡터는 위에서 언급한 LLL 알고리즘 기반 선형 복호기에서 얻어진 벡터를 사용할 수 있다. 따라서 차원별 노드 제한 알고리즘은 기존 Sphere의 제한에 추가로 차원별 Circle 제한을 하므로 이중 sphere으로 제한하여 복잡도를 낮추는 알고리즘이라 할 수 있다. 이러한 알고리즘을 기반으로 격자 감소 기반 차원별 노드 제한 스피어 복호기(SD-LRLC, sphere decoder - lattice reduction limited constellation)를 제안한다. 기존 Sphere Decoding의 알고리즘에 비해 LLL 알고리즘에 기반한 차원별 노드 제한 알고리즘을 적용하므로 복잡도는 급격히 낮아진다. 예를들어 4x4 MIMO 시스템에서 64QAM의 변조 방식을 사용한다 가정하면, 기존 스피어 복호기에서 트리 탐색은 $(64)^4 = 1.67*10^7$만큼의 후보군중 sphere constraint를 가지고 트리 탐색을 진행한다. 하지만 차원별 노드 제한 알고리즘을 적용하고, 만약 노드 제한을 각 차원당 64-12(k-1)로 제한한다면 (k=차원), $64*52*40*28 = 1.584*10^5$ 만큼의 벡터만 가지고 sphere constraint를 적용하여 탐색해도 되므로 기본적으로 탐색해야할 벡터수가 무려 99.1%나 감소하게 된다. 그리고 이때 제한된 이중 스피어 안의 노드(lattice point)들은 LLL 알고리즘에 기반해 실제 원추정 신호에 가깝도록 설정되어있기에 반지름도 업데이트를 통해 전통적인 스피어 디코딩 알고리즘에 비해 급격하게 작아져서 더욱 스피어 복호기의 탐색할 벡터 수를 줄여주게 된다. 격자 제한 수는 1번째 단계 (layer)는 전체 모든 노드를 검색하고, layer가 올라가면서 점차 줄어들도록 하면 최적해를 가지는 벡터를 찾는데 크게 벗어나지 않는다. 즉 차원별 제한 노드 수는 실험을 통하여 최적의 성능과 복잡도를 보이도록 테이블 기반으로 정할 수 있다.

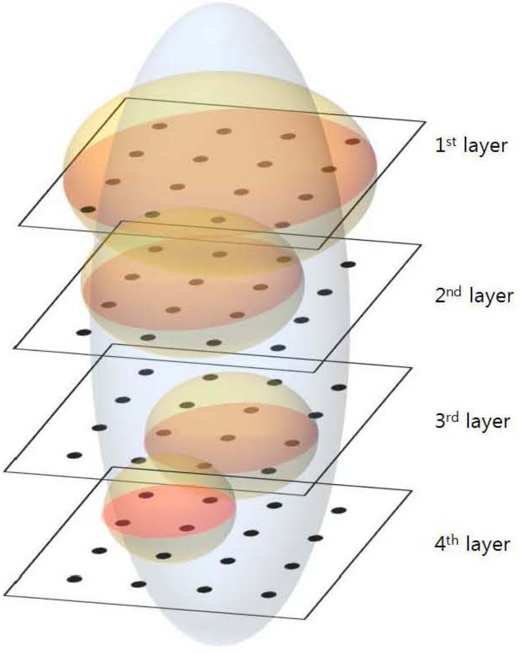

그림 1 이중 스피어 차원별 노드 수 제한 개요도

4. 성능 분석

성능을 분석하는데 있어서, 오류율 측면과 복잡도 측면에서 보고자 한다. 그림 2 에서는 기존의 수신기들과 제안된 격자 감소 (lattice reduction)기반의 차원별 노드수 제한 이중 스피어 복호기의 비트 오류율 (bit error error)의 성능을 비교했다. 4x4 MIMO, 64 QAM 시스템을 가정했으며 격자 제한 수는 1st layer는 전체 모든 노드를 검색하고, layer가 올라가면서 점차 줄어들도록 설정 하였다. Lattice reduction 알고리즘에 기반한 MMSE 수신기를 선추정 복호기로 사용하였다.

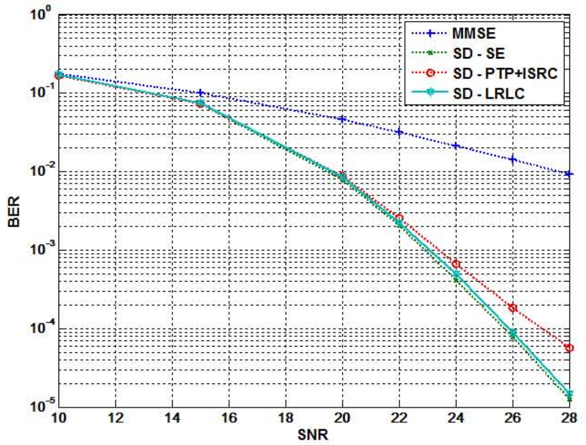

그림 2 제안된 알고리즘의 BER 성능 비교

이때 SD-SE는 전통적인 Schnorr-Euchner 스피어 복호기[5], SD-PTP+ISRC는 probabilistic tree pruning 방식 스피어 복호기[6], SD-LRLC (lattice reduction limited constellation)는 제안된 격자 감소 기반 차원별 노드 제한 이중 스피어 복호기를 나타낸다. SD-SE의 경우 최적의 벡터를 찾아낼수 있어서 ML과 같은 BER 성능을 나타낸다. 제안된 SD-LRLC 알고리즘의 성능이 최적인 SD-SE에 0.1dB 이내로 매우 근접함을 확인할 수 있다.

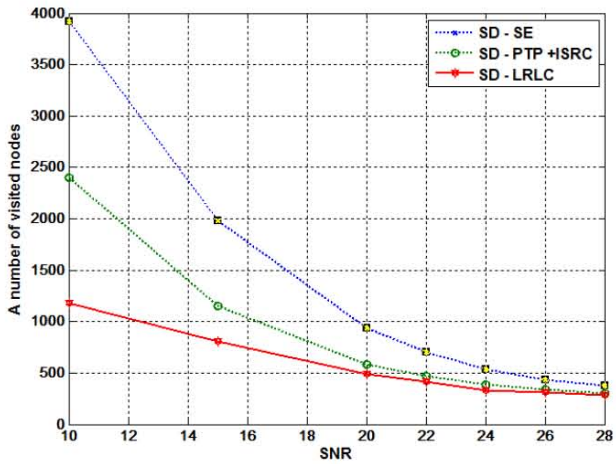

그림 3 트리 탐색 시 평균 방문 노드 수

복잡도 측면을 알아보기 위해 스피어 복호기의 알고리즘들의 트리 탐색시 방문 노드 수를 비교한다. 제안된 SD-LRLC 알고리즘이 전 SNR구간에 걸쳐서 다른 알고리즘보다 방문 노드수가 낮음을 확인할 수 있다. 즉 SD-LRLC 알고리즘이 BER이 최적인 SD-SE에 근접하면서도 복잡도 측면에서 큰 이득이 있음을 알 수 있다.

.

5. 결론

스피어 복호기 알고리즘은 ML에 근접한 성능을 내지만 여전히 선형복호기에 비해 높은 복잡도를 가지고 있으며 효율적인 반지름 설정 등의 문제가 남아있다. 제안된 SD-LRLC 알고리즘은 lattice reduction 기법에 기반하여 기존 SD에 비해 BER 성능 열화를 0.1 dB 이내로 최소로 하면서 복잡도를 나타내는 방문노드수를 SNR에 따라 최대 70% 가까이 줄일수 있다. 신뢰성과 복잡도사이의 Trade-Off 특성에 관한 연구가 필요하며, LLL 알고리즘의 복잡도를 줄이는 연구가 필요하다.

ACKNOWLEDGMENT

이 연구는 한국연구재단의 일반연구자(2010-0013397) 지원 사업, 중견연구자 (2010-0027155) 지원 사업, 한국 에너지 기술평가원 (KETEP) (No. 2011T100100151), INMC와 BK21의 지원을 받아 수행한 연구입니다.

참고 문헌

[1] U. Fincke and M. Pohst, "Improved methods for calculating vectors of short length in a lattice, including a complexity analysis," *Math. Comput.*, vol.44, no. 5, pp. 463-471, May 1985.

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

[3] A. K. Lenstra, H. W. Lenstra, and L. Lovász, ""Factoring polynomials with rational coefficients," Mathematische Annalen, vol. 261, no. 4, pp. 515 - -534, 1982.

[4] D. Wübben, D. Seethaler, J. Jaldén, G. Matz "Lattice reduction - A survey with applications in wireless communications," IEEE Signal Processing Magazine, 28(3) , pages 70 - 91, 2011.

[5] C. P. Schnorr and M. Euchner, "Lattice basis reduction: Improved practical algorithms and solving subset sum problems," *Math. Programming*, vol.66, no. 2, pp. 181-191, Sep. 1994.

[6] B. Shim and I. Kang, ""On further reduction of complexity in tree pruning based sphere search,"" IEEE Transactions on Communication, vol. 58, no.2, pp. 417-422, Feb. 2010.