Suboptimal MIMO Detector based on Viterbi Algorithm

Jin Lee and Sin-Chong Park
System Integration Technology Institute
Information and Communications University
119, Munji-ro, Yuseong-gu, Daejeon
Republic of Korea

Abstract: - Suboptimal detectors of multiple-input multiple-output (MIMO) have been studied because the implementation of the optimum detector, the maximum-likelihood (ML) detector, has so far been considered infeasible for high-rate system. Sphere decoder (SD) using depth-first tree searching and K-best algorithm are used for near optimum detector. SD has the non-deterministic computational throughput and K-best requires the sorting unit whose complexity is significantly high when a large K is used together with high modulation constellation. In this paper, we propose a suboptimal MIMO detector employing Viterbi algorithm instead of tree searching. This detector can keep the computational throughput constant and reduce the complexity because the sorting is not required. In the simulation, we analyze the advantage and the drawback of the proposed detector in the environment of IEEE 802.11n system.

Key-Words: - MIMO system, Viterbi detection, Sphere decoder.

1 Introduction

A large number of works on the physical-layer study of MIMO techniques have been done in the past decade. The optimal MIMO detection, maximum likelihood (ML) decoder, is infeasible by its complexity when a large number of antennas are used together with higher modulation constellation [1]. The efficient algorithms for reducing complexity of ML have been proposed such as linear detectors, sphere decoder (SD) and K-best algorithm. Linear detectors are based on principles of minimum mean-square error (MMSE) and zero-forcing (ZF). Although they can reduce the complexity dramatically, their performances are significantly degraded. Certain techniques such as soft interference cancellation can improve the performance. To achieve the near optimal performance, SD based on non-exhaustive tree searching has been researched.

In general, suboptimal detectors using tree searching are divided into two types, depth-first SD and breadth-first using M-algorithm, also known as K-best [2]. Depth-first SD can obtain better performance than K-best detection with small K. In spite of this penalty, K-best detection has the advantage in terms of the fixed computation throughput and latency while those of depth-first SD vary randomly. For the soft-output detector, when iterative detection or soft decoder is used, depth-first SD is advanced to list sphere decoder (LSD), and it degrades the computational throughput severely [3]. The throughput of K-best detection is not decreased even though it provides the soft-output. In K-best detection, the nodes at a depth must be sorted according to their metrics and the complexity of sorting is significantly high when a large K is used together with high modulation constellation.

In this paper, we propose the Viterbi algorithm based suboptimal detector which converts tree searching to trellis searching. The trellis searching in Viterbi algorithm is truncated version of breadth-first tree searching, hence the performance is degraded. However, the proposed detector overcomes the weak point of depth-first SD because it can guarantee the constant throughput, and it can ease the pressure of sorting in K-best algorithm.

This paper consists of the following contents. In Section 2, the fundamentals of depth-first SD, K-best algorithm and soft-output generation are summarized. In Section 3, the proposed Viterbi algorithm based MIMO detector is described. The error performance in the environment of IEEE 802.11n system is simulated in Section 4. We conclude the paper in Section 5.

2 Fundamentals of MIMO Detectors

2.1 System Model

We consider a spatial multiplexing MIMO system with N_t transmit and N_r receive antennas. The transmitter sends N_t spatial streams. Assume that the transmitted symbol is taken from a Gray-labeled M-QAM constellation $\left(M = 2^q\right)$. At once, the transmitter maps one $qN_t \times 1$ coded bit vector $\mathbf{x}$ onto a $N_t \times 1$ symbol vector $\mathbf{s}$. The transmission of each vector $\mathbf{s}$ over MIMO channels can be modeled as $\mathbf{y} = \mathbf{Hs} + \mathbf{n}$, where $\mathbf{y}$ is a $N_r \times 1$ vector of received signals. $\mathbf{H}$ is a $N_r \times N_t$ MIMO channel matrix which is i.i.d. zero-mean unit variance complex Gaussian matrix, perfectly known to the receiver. $\mathbf{n}$ is a vector of independent zero-mean complex Gaussian noise entries with variance $N_0/2$.

2.2 Depth-first Sphere Decoder

Depth-first SD algorithm can be divided into two parts, the first part is the PED calculation and the other part is the tree traversal. The following inequality expresses the sphere constraint [1]:

$$\left\|\mathbf{y} - \mathbf{Hs}\right\|^2 < r^2 . \qquad (1)$$

The left part of inequality (1) can be decomposed with PEDs as

$$\left\|\hat{\mathbf{y}} - \mathbf{Rs}\right\|^2 = \sum_{i=1}^{N_t} \left| \hat{y}_i - \sum_{j=i}^{N_t} r_{ij} s_j \right|^2 = \sum_{i=1}^{N_t} T_i\left(\mathbf{s}^i\right), \qquad (2)$$

where $\hat{\mathbf{y}} = \mathbf{Q}^H \mathbf{y} = \mathbf{Rs}^{ZF}$, $\mathbf{Q}$ and $\mathbf{R}$ are the results of QR decomposition of $\mathbf{H}$, and r_{ij} is the element of the upper triangular matrix $\mathbf{R}$. $\hat{y}_i$ and s_j are the element of $\hat{\mathbf{y}}$ and $\mathbf{s}$, respectively. $T_i\left(\mathbf{s}^i\right)$ is called PED.

An efficient depth-first tree searching is operated with PED calculations of whole children nodes of a mother node. Fig. 1 shows the simple example of tree searching and tree traversal of depth-first SD when $N_t = 3$ and $M = 4$. The number in each circle (node) is the accumulated PED of the node. In this paper, tree level i is presented in descending order. A dotted rectangle means the nodes which are calculated at the same time. Dark gray nodes are pruned nodes because the accumulated PEDs of their mother nodes already exceed the radius r.

A considerable problem of depth-first SD is the fact that the searching complexity critically depends on the a-priori choice of the radius [4]. If r is chosen too small, no solution is found, and the search must be restarted with a larger radius. If it is chosen too large, many candidate vector symbols lie within the sphere, and the detection effort is high. A technique known as radius reduction allows us to avoid this problem if the depth-first SD is used in conjunction with Schnorr-Euchner (SE) enumeration [5]. However, the detection effort varies randomly according to the received SNR and channel matrix, so the computational throughput is non-deterministic. In practice, the maximum detection effort must be limited to depth-first SD because the effort to find the solution sometimes even corresponds to an exhaustive search.

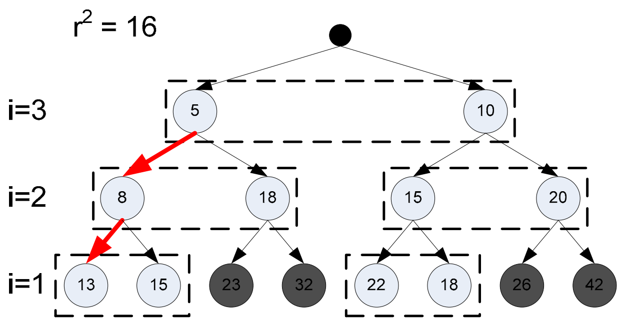

Fig. 1. Example of depth-first SD

2.3 K-best Algorithm

The K-best algorithm traverses the tree in breadth-first manner. The detector selects only K nodes at each level of the tree which have the minimum accumulated PEDs and computes the PEDs of all their children. Among these children, it selects the K nodes with the smallest PEDs as the parent nodes to be visited at the next level. This algorithm is designed such that the detection effort is constant for each transmitted vector. K-best have better parallelism than depth-first SD.

It is clear that K determines the trade-off among silicon area, throughput and the error performance of the system. Computational complexity for PED grows with increasing K and the sorting complexity to select K nodes also increases with larger K and M. K-best detection with small K has the worse error performance than depth-first SD.

2.4 Soft-output MIMO Detector

Soft-output MIMO detector provides a posteriori probability (APP) information for each coded bit.

The log-likelihood ratio (LLR) of a posteriori is defined as

$$L_D(x_k \mid \mathbf{y}) = \ln \frac{\Pr[x_k = 1 \mid \mathbf{y}]}{\Pr[x_k = 0 \mid \mathbf{y}]} = L_A(x_k) + L_E(x_k \mid \mathbf{y}), \quad (3)$$

where $k = 0, \ldots, qN_t - 1$.

$L_A(x_k)$ is a priori log-likelihood ratio and $L_E(x_k \mid \mathbf{y})$ is an extrinsic log-likelihood ratio.

If we assume that $\Pr[x_k = 1] = \Pr[x_k = 0]$, the extrinsic log-likelihood ratio can be approximated as [6]

$$L_E(x_k \mid \mathbf{y}) \approx -\frac{1}{N_0}\left[\min_{\mathbf{x}\in X_{k,1}}\left(\|\mathbf{y}-\mathbf{Hs}\|^2\right) - \min_{\mathbf{x}\in X_{k,0}}\left(\|\mathbf{y}-\mathbf{Hs}\|^2\right)\right], \quad (4)$$

where $X_{k,b} = \{\mathbf{x} \mid x_k = b\}$, $b = 0,1$.

Computing $L_E(x_k \mid \mathbf{y})$ requires exhaustive search to find the minimum values for each x_k because there are 2^{qN_t-1} candidates for each term. In order to reduce the complexity, the searching sets to select the minimum values are changed from $X_{k,b}$ to $L_{k,b}$, where L is the list which includes a certain number of candidates which have the smallest distances [6]. Equation (4) can be approximated as

$$L_E(x_k \mid \mathbf{y}) = -\frac{1}{N_0}\left[\min_{\mathbf{x}\in L_{k,1}}\left(\|\mathbf{y}-\mathbf{Hs}\|^2\right) - \min_{\mathbf{x}\in L_{k,0}}\left(\|\mathbf{y}-\mathbf{Hs}\|^2\right)\right], \quad (5)$$

where $L_{k,b} = \{\mathbf{x} \mid x_k = b,\ \mathbf{x} \in L\}$, $b = 0,1$.

In depth-first SD, A simple modification to the SD helps us to generate the list, but degrades the computational throughput severely [3]. This modification is called to list sphere decoder (LSD). LSD does not decrease radius and add all searched points to list if the list is not full, or if the list is full, it compares the searched point in the list with the largest radius and replaces this point if the new point has smaller radius. In K-best detection, we can generate the list of K candidates without any degradation of throughput.

3 Proposed MIMO Detection Algorithm

In this paper, we propose the suboptimal MIMO detector using Viterbi algorithm instead of tree searching. In Fig. 2, the original tree structure and trellis structure for Viterbi algorithm are showed.

In the trellis structure, there are N_t depths and M stages at each depth. Every stage has M outputs toward every stage at the next depth and *M* inputs from every stage at the previous depth. *M* outputs represent the path metrics which are accumulated PEDs of all children nodes of the current stage.

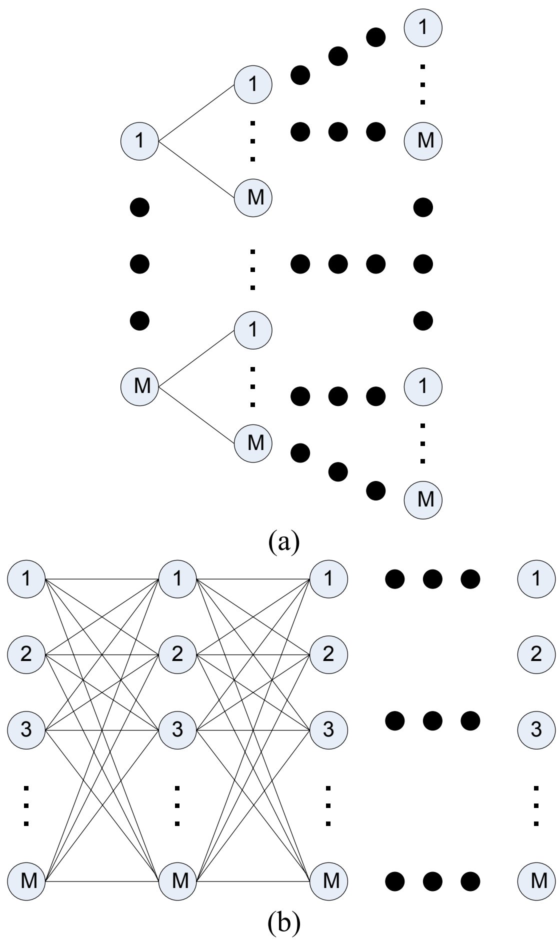

Fig. 2. (a) Original tree structure and (b) Trellis structure

From the equation (2), we re-formulate the three parameters, the initial values, branch metric and path metric for the trellis searching.

The initial values of the first depth are expressed as

$$P_{N_t,i} = \left|\hat{y}_{N_t} - r_{N_t,N_t} s_{N_t,i}\right|^2, \ i = 1,2,\cdots,M \quad (6)$$

where $s_{n,i}$ represents the symbol of the stage *i* on the depth *n*.

The branch metric of the stage *j* at the next depth *n*-1 from the stage *i* at the current depth *n* can be written as

$$B_{n,i,j} = \left|\hat{y}_{n-1} - \sum_{m=n}^{N_t} r_{n-1,m} s_m - r_{n-1,n-1} s_{n-1,i}\right|^2 \quad (7)$$

$$, \ i = 1,2,\cdots,M \ \ and \ \ n = N_t, N_t - 1, \cdots, 1$$

where s_m is the symbol in the memorized path history.

In this paper, the depth *n* is presented in descending order. The branch metric corresponds to PED.

From the branch metric, the path metric of the stage j at the next depth n-1 can be written as

$$P_{n-1,j} = \min_i \left(P_{n,i} + B_{n,i,j} \right) \quad (8)$$

Among M pairs of previous path metric and branch metric, every stage selects the minimum path metric and memorizes its path. At the final depth, we can obtain M candidates and their distances. For hard-output detection, the candidate with the smallest distance at the final depth becomes the solution. For soft-output detection, M candidates make up the list for generating soft-output as in (5).

The effort required to find the solution of depth-first SD varies randomly according to the realization of the channel and noise and sometimes even corresponds to an exhaustive search. However, the proposed detector can keep the computation throughput constant and it can generate output at every cycle if computation is fully parallelized and pipelined.

In terms of the complexity, the proposed detector is more efficient than K-best because the sorting is not required. If PED computation unit is implemented as described in [1], which can compute PEDs of all children at one time, the number of PED computation is $K \cdot (N_t - 1) + 1$ for K-best and $M \cdot (N_t - 1) + 1$ for the proposed detector. The number of compare operations is $\left((K \cdot M)^2 - K \cdot M \right) / 2$ for K-best based on the sorting network in [8] and $M \cdot (M - 1)$ for the proposed detector because only searching the minimum is required instead of the sort as shown in equation (8). Table I compares the complexity between K-best and the proposed detector in terms of the number of comparisons for a level. The proposed detector can reduce the complexity, significantly.

Fig. 3 shows an example of the proposed MIMO detection when four transmit antenna and 4 QAM are used. The thick lines represent the selected paths and the thick and dotted lines represent the path of the solution for hard-output detection. The number on each stage is the path metric and the binary bits in the bracket are the bits of the survived symbols. The decision bits are the reverse order of the survived symbols.

Table 1. Number of compare operations for K-best and the proposed detector

	4-QAM (K=4)	16-QAM (K=16)	64-QAM (K=64)
K-best	112	32,512	8,384,512
Proposed	12	240	4032

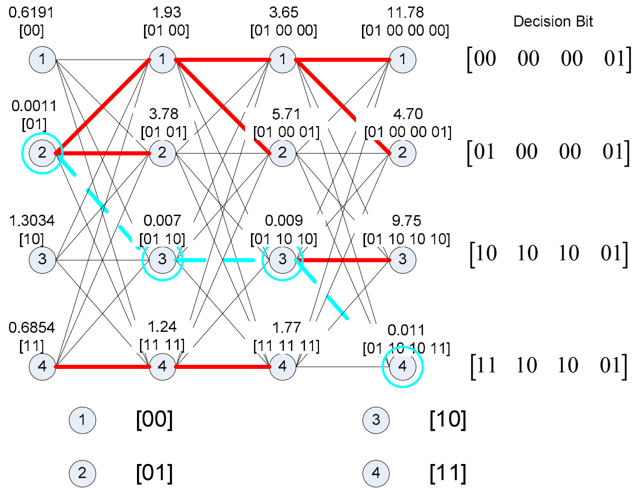

Fig. 3. Example of the proposed MIMO detection

3 Simulation Result

In this section, the error performance of the proposed MIMO detection is compared with depth-first SD, K-Best and ZF detector.

In this simulation, the target system is IEEE 802.11n [7] which employs a spatial multiplexing, convolutional code and orthogonal frequency division multiplexing (OFDM) with up to 64-QAM modulation. Only 20 MHz bandwidth is considered in this simulation. Iterative detection and decoding is not considered, but soft Viterbi decoder for convolutional code is used.

Frequency selective MIMO channel with three delay taps is considered. It has 50 ns of the delay resolution, exponential power delay profile and no spatial correlation. Since the block fading channel is assumed, the channel coefficients of each tap are constant during the duration of a packet transmission and changed randomly between two consecutive packet transmissions. The length of packet is 1000 bytes.

Fig. 4 shows the error performance of the proposed detector against other MIMO detectors. In this simulation, the modulation and coding scheme (MCS) is 16-QAM and convolutional code with code rate of 1/2. The number of transmit antenna (N_t) is four. The size of the list for soft-output detector is 16 so that K is also 16.

Table 2. Summary of the comparison of the error performance

Output	Type	SNR (dB)	Gain (dB)
Hard	ZF	25.4	4.9
	Depth-first	19.8	-0.7
	K-best	19.8	-0.7
	Proposed	20.5	0
Soft	ZF	22.4	3.2
	Depth-first	17.2	-2
	K-best	17.5	-1.7
	Proposed	19.2	0

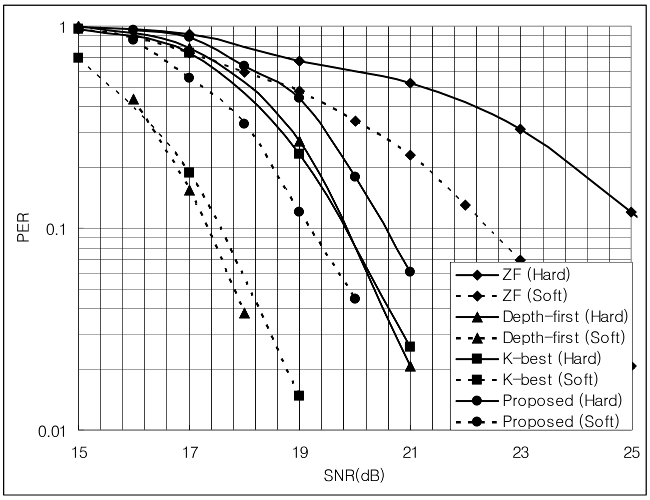

Fig. 4. Error performance in IEEE 802.11n system

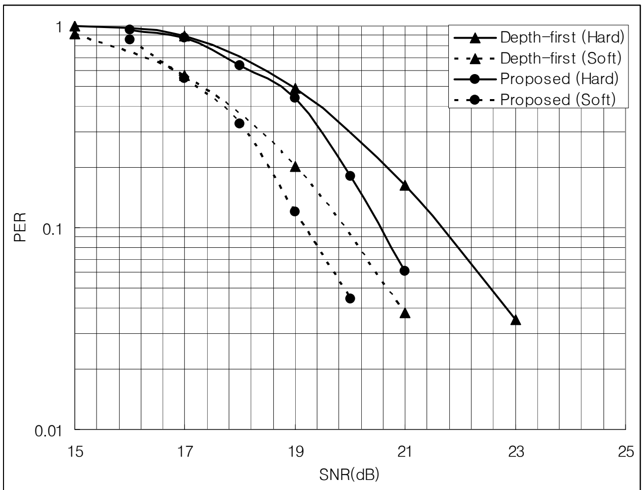

Fig. 5. Comparison of the error performance with limited depth-first SD

Table 2 summarizes the comparison of the error performance at the packet error rate of 10^{-1}. In this simulation, the proposed MIMO detector performs worse than depth-first and K-best about 0.7 dB in hard-output detection and 2 dB in soft-output detection. This is obvious because the proposed detector can lost the ML solution in the early depth.

In practice, the maximum detection effort must be limited in depth-first SD because the effort required to find the solution of depth-first SD varies randomly and sometimes even corresponds to an exhaustive search. Fig. 5 shows the comparison of the error performance between the proposed detector and the limited depth-first SD when MCS is 27 in IEEE 802.11n. The limited depth-first SD has the real-time constraint by early termination. The detection effort, the number of visited nodes, of depth-first SD with the architecture described in [1] and [3] does not exceed 16 for hard-output 50 for soft-output. The proposed detector can have better performance than the limited depth-first SD.

4 Conclusion

This paper presents a suboptimal hard/soft-output MIMO detection algorithm with fixed throughput and low complexity. The basic idea is to re-formulate the tree structure to the trellis structure similar as Viterbi algorithm. Since the trellis truncates the tree, the trellis structure of the proposed detector causes performance degradation. However, in terms of complexity, the proposed detector is more efficient than K-best, and guarantees the constant throughput while the throughput of depth-first SD varies randomly. In addition, the error performance is better than the constrained depth-first SD for fixed throughput.

Acknowledgement

This work was partly sponsored by ETRI SoC Industry Promotion Center, Human Resource Development Project for IT SoC Architect.

References:

[1] A.Burg, et al., “VLSI Implementation of MIMO Detection Using the Sphere Decoding Algorithm,” *IEEE Jour. Of Solid-State Circuits,* vol 40, Issue 7, pp. 1566-1577, July 2005.

[2] Z. Guo and P. Nilsson, “VLSI implementation issues of lattice decoders for MIMO systems,” *IEEE International Symposium on Circuits and Systems,* May 2004, pp. IV-477-IV480.

[3] Jin Lee and Sin-chong Park, “A pipelined VLSI architecture for a list sphere decoder,” *IEEE International Symposium on Circuits and Systems,* May 2006, pp 397-400

[4] Q. Liu and L. Yang, "A Novel Method for Initial Radius Selection of Sphere Decoding," *IEEE Vehicular Technology Conference-Fall,* vol. 2, Sep. 2004, pp. 1280-1283.
[5] C. P. Schnorr and M. Euchner, "Lattice basis reduction: Improving practical lattice basis reduction and solving subset sum problems," *Math. Program.*, vol. 66, pp. 181-194, 1994.
[6] B. M. Hochwald and S. ten Brink, "Achieving near-capacity on a multiple-antenna channel," *IEEE Trans. on Commun.,* vol. 51, no. 3, pp. 389-399, March 2003.
[7] Sean Coffey and Assaf Kasher and Adrian Stephens, "Joint Proposal: High throughput extension to the 802.11 Standard: PHY," *IEEE 802.11n,* Jan. 2006.