효율적인 리스트 구복호기 검출방식을 위한 구반경의 최적화에 관한 연구

*이재석 **이병주 ***심병효

고려대학교

*infinite0@korea.ac.kr

Radius optimization for efficient list sphere decoding

*Lee, Jaeseok **Lee, Byoung-ju ***Shim, Byonghyo

Korea University

요약

최근의 iterative detection and decoding (IDD) 기법에서의 soft 복호화방식은, log-likelihood ratio (LLR) 값의 신뢰도를 높이기 위해 기존의 구복호화 (sphere decoding) 방식보다는 리스트를 형성하는 구복호화방식 (list sphere decoding : LSD)이 대두되고 있다. 기존의 구복호화 방식과는 달리 리스트 구복호화 방식은 그 성능의 우수함에도 불구하고, 여러 격자 포인트들을 검출해야 하므로 신호대잡음비 (signal-to-noise ratio : SNR) 의 증가에 따른 복잡도의 이득을 거의 취할 수 없을 뿐만 아니라, 무엇보다 신뢰도가 높은 LLR 값을 얻는 데에 영향이 작은 포인트를 검출하는 경우도 생긴다는 점에서 비효율적인 측면이 있다. 이에 본 논문에서는 리스트 구복호화 검출방식의 효율성을 높이기 위해 LLR 값에 적은 영향을 미치는 격자 포인트들을 제거하는 방식에 대해 연구하였다. 본 연구의 목표는 MIMO 시스템에서의 기존의 리스트 구복호화 기법의 capacity와 실제 성능과 최대한 유사한 성능을 내면서도 그 복잡도를 현저히 줄이는 것이며, 구체적으로는 검출을 위한 초기 구반경의 최적화를 기반으로 한다.

1. 서론

본 논문에서 다루는 다중안테나 (multiple-input / multiple-output : MIMO) 시스템에서의 soft-decision 방식은 hard-decision 방식에 비해 비트 에러율 (bit error rate : BER)을 크게 줄이는 효과를 가지고 있다. 이러한 soft-decision 방식을 구동하고, 그 성능을 보다 높이기 위해 IDD 기법이 소개된 바 있다 [1]. 이러한 IDD 기법에서는 채널 복호기에서 출력된 soft LLR 값들을 다시 APP (a posteriori probability) 검출기로 돌려보내어, 검출기의 성능과 더불어 채널 복호기의 그것의 향상을 도모한다. APP 검출기에서 출력되는 LLR 값들은 다음과 같이 정의된다.

$$L(b_k) = \ln \frac{P(b_{k+}|y)}{P(b_{k+}|y)} \tag{1}$$

여기서 y는 수신기에서 수신한 벡터신호이고, b_{k+}와 b_{k-}는 각각 k 번째 비트가 +1과 -1인 비트신호를 나타낸다. 일반적인 통신 모델은 다음과 같다.

$$y = Hs + v \tag{2}$$

여기서 H는 멀티안테나 채널, s는 송신된 벡터신호 그리고 v는 uncorrelated된 가우시안 잡음 벡터이다. (1) 식을 다시 정리하면,

$$\begin{aligned} L(b_k) &= \ln \frac{\sum_{b_{k+}} P(y, b_{\bar{k}}, b_{k+})}{\sum_{b_{k-}} P(y, b_{\bar{k}}, b_{k-})} \\ &= \ln \frac{P(b_{k+})}{P(b_{k-})} + \ln \frac{\sum_{b_{k+}} P(y|b_{k+}) P(b_{\bar{k}})}{\sum_{b_{k-}} P(y|b_{k-}) P(b_{\bar{k}})} \end{aligned} \tag{3}$$

이고, $f(\bullet)$를 변조함수라고 가정하면 $s = f(b)$이므로,

$$L(b_k) = \ln \frac{P(b_{k+})}{P(b_{k-})} + \ln \frac{\sum_{b_{k+}} P(y|s_{k+}) P(b_{\bar{k}})}{\sum_{b_{k-}} P(y|s_{k-}) P(b_{\bar{k}})} \tag{4}$$

과 같이 정리된다. 여기서 $b_{\bar{k}}$는 k번째 비트를 제외한 비트벡터이다. 또한 (4) 식의 첫 번째 항은 LLR 값의 *a priori* 값 $(L_A(b_k))$에 해당하고, 두 번째 항은 LLR 값의 *extrinsic* $(L_E(b_k))$값에 해당한다. 이 중에서 APP 검출기에 입력되는 $L_E(b_k)$는 가우시안 잡음 전제에 의해

$$\begin{aligned} L_E(b_k) &= \ln \frac{\sum_{b_{k+}} P(y|s_{k+}) P(b_{\bar{k}})}{\sum_{b_{k-}} P(y|s_{k-}) P(b_{\bar{k}})} \\ &= \ln \frac{\sum_{b_{k+}} \exp\left(-\frac{\|y - Hs_{k+}\|^2}{2\sigma^2} + \frac{1}{2} b_{\bar{k}}^T L_{A,\bar{k}} \right)}{\sum_{b_{k-}} \exp\left(-\frac{\|y - Hs_{k-}\|^2}{2\sigma^2} + \frac{1}{2} b_{\bar{k}}^T L_{A,\bar{k}} \right)} \end{aligned} \tag{5}$$

가 되고, 여기서 $L_{A,\bar{k}}$ 는 k번째 LLR 값을 제외한 LLR 벡터이다. 자세한 과정은 [2]를 참고하였다.

(5) 식을 통해 얻을 수 있는 LLR 값은 상당히 높은 신뢰도를 가지고 이는 성능의 향상과 이어지지만, 변조단계가 높아질수록 가능한 모든 조합의 신호벡터의 수가 기하급수적으로 증가하여 상당히 높은 복잡도를 가지는 단점이 있다. 이를 보완하기 위해 리스트 구복호기 검출방식이 제안되었다 [1]. 리스트 구복호화 기법은 구복호기 검출방식에 약간의 수정이 더해진 것으로, ML-solution만을 찾는 구복호기에 비해 더 많은 수의 심볼벡터들을 리스트 안에 포함시키는 것이 그 특징이다. 리스트 $L=\{s_1=s_{ml},s_2,\cdots,s_N\}$은 심볼벡터들로 이루어진 집합으로, N은 유저에 의해 미리 정의된 리스트의 크기이고, 리스트 안의 각 심볼벡터는 각 벡터가 가지는 cost function $J(s_i)=\|y-Hs_i\|^2$ 가 작은 순대로 배열되어 있다. APP 검출기에서 리스트 구복호화를 마무리한 뒤에는 리스트 L을 두 부분집합 L_{k+}와 L_{k-}으로 나누는데, 이는 (5) 식을 효과적으로 단순화시킨다.

$$L_E(b_k)\approx \ln\frac{\sum_{b_{k+}\in L}\exp\left(-\frac{\|y-Hs_{k+}\|^2}{2\sigma^2}+\frac{1}{2}b_{\bar{k}}^{T}L_{A,\bar{k}}\right)}{\sum_{b_{k-}\in L}\exp\left(-\frac{\|y-Hs_{k-}\|^2}{2\sigma^2}+\frac{1}{2}b_{\bar{k}}^{T}L_{A,\bar{k}}\right)} \quad (6)$$

이처럼 리스트 구복호화 기법은 리스트 안에 포함된 심볼벡터만으로 LLR 값을 계산하고, 검출기에서 또한 적은 수의 격자 포인트만을 검색함으로서 복잡도를 크게 줄일 수 있다.

2. 본론

가. IDD 시스템

IDD 시스템은, soft decision 방식에서 성능과 전체 capacity를 이론적 수치에 가깝게 올리기 위한 방식이며, 채널 복호기와 APP 검출기 사이에서 *extrinsic* LLR 값들을 차례로 주고받는 구조의 시스템이다. 본 논문에서의 총 시스템을 포함한 IDD 시스템 모델은 그림. 1에 소개되어 있다.

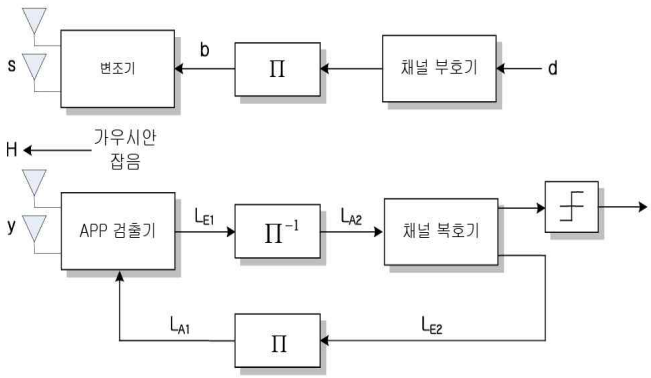

그림 1. IDD 시스템 모델

그림. 1에서, 첨자 '1'은 APP 검출기와 관련된 LLR 정보를 말하고, 첨자 '2'는 채널 복호기와 관련된 정보를 나타낸다.

나. 리스트 구복호기와 구반경

기본적인 알고리즘은 [3]의 그것을 바탕으로 한다. 구복호기가 반경 안의 격자 포인트가 검출되면 바로 cost function을 비교하여 반경 update의 여부를 결정하는 데 반해, 리스트 구복호기는 미리 정의된 N개의 격자 포인트가 리스트에 포함되기 전까진 반경을 update하지 않는다. 다시 말해, 구복호기의 초기 구반경은 Babai 포인트 [4]가 가지는 cost function이 되지만, 리스트 구복호기에서는 리스트에 이미 정의된 N개의 격자 포인트가 포함되기 전까지는 구반경의 update가 이루어지지 않고 검출되는 격자 포인트를 모두 리스트 안에 포함시키도록 되어 있다. 리스트가 가득 찼을 때 새로운 격자 포인트가 검출되었을 때에는 리스트 안의 격자 포인트 중 가장 큰 cost function을 가지는 격자 포인트와 새로운 격자 포인트의 cost function을 비교하여 그 크기가 작은 포인트를 리스트 안에 새로 추가한다.

리스트 구복호화 기법은 이런 식으로 구반경의 update가 늦어지므로 구복호기에 비해 구동시간도 길어지고 전체적인 복잡도 역시 크게 증가한다. 또한 알고리즘의 속도를 줄이는 데에는, 구복호기 구조상 늦게 검출되는 격자 포인트들이 먼저 발견되는 격자 포인트들의 cost function보다 월등히 큰 경우가 많은 점에서도 그 이유를 찾을 수 있다.

다. 복잡도를 줄이기 위한 구반경의 최적화

리스트 구복호기는 APP 검출기에서 구동되고, 이 과정은 IDD 시스템에서의 외부 반복이 이루어지기 전에 종료된다. 그러므로 처음 리스트를 형성할 때엔 인터리버에 의해 모든 비트가 가지는 $L_A(b_k)$ 값이 0이 되고, 리스트 구성에는 영향을 미치지 않는다. 이로 인해 리스트 구복호화가 이루어진 직후의 APP 검출기의 출력값은

$$L_E(b_k)=\ln\frac{\sum_{b_{k+}}\exp\left(-\frac{\|y-Hs_{k+}\|^2}{2\sigma^2}\right)}{\sum_{b_{k-}}\exp\left(-\frac{\|y-Hs_{k-}\|^2}{2\sigma^2}\right)} \quad (7)$$

이 되고, 이 때 굉장히 큰 cost function을 가지는 심볼벡터 혹은 각 분모와 분자의 exponential 값을 작게 하는 격자 포인트들은 LLR 값이나 성능에 아주 작은 영향을 미치게 된다. 이 격자 포인트들까지 모두 검출하는 것은 비효율적인 부분이 많다. 이러한 심볼벡터들은 구반경을 적절히 제한함으로써 리스트를 형성하는 과정에서 제거할 수 있다.

일단 새로운 리스트 구복호기에서는 다음을 만족하는 격자 포인트 s_i들만 검출하도록 가정한다.

$$\rho P(y|s_{ml})\le P(y|s_i) \quad (8)$$

가우시안 잡음 전제 하에선 $P(y|s_i)=\frac{1}{2\sigma^2}exp\left(-\frac{J(s_i)}{2\sigma^2}\right)$이므로, (8) 식은 다음과 같이 정리된다.

$$J(s_i)\le J(s_{ml})-2\sigma^2\ln\rho \quad (9)$$

(8) 식의 전제에 따라 리스트를 형성 하였을 때, 리스트 안의 모든 격자 포인트의 cost function은 (9) 식의 우변보다 작게 되므로 우리가 제안하는 리스트 구복호기의 초기 구반경은 (9) 식의 우변이 된다. 즉,

$$r=J(s_{ml})-2\sigma^2\ln\rho \quad (10)$$

초기 구반경은 (10) 식과 같이 ML-solution의 cost function과 ρ에 의해 결정된다. 이러한 특징 때문에, 본 논문에서 제안하는 리스트 구복호기는 처음엔 구복호기를 통해 ML-solution을 찾은 후에 리스트를

다시 형성하는 2단계 과정을 거치게 된다.

라. ρ의 상한

(8) 식을 전제로 한 리스트 구복호기의 복잡도를 최소하능로 줄이기 위해서는 (10) 식의 초기 구반경이 최소가 되어 가능한 가장 적은 수의 격자 포인트가 반경 안에 포함되도록 해야 한다. 이를 위해서는 ρ값이 최대가 되어야 하는데, 이를 위해서는 몇가지 수학적인 단계가 필요하다. 먼저, (8) 식으로부터

$$\rho P(y|s_{ml}) \le \frac{1}{N}\sum_{i=1}^{N} P(y|s_i) \tag{11}$$

임을 알 수 있다. 여기서 $w_i = \frac{J(s_i)}{\sigma^2}$, $g(x) = \exp\left(-\frac{x}{2}\right)$임을 가정하면 $P(y|s_i) = g(w_i)$이고, (11) 식의 우변은 다음과 같이 간단화 할 수 있다.

$$\frac{1}{N}\sum_{i=1}^{N} g(w_i) \approx \int_{w_{\min}}^{w_{\max}} g(w) f_{\Omega}(w) dw \tag{12}$$

이는 $g(w)$가 일대일 대응이기 때문이고, 여기서 $f_{\Omega}(w)$는 w의 확률함수, $w_{\min} = w_{ml}$, $w_{\max} = w_N$이다. 그리고 실제 전송된 신호벡터 s_t에 대해 $\mu = H(s_t - s)$라고 한다면,

$$w = \frac{\|y - Hs\|^2}{\sigma^2} = \frac{\|H(s_t - s) + v\|^2}{\sigma^2} = \frac{\|\mu + v\|^2}{\sigma^2} \tag{13}$$

인 w는 비중심 카이제곱분포가 된다. 이 비중심 카이제곱분포 함수를 (12) 식과 같이 적분하는 것은 상당히 어려운 작업을 수반하므로, [5]에서 소개된 간단화과정을 거치도록 한다. Patnaik에 의하면, λ와 w의 비중심 카이제곱분포는 상수 c와 K의 자유도를 가진 카이제곱분포의 곱으로 나타낼 수 있다. 이는

$$f_{\Omega}(w) \approx \frac{c}{2^{K/2}\Gamma(K/2)} x^{K/2-1} \exp\left(-\frac{x}{2}\right)$$

(14) 로 나타내고, $c = \frac{v + 2\lambda}{v + \lambda}$, $K = \frac{(v+\lambda)^2}{v + 2\lambda}$ 이다. 이러한 간단화 과정을 거친 뒤, (11), (12) 식과 (14) 식의 관계를 정리하면

$$\rho \le \frac{c}{2^{K/2}}\left(1 - \frac{\gamma(K/2, w_{ml})}{\Gamma(K/2)}\right)\exp\left(\frac{J(s_{ml})}{2\sigma^2}\right) \tag{15}$$

이 되고, 여기서 $\gamma(\bullet)$는 incomplete 감마함수이다. (15) 식을 (10) 식에 대입하면, 마지막으로 (8) 식을 전제한 초기 구반경을 구할 수 있다.

$$r = \sigma^2 \ln\left(\frac{2^{K/2}}{c}\left(\frac{\Gamma(K/2)}{\Gamma(K/2) - \gamma(K/2, w_{ml})}\right)\right) \tag{16}$$
$$= \sigma^2 \ln\left(\frac{2^{K/2}}{c}\left(\frac{\Gamma(K/2)}{\Gamma(K/2) - \gamma(K/2, \frac{J(s_{ml})}{\sigma^2})}\right)\right)$$

기존의 리스트 구복호기와는 달리, 새로 제안된 리스트 구복호기는 항상 고정된 수의 격자 포인트를 포함한 리스트를 구성할 필요가 없고, 높은 SNR일 수록 더 적은 수의 격자 포인트만을 검출하게 된다. 마지막으로, 단순히 초기 구반경만을 조정하지 않고 트리 구조의 각 레벨에서의 검출 효율성을 높이기 위해 Probabilistic tree pruning (PTP) [9] 방식을 더해서 구동하도록 한다.

3. 결론

앞서 구한 초기 구반경에 의한 성능을 체크하기 위하여, 컴퓨터 시뮬레이션을 통해 다른 알고리즘들과의 성능 비교를 하도록 한다.

1) LSD [1] : 충분히 큰 구반경을 잡은 뒤 기존의 구복호기를 구동하도록 한다.

2) Increasing radii algorithm (IRA-LSD) [6] : 큰 반경을 잡고 시작하는 기존의 구복호기와는 달리, 비교적 작은 반경을 잡은 뒤 리스트가 가득 찰 때까지 반경을 서서히 늘려가는 알고리즘이다.

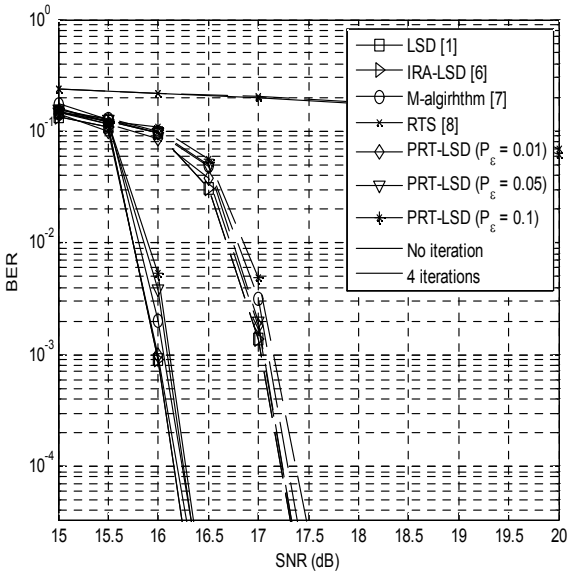

그림 2. 비트에러율의 비교

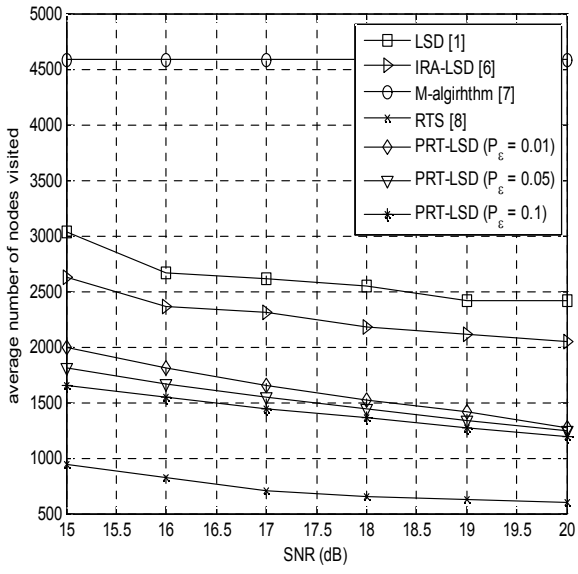

그림 3. 복잡도의 비교

3) M-알고리즘 [7] : 트리 구조에서 각 레벨마다 가장 좋은 M개의 가지만을 남기는 알고리즘이다.

4) Repeated tree search (RTS) [8] : 처음엔 ML-solution, 두 번째 단계에선 그의 binary complement를 찾아 리스트를 구성하는 알고리즘이다.

5) 본 논문에서 제안된 리스트 구복호기

시뮬레이션은 64-QAM 변조방식을 이용하고, Rayleigh 페이딩 특성을 가진 4×4 MIMO 채널에서의 전송을 가정하였다. 또한 코드 비율 1/2의 터보 부호기와 복호기를 이용하였고, 각 코드 블록마다 9,216개의 비트를 생성하였다. 또한 성능으로는 비트에러율을 사용하였고, 복잡도로는 검출 단계에서 방문한 트리 구조상의 모든 노드의 수를 사용하였다. 또한 충분한 반복 이득을 얻기 위해 터보 복호기 안에서는 8번, 그리고 복호기 외부의 반복은 4번으로 하도록 하였다.

그림. 2는 $N=100$으로 가정한 경우의 각 알고리즘들의 비트에러율을 나타낸다. IRA-LSD 방식은 리스트를 구성하는 방법은 다르지만 구성된 리스트는 기존의 LSD 방식과 같게 나타나므로 성능 또한 동일하게 나타났다. 그리고 단 두 개의 격자 포인트만으로 리스트를 구성하는 RTS 기법이 성능면에선 가장 떨어지는 것으로 나타났고, 제안된 방식은 외부 반복 횟수에 상관없이 다른 리스트 구성 기법들에 아주 근접한 성능을 나타낸다. 무엇보다 그림. 2에서 언급된 경우 중 PTP 기법을 더했을 때 가장 성능이 떨어지는 $P_{\epsilon}=0.1$인 경우에서마저도 기존의 기법들과 0.2 dB 이상의 차이는 보이지 않을 정도의 근접성을 확인할 수 있다. 그림. 3을 확인했을 때, 조건에 상관없이 트리 구조의 각 레벨에서 항상 같은 수의 가지만 남기는 M-알고리즘은 항상 고정된 수의 노드를 방문하는 것으로 나타났다. 항상 고정된 복잡도라는 점에서 유리한 점을 가지지만, 그 수는 비교된 기법들 중에선 가장 큰 것으로 나타났다. 이에 비해 구복호기 기반 기법들은 이에 비해 큰 폭으로 감소한 복잡도를 나타내며, 특히 본 논문에서 제안된 방식은 그 중에서 가장 작은 복잡도를 가진 것으로 확인할 수 있다. RTS 방식은 다른 기법들에 비해 월등히 뛰어난 복잡도를 가지는 것으로 나타났지만, 이는 단 두 개만으로 리스트를 구성하는 특징 때문이고, 오히려 이 특징 때문에 성능의 차이가 더욱 크게 나타났다. 이는 실제로 구현하기는 용이한 특징에 비해 그 성능은 크게 기대할 수 없는 단점을 나타낸다.

본 논문에서 저자들은 IDD 시스템에서의 APP 검출기 내부에서 계산상의 효율성을 높이고 전체 복잡도를 감소시킬 수 있는 리스트 생성 기법을 제안하였다. 리스트 구복호기가 항상 고정된 크기의 리스트만을 생성할 필요는 없다는 점에 착안하여, 불필요한 격자 포인트를 미리 제거할 수 있는 최적화된 초기 구반경을 설정하여 *extrinsic* LLR 값을 좀 더 간단히 구할 수 있도록 하였고, 또한 검출 단계에서 트리 구조의 각 레벨 상에서의 효율성을 높이기 위하여 PTP 방식 역시 합쳐진 기법을 제안하였다. 초기 구반경의 최적도는 시뮬레이션을 통한 비트에러율을 비교함으로서 일정 수준 이상의 최적도를 확인할 수 있었고, 또한 이를 통해 아주 작은 성능의 감소만으로 아주 큰 폭으로 감소된 복잡도를 확인할 수 있었다.

감사의 글

이 연구에 참여한 연구자의 일부는 '2단계 BK21사업' 지원비를 받았음.

참 고 문 헌

[1] B. M. Hochwald and S. ten Brink, "Achieving near-capacity on a multipleantenna channel," *IEEE Trans. Commun.*, vol. 51, pp. 389-399, March 2003.

[2] J. Hagenauer, E. Offer and L. Papke, "Iterative decoding of binary and block convolutional codes," *IEEE Trans. Inform. Theory*, vol. 42, pp. 429-445, March 1996.

[3] E. Viterbo and J. Boutros, "A universal lattice code decoder for fading channels," *IEEE Trans. Inform. Theory*, vol. 45, pp. 1639-1642, July 1999.

[4] L. Babai, "On Lov´asz' lattice reduction and the nearest lattice point problem," *Combinatorica*, vol. 6, pp. 1-13, 1986.

[5] P. B. Patnaik, "The noncentral $\hat{A}2$ and F-distributions and their applications," *Biometrika*, vol. 36, pp. 202-232, 1949.

[6] R. Gowaikar and B. Hassibi, "Statistical pruning for near-maximum likelihood decoding," *IEEE Trans. Signal Process.*, vol. 55, pp. 2661-2675, June 2007.

[7] Yvo L. C. de Jong and T. J. Willink, "Iterative tree search detection for MIMO wireless systems," *IEEE Trans. Commun.*, vol. 53, pp. 930-935, June 2005.

[8] C. Studer, A. Burg and H. BÄolcskei, "Soft-output sphere decoding: algorithms and VLSI implementation," *IEEE Journal on Selected Areas in Commun.*, vol. 26, pp. 290-300, Feb. 2008.

[9] B. Shim and I. Kang, "Sphere decoding with a probabilistic tree pruning," *IEEE Trans. Signal Process.*, vol. 56, pp. 4867-4878, Oct. 2008.