Implementation Aspects of List Sphere Detector Algorithms

Markus Myllylä and Markku Juntti
Centre for Wireless Communications
P.O. Box 4500, FIN-90014 University of Oulu, Finland
{markus.myllyla, markku.juntti}@ee.oulu.fi

Joseph R. Cavallaro
Dept. of Electrical & Computer Engineering
Rice University, Houston, TX 77251-1892, USA
cavallar@rice.edu

ABSTRACT

A list sphere detector (LSD) can be used to approximate the optimal maximum a posteriori (MAP) detection. The total complexity of the LSD algorithms is relative to the number of visited nodes in the search tree. We compare the differences between real and complex signal model in the LSD algorithm implementation and study its impact on the complexity and performance with different search strategies. In hardware implementation, the number of visited nodes needs to be bounded in order to determine the complexity and the latency of the implementation. Thus, we study the performance of LSD algorithms with a limited number of nodes in the search. We show that the algorithms with real signal model are less complex compared to the complex signal model, and that the performance may suffer significantly with limited search depending on the search strategy.

I. INTRODUCTION

Orthogonal frequency division multiplexing (OFDM) [1] has become a widely used technique to significantly reduce receiver complexity in broadband wireless systems. Multiple-input multiple-output (MIMO) channels offer improved capacity and significant potential for improved reliability compared to single antenna channels [2]. MIMO techniques in combination with OFDM (MIMO-OFDM) have been identified as a promising approach for high spectral efficiency wideband systems [3].

The optimal detection for coded system would require the use of a maximum a posteriori (MAP) detection. However, the computational complexity of optimal MAP detection is very high, and such an approach is not feasible for systems with high number of antennas and high order modulation. A suboptimal approach is to use linear zero forcing (ZF) or minimum mean square error (MMSE) criterion based detectors [4]. However, their performance can be rather poor in bad channel conditions, i.e., when the eigenvalue spread of the channel matrix is large [5]. Sphere detector (SD) calculates the maximum likelihood (ML) solution with reduced complexity compared to full-complexity ML detectors [6], [7]. A list sphere detector (LSD) [8] is a variant of the sphere detector that can be used to approximate MAP detection.

In this paper, we consider three LSD algorithms with different search strategies, and study and compare the implementation aspects related to the algorithms. The considered algorithms are the K-best-LSD algorithm [9], [10], the Schnorr-Euchner enumeration (SEE)-LSD algorithm [10], [11], and the increasing radius (IR)-LSD algorithm [12], [13], [14]. The total complexity of the LSD algorithms is relative to the number of visited nodes in the search tree. The size of the search tree depends on the applied signal model. Usually a real signal model is assumed with sphere detection. However, the dimensions of the search tree structure can be decreased by applying a complex signal model [8], [15]. We study and analyze the differences between the real and complex signal model and its effect on the complexity. In hardware implementation, the number of visited nodes needs to be bounded in order to determine the complexity and the latency of the implementation. Thus, we consider the performance of LSD algorithms with limited number of nodes in the search.

The paper is organized as follows. The system model, sphere detection, and the detection algorithms are presented in Section II. The implementation aspects are analyzed and discussed in Section III. Performance examples are presented in Section IV. Summary is presented in Section V.

II. SYSTEM MODEL

An OFDM based multiple antenna system with $N_T$ transmit antennas and $N_R$ receive antennas is considered with the assumption $N_R \geq N_T$. A vertical Bell Labs’ layered space-time (VBLAST) architecture [16] with quadrature amplitude modulation (QAM) is applied to the system. A block diagram of the system is shown in Figure 1. The received signal can be expressed in terms of code symbol interval as

$$y_p = H_p x_p + n_p, \quad p = 1, 2, \ldots, P,$$

where $P$ is the number of subcarriers, the received signal vector $y_p \in \mathbb{C}^{N_R \times 1}$, the transmit symbol vector $x_p \in \mathbb{C}^{N_T \times 1}$.
and the noise vector $\eta_p \in \mathbb{C}^{N_R \times 1}$ are defined in the frequency domain. The elements of $\eta_p$ are independent and complex Gaussian with equal power real and imaginary parts, i.e., $\eta_p \sim \mathcal{CN}(0, N_0 I_{N_R})$ and represent the frequency domain thermal noise at the receiver. The channel matrix $H_p \in \mathbb{C}^{N \times N_T}$ contains complex Gaussian fading coefficients with unit variance. The entries of $x_p$ are chosen independently from a complex QAM constellation $\Omega$ with $Q$ bits per symbol, i.e., $|\Omega| = 2^Q$. The set of all possible transmitted symbol vectors is denoted by $\Omega^{N_T}$. The corresponding uncoded transmission rate is $R = M_T Q$ bits per channel use (bpcu).

A. ML detector

The ML detector solves optimally the so-called closest lattice point problem by calculating the Euclidean distances (EDs) between received signal $y$ and lattice points $Hx$ and selects the lattice point that minimizes the Euclidean distance to receive vector $y$, i.e.,

$$\hat{x}_{ML} = \arg \min_{x \in \Omega^{N_T}} ||y - Hx||_2^2. \quad (2)$$

For simplicity, the subindices are omitted in (2) and in the sequel. The ML detection problem can be solved with an exhaustive search, i.e., checking all the possible symbol vectors and selecting the closest point. However, this is computationally very complex and not feasible as the set of possible points $\Omega^{N_T}$ increases.

B. Sphere detector

The SD algorithms achieve the ML solution in (2) with reduced number of considered candidate symbol vectors in the search. This is done by limiting the search to points that lie inside a $N_R$-dimensional hyper-sphere $S(y, \sqrt{C_0})$ centered at $y$. The condition can be written as [7]

$$||y - Hx||_2^2 \leq C_0, \quad (3)$$

where $C_0$ is the squared radius of the sphere. The channel matrix $H$ can be decomposed by QR decomposition (QRD) and then (3) can be written as

$$||y - QRx||_2^2 \leq C_0 \quad (4)$$
$$||Q^H y - Rx||_2^2 \leq C_0 \quad (5)$$
$$||\tilde{y} - Rx||_2^2 \leq C_0, \quad (6)$$

where $R \in \mathbb{R}^{N \times N_T}$ is an upper triangular matrix with positive diagonal elements, $Q \in \mathbb{R}^{N \times N_R}$ is a orthogonal matrix, and $\tilde{y} = Q^H y$.

Let $x^{N_T} = (x_i, x_{i+1}, \ldots, x_{N_T})^T$ denote the last $N_T - i + 1$ components of the vector $x$. The sphere search can be illustrated with a tree structure, where the root layer corresponds to $x^{N_T}$. Due to the upper triangular form of $R$ the values of $x$ can be solved from (6) level by level using the back-substitution algorithm. First, the last elements of the possible symbol vectors are calculated, i.e., $x_{N_T}$ and then $x_{N_T-1}$ and so on. The squared partial ED (PED) of $x^{N_T}$ can be calculated as [15]

$$d(x^{N_T}) = d(x^{N_T+1}) + |\tilde{y}_i - \sum_{j=i}^{N_T} r_{i,j} x_j|^2, \quad (7)$$

where $d(x^{N_T}) = 0$, $r_{i,j}$ is the $(i,j)$th term of $R$ and $i = N_T, \ldots, 1$. The next admissible node for partial candidate $x^{N_T+1}$ can be determined by calculating all $|\Omega|$ possible PEDs of $x^{N_T}$ and selecting the lowest one or by applying an enumeration method to determine the next admissible node [7]. The Schnorr-Euchner enumeration (SEE) [11] determines the next best admissible node, and, thus, the search can be started initially with $C_0 = \infty$. Depending on the search strategy and the channel realization the SD searches a variable number of nodes in the tree structure, and selects the point $x = x^{N_T}$, also called a leaf node, for which the ED $d(x^{N_T})$ is minimum.

C. List sphere detector

The SD algorithms give the ML solution as an output. However, the performance of a channel coded system may suffer significantly with ML detector compared to the optimal MAP detector. The list sphere detector (LSD) [8] can be used for obtaining a list of the most probable candidate symbol vectors $L \in \mathbb{Z}^{N_{cand} \times N_T}$ as an output, where $N_{cand}$ is the size of the candidate list so that $1 \leq N_{cand} \leq 2^Q N_T$. The list can then be used to approximate the MAP solution. Depending on the list size $N_{cand}$, it provides a tradeoff between the performance and the computational complexity. In this paper, we consider three LSD algorithms with different search strategies.

1) K-best-LSD: The K-best-LSD algorithm [10] is a modification from K-best-SD algorithm [9] to LSD algorithm. The algorithm is based on the breadth first strategy, i.e., the search proceeds one level at a time by calculating the PEDs of the admissible nodes and selecting the $K = N_{cand}$ best candidates to the next level. Then the search is continued with admissible nodes of the selected $K$ partial candidates. The algorithm search goes through a fixed number of branches in the tree structure if no enumeration is applied. However, it should be noted that the output of the algorithm is not necessarily the candidates with the lowest EDs.

2) SEE-LSD: The SEE-LSD algorithm [10] is a depth first search strategy based algorithm. The search proceeds with the next best admissible node defined by SEE [11] for given partial candidate until a leaf level is reached or the sphere radius is exceeded. The algorithm starts with infinite radius and the radius is updated if a new leaf is reached and the final list is full. The depth first search is based on sequential search with variable throughput. The output of the algorithm is the candidates with lowest EDs.

3) IR-LSD: The increasing radius (IR)-LSD algorithm [14] is a modification of Dijkstra’s algorithm [12] to the LSD algorithm. The algorithm is based on metric first search strategy and it is optimal in the sense of visited number of nodes in the tree structure [12, 13, 17]. The search proceeds by calculating one branch extension at a time and stores the
partial candidate to a stack memory. Then the search is always continued with the partial candidate with the lowest PED. The output of the algorithm is the candidates with lowest EDs.

III. IMPLEMENTATION ASPECTS

The main complexity of the LSD algorithm is in the expand loop of a partial candidate, i.e., calculating the next node for the partial candidate. The operations required to expand a node in the tree depend on the applied algorithm and the search strategy. The K-best-LSD [10] requires the calculation of PED in (7) and sorting between layers. Also enumeration can be applied, but it requires a predefined \( C_0 \). Both IR-LSD [14] and SEE-LSD [10] require the calculation of PED and SEE to expand a node. The IR-LSD also requires memory to store the partial candidates and a min-search operation [14], [17].

A. Real and complex signal models

The total complexity of the LSD algorithms is relative to the number of visited nodes in the search tree. The size of the search tree depends on the applied signal model. The real and LSD algorithms are often assumed with a real equivalent system model. The complex MIMO system model in (1) can be reduced into an equivalent real model as follows

\[
\begin{bmatrix}
\text{Re}(r) \\
\text{Im}(r)
\end{bmatrix} = \begin{bmatrix}
\text{Re}(H) - \text{Im}(H) \\
\text{Im}(H) \text{ Re}(x)
\end{bmatrix} \begin{bmatrix}
\text{Re}(x) \\
\text{Im}(x)
\end{bmatrix} + \begin{bmatrix}
\text{Re}(\eta) \\
\text{Im}(\eta)
\end{bmatrix}.
\]

(8)

Let us define the new real dimensions \( M_T = 2N_T \), \( M_R = 2N_R \). The real symbol alphabet is now \( \Omega_R = \mathbb{Z} \), e.g., \( \Omega_R \in \{-3, -1, 1, 3\} \) in the case of 16-QAM.

The use of a real system model doubles the depth of the search tree, but decreases the number of branches at each level compared to the complex signal model. The total number of possible branches with real signal model is \( B_R = \sum_{i=1}^{M_R} |\Omega_R|^i \), and with complex signal model \( B_C = \sum_{i=1}^{N_R} |\Omega|^i \). The number of branches in the search tree increases with real signal model, and, e.g., in \( 4 \times 4 \) system with 16-QAM \( B_C = 69904 \) and \( B_R = 87380 \). Thus, it is likely that the number of visited nodes by the K-best-LSD algorithm increases with real signal model. It should be noted that the difference in the total number of branches decreases as the number of transmit antennas \( N_T \) and the size of the constellation \( \Omega \) increases. However, it should be noted that the complexity of the PED calculation in (7) is different with real valued signals and complex valued signals. Equation (7) includes multiplications and subtractions, and, the complex valued calculation is approximately three times the complexity of the real valued calculation.

The LSD algorithm needs to determine the next admissible node for the current partial candidate at each level in the search process. In the K-best-LSD this can be done by calculating all the PEDs of the admissible nodes and selecting the minimum ones. The Schnorr-Euchner enumeration (SEE) [11] is applied for finding the next best node effectively with the SEE-LSD and the IR-LSD. The real valued SEE is simpler and less complex compared to the complex valued SEE.

1) Real valued SEE: The SEE middle point, i.e., the best node to proceed with, can be solved with real representation for partial candidate \( x_{i+1} \) as follows

\[
S_i(x_{i+1} | x_i) = \frac{1}{r_{i+1}} (\tilde{y}_i - \sum_{j=i+1}^{M_R} r_{i,j} x_j)
\]

(9)

where \([ \cdot ]\) denotes rounding to the closest integer. The symbols can be ordered into sequence according to their PEDs [7].

2) Complex valued SEE: Hochwald and ten Brink [8] proposed a scheme to compute boundaries of admissible intervals for complex valued constellations. Burg et al. [15] proposed a modification of the scheme in [8] which results in a low complexity VLSI implementation. The method can be applied to calculate SEE for phase shift keying (PSK) and quadrature amplitude modulation (QAM). However, the method is still much more complex compared to the real valued SEE.

It should also be noted that some additional information is needed when applying SEE to define the next admissible node for partial candidate. The information of how many nodes have already been checked from the current candidate is needed, and, e.g., this information may be stored to memory.

B. Limited search

As previously mentioned, the number of visited nodes depends on the search strategy and the signal model, and with the IR-LSD and the SEE-LSD algorithms the number varies depending on the channel realization. The number of visited nodes with the K-best-LSD is fixed due to breadth first search strategy, but the performance may suffer in bad channel realization. In hardware implementation, the number of visited nodes needs to be bounded in order to determine the complexity and the latency of the implementation. Thus, we need to study how the different algorithms perform when the number of nodes to be visited is limited in the search process.

IV. PERFORMANCE EXAMPLES

In this section the performance examples are presented. We compare the impact of real and complex signal models on the complexity and performance of LSD algorithms. Also the performance of the IR-LSD and the SEE-LSD algorithms with limited search is studied. In the computer simulations, a \( 4 \times 4 \) MIMO-OFDM system was assumed with 512 subcarriers (300 used) according to 3G long term evolution (LTE) parameters [18]. A turbo coded VBLAST architecture with 1/2 code rate was applied in a typical urban (TU) 6 tap channel with a user velocity of 120 kmph. The system was operating with 5 MHz bandwidth at a carrier frequency of 2.4 GHz. Soft outputs of the LSD were decoded in a turbo decoder with 8 iterations.

A. Real and complex signal models

The number of visited nodes by the K-best-LSD algorithm depends on the signal model and the applied list size \( N_{\text{cand}} \) [10]. We studied the performance of real and complex algorithms in \( 4 \times 4 \) system with determined list sizes [14]. The performance results are shown in Figure 2. It can be noted
that the algorithm with real signal model slightly outperforms the algorithm with complex signal model with the same list size. It should also be noted that the number of visited nodes is different with real and complex signal models. The numbers of visited nodes are listed in Table I.

We have studied the number of visited nodes by the IR-LSD and the SEE-LSD algorithms with both real and complex signal model. The number of visited nodes with the SEE-LSD in $4 \times 4$ system with 16-QAM and 64-QAM are shown in Figures 3 and 4, respectively. It can be seen that number of visited nodes by the real SEE-LSD algorithm is approximately 1.5-2 times the amount of nodes by the complex algorithm. The number of visited nodes with the IR-LSD algorithm in $4 \times 4$ system with 16-QAM and 64-QAM are shown in Figures 5 and 6, respectively. It can be noted that the signal model has no significant difference in the number of visited nodes with the IR-LSD.

The number of visited nodes with the real valued SEE-LSD is 1.5-2 times the complex valued SEE-LSD and there is almost no difference with the IR-LSD. The real K-best-LSD visits less nodes with higher order modulations compared to the complex algorithm. It should be noted that the PED calculation in (7) with complex signal model has approximately three times the complexity of the real signal model. Also the complex valued SEE used in both the SEE-LSD algorithm and the IR-LSD algorithm is remarkably more complex compared to real valued SEE. Thus, it can be concluded that the algorithms with real signal model are clearly less complex.

**B. Limited search**

The performance of the real and complex SEE-LSD algorithm with the number of nodes in the search limited in $4 \times 4$
system with 16-QAM and 64-QAM is shown in Figures 7 and 8, respectively. It can be seen that the complex SEE-LSD algorithm performs relatively fine with 16-QAM with the search limited to 8000 nodes, but the performance of real SEE-LSD is very poor. Both real and complex SEE-LSD algorithms perform poorly in $4 \times 4$ system with 64-QAM if the number of nodes is limited in the search.

The performance of the real and complex IR-LSD algorithm with the number of nodes in the search limited in $4 \times 4$ system with 16-QAM and 64-QAM is shown in Figures 9 and 10, respectively. It can be noted from the results that both real and complex IR-LSD algorithms perform well with limited search. The limit can be set with 16-QAM to 3000 with real algorithm and to 2000 with complex algorithm without performance degradation. Similarly with 64-QAM the limit with both real and complex algorithms can be set to 6000 nodes without performance degradation.

C. Discussion on the results

The comparison between real and complex valued LSD algorithms clearly showed that the real signal model is less complex and feasible for practical implementation. The real valued SEE-LSD visited more nodes, but the complex valued
algorithm requires significantly more complex operations in the expand loop. There was almost no difference with IR-LSD in the number of visited nodes with different signal models. The K-best-LSD has actually better performance with the same list size and visits less number of nodes with real signal model and with higher constellations compared to the complex signal model. The number of visited nodes is higher with complex valued K-best-LSD, because the algorithm calculates all $|\Omega|$ or $|\Omega_\ell|$ child nodes of the considered partial candidates at each layer. Thus, the number of calculated PEDs is larger with higher constellations with complex valued algorithm.

It was noted that the SEE-LSD algorithm performs poorly with a limited search. This is due to the search strategy that leaves some part of the tree totally unchecked if the search is limited, and, thus, the MAP approximation may be very poor. The IR-LSD algorithm metric first search, however, performs well with a limited search as the search continues always with the partial candidate with the lowest PED. The K-best-LSD algorithm has a limited number of nodes due to the search without any modifications.

The IR-LSD and the K-best-LSD algorithms are shown to be feasible for practical implementation. The K-best-LSD algorithm requires less operations in the node expand loop compared to the IR-LSD algorithm, and the algorithm is rather straightforward to implement [9]. Thus, the real valued K-best-LSD algorithm is the least complex choice to be used in implementation in a $4 \times 4$ system with 16-QAM. However, it can be noted that the breadth first search strategy is efficient with lower sized constellations, but with higher order constellations the number of visited nodes is remarkably larger compared to other more sophisticated search strategies. The IR-LSD algorithm requires far fewer checked nodes in the search in a $4 \times 4$ system with 64-QAM compared to the K-best-LSD algorithm. Thus, the IR-LSD algorithm is proposed to be used in a $4 \times 4$ system with 64-QAM.

V. SUMMARY

A LSD can be used to approximate the optimal MAP detection. We considered three LSD algorithms with different search strategies, namely the K-best-LSD, the SEE-LSD, and the IR-LSD algorithms, and studied and compared the implementation aspects related to the algorithms.

We compared the real and complex signal models in LSD algorithm implementation and studied its impact on the complexity and performance of three different LSD algorithms. It was shown that the real valued algorithms are less complex and feasible for practical implementation.

In hardware implementation, the number of visited nodes needs to be bounded in order to determine the complexity and the latency of the implementation. The performance of LSD algorithms was studied with limited number of nodes in the search. It was noted that the performance may suffer significantly with a limited search depending on the search strategy and that the SEE-LSD algorithm does not perform well with limited search. The IR-LSD and the K-best-LSD were found to be feasible for practical implementation.

ACKNOWLEDGEMENTS

This work was done in MITSE project which was supported by Elektrobit, Nokia, Texas Instruments and the Finnish Funding Agency for Technology and Innovation, Tekes. The first author would also like to thank Nokia foundation, and Jenny and Antti Wihuri foundation for supporting the research.

REFERENCES