On the Improvement of the Soft-Output Sphere Decoding

Shih-Lun Feng

Institute of Communications Engineering,
National Chiao-Tung University

August 31, 2010

Outline

- Introduction
- System Model
- Soft-Output Sphere Decoding Algorithm
- Methods for Complexity Reduction
- New Tree Traversal Strategy
- Simulation Results
- Conclusion and Future work

Outline

- Introduction
- System Model
- Soft-Output Sphere Decoding Algorithm
- Methods for Complexity Reduction
- New Tree Traversal Strategy
- Simulation Results
- Conclusion and Future work

- In recent years, the multiple-input multiple output (MIMO) system has attracted enormous interest because it can provide significant capacity improvement over traditional communication systems.
- To provide a better performance, the soft-output sphere decoding (SD) algorithm has been used as a support to an outer decoder in MIMO systems.
- At this background, this thesis presents a new soft-output detection algorithm to further improve the performance of existing methods.

Outline

- Introduction
- System Model
- Soft-Output Sphere Decoding Algorithm
- Methods for Complexity Reduction
- New Tree Traversal Strategy
- Simulation Results
- Conclusion and Future work

- Consider an MIMO communication system with N_T transmit antennas and N_R receive antennas, where $N_T \leq N_R$.
- At the transmitter, Q coded information bits are mapped onto a complex constellation $\mathcal{O}$, e.g., QPSK, 16-QAM, etc.

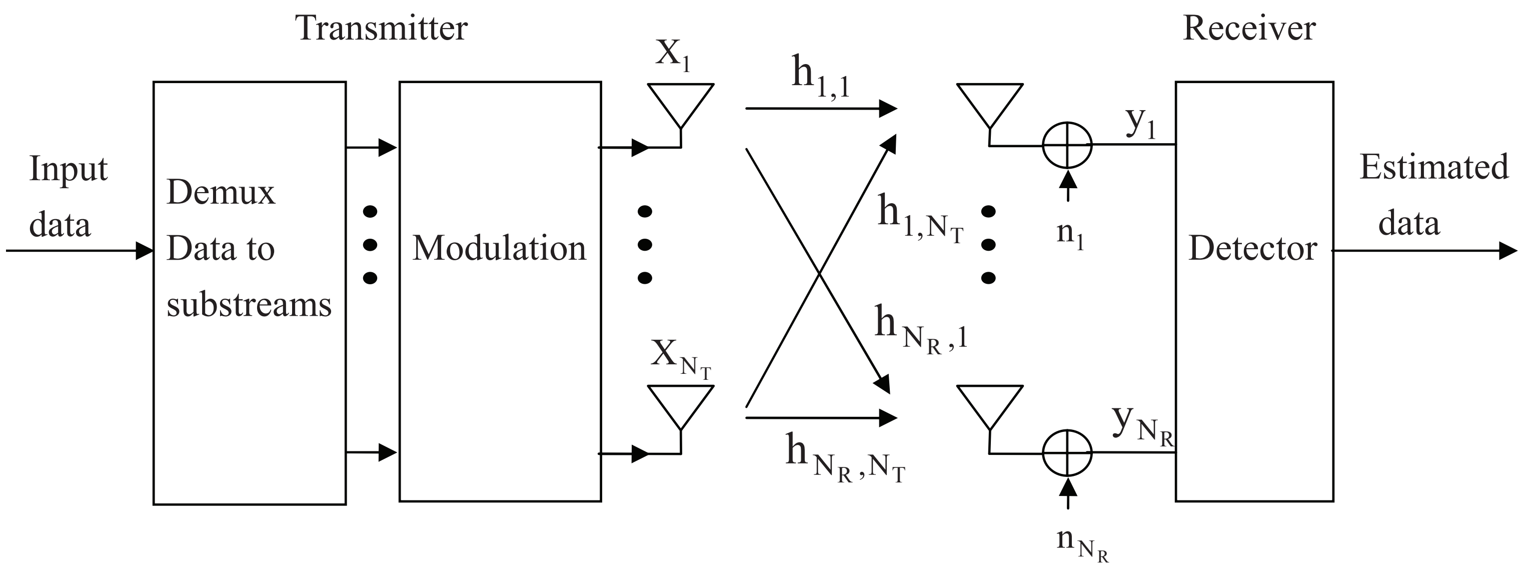


Figure 1: An MIMO channel model

- The received symbol vector thus can be written as:

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n} \tag{1}$$

where

$$\mathbf{x} = [x_1, \cdots, x_{N_T}]^T \in \mathcal{O}^{N_T} \tag{2}$$

is the transmitted symbol vector with $\mathbb{E}[|x_i|^2] = 1$,

$$\mathbf{y} = [y_1, \cdots, y_{N_R}]^T \in \mathbb{C}^{N_R} \tag{3}$$

is the received symbol vector,

$$\mathbf{n} = [n_1, \cdots, n_{N_R}]^T \in \mathbb{C}^{N_R} \tag{4}$$

is the independent zero-mean Gaussian-distributed complex noise vector with common variance N_0 per entry, and $\mathbb{C}$ denotes the domain of complex numbers.

- The channel matrix is

$$\mathbf{H} = \begin{bmatrix} h_{1,1} & \cdots & h_{1,N_T} \\ \vdots & \ddots & \vdots \\ h_{N_R,1} & \cdots & h_{N_R,N_T} \end{bmatrix} \in \mathbb{C}^{N_R \times N_T} \tag{5}$$

- We assume that the elements of channel matrix $\mathbf{H}$ are complex Gaussian variables with zero mean and unit variance and can be perfectly estimated by the receiver.

Outline

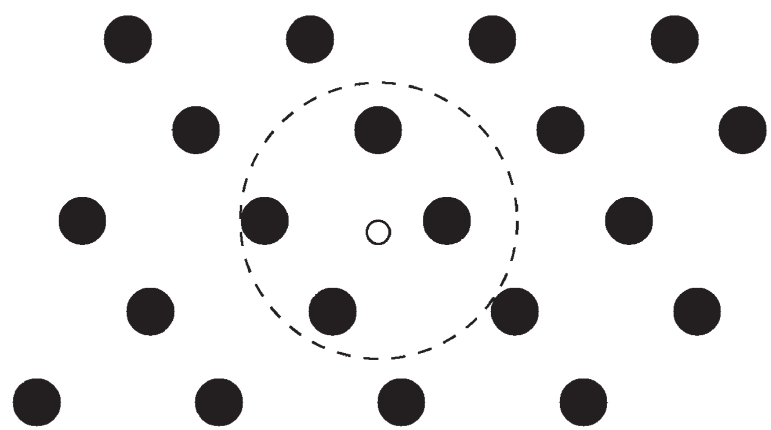
- Introduction
- System Model
- Soft-Output Sphere Decoding Algorithm
- Methods for Complexity Reduction
- New Tree Traversal Strategy
- Simulation Results
- Conclusion and Future work

- The ML rule:

$$\hat{\mathbf{x}}_{\mathsf{ML}} = \arg \min_{\mathbf{x} \in \mathcal{O}^{N_T}} \| \mathbf{y} - \mathbf{H}\mathbf{x} \|^2 \tag{6}$$

- The sphere decoding (SD) algorithm can significantly reduce the computational complexity while maintaining the ML performance

$$\hat{\mathbf{x}}_{\mathsf{ML}} = \arg \min_{\mathbf{x} \in \{ \mathbf{x} : r \geq \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 \}} \| \mathbf{y} - \mathbf{H}\mathbf{x} \|^2$$



- As contrary to the hard-output SD algorithm that targets to find $\hat{\mathbf{x}}_{\mathrm{ML}}$, the soft-output SD algorithm provides soft-valued log-likelihood ratios (LLR) (in order to cooperate with an outer coder).
- For clarity, we present the soft-output SD algorithm from two angles:
  - Derivation of the Max-Log LLRs
  - Computation of Max-Log LLRs via a Tree Search

Derivation of the Max-Log LLRs

- Denote by $x_{j,b}$ the b-th bit in the constellation point corresponding to the j-th component of vector $\mathbf{x}$, where $1 \leq j \leq N_T$ and $1 \leq b \leq Q$.

- The Max-Log LLRs for bit $x_{j,b}$ is given by

$$L(x_{j,b}) = \min_{\mathbf{x} \in \mathcal{X}_{j,b}^{(0)}} \parallel \mathbf{y} - \mathbf{H}\mathbf{x} \parallel^2 - \min_{\mathbf{x} \in \mathcal{X}_{j,b}^{(1)}} \parallel \mathbf{y} - \mathbf{H}\mathbf{x} \parallel^2 \qquad (7)$$

where $\mathcal{X}_{j,b}^{(0)}$ and $\mathcal{X}_{j,b}^{(1)}$ are respectively the sets of symbol vectors that have the b-th bit in the j-th entry equal to 0 and 1.

- It can be easily seen that one of the two minima in (7) is

$$\lambda^{\mathsf{ML}} = \| \mathbf{y} - \mathbf{H}\mathbf{x}^{\mathsf{ML}} \|^2 \tag{8}$$

which is the metric associated with the ML solution $\mathbf{x}^{\mathsf{ML}}$.

- Denote the b-th bit in the j-th entry of $\mathbf{x}^{\mathsf{ML}}$ as $x_{j,b}^{\mathsf{ML}} \in \{0, 1\}$, and let its binary complement be denoted by $\overline{x_{j,b}^{\mathsf{ML}}}$.

- Then, the minimum, other than λ^{ML} in (7), can be written as

$$\lambda_{j,b}^{\overline{ML}} = \min_{\mathbf{x} \in \mathcal{X}_{j,b}^{\left(\overline{x_{j,b}^{\mathsf{ML}}}\right)}} \| \mathbf{y} - \mathbf{H}\mathbf{x} \|^2 \tag{9}$$

- By combining (8) and (9), the Max-Log LLRs can be equivalently expressed as

$$L(x_{j,b}) = \begin{cases} \lambda^{\mathsf{ML}} - \lambda_{j,b}^{\overline{ML}}, & \text{if } \; x_{j,b}^{\mathsf{ML}} = 0; \\ \lambda_{j,b}^{\overline{ML}} - \lambda^{\mathsf{ML}}, & \text{if } \; x_{j,b}^{\mathsf{ML}} = 1. \end{cases} \tag{10}$$

Computation of Max-Log LLRs via a Tree Search

- We next transform the computations of (8) and (9) into a tree search problem.

- The channel matrix $\mathbf{H}$ is QR-decomposed as

$$\mathbf{H} = \mathbf{Q} \begin{bmatrix} \mathbf{R} \\ \mathbf{0}_{(N_R - N_T) \times N_T} \end{bmatrix} \tag{11}$$

  where the $N_R \times N_R$ matrix $\mathbf{Q}$ is unitary, and $\mathbf{R}$ is an $N_T \times N_T$ upper triangular matrix with diagonals being real-valued.

- Multiplying (1) by $\mathbf{Q}^H$ leads to a modified input-output relation as

$$\tilde{\mathbf{y}} = \mathbf{R}\mathbf{x} + \tilde{\mathbf{n}} \tag{12}$$

  where $\tilde{\mathbf{y}}$ and $\tilde{\mathbf{n}}$ respectively contain the first N_T components of $\mathbf{Q}^H\mathbf{y}$ and $\mathbf{Q}^H\mathbf{n}$.

- In matrix form, (12) can be written as

$$\begin{bmatrix} \tilde{y}_1 \\ \vdots \\ \tilde{y}_{N_T} \end{bmatrix} = \begin{bmatrix} r_{1,1} & r_{1,2} & \cdots & r_{1,N_T} \\ 0 & r_{2,2} & \cdots & r_{2,N_T} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & r_{N_T,N_T} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_{N_T} \end{bmatrix} + \begin{bmatrix} \tilde{n}_1 \\ \vdots \\ \tilde{n}_{N_T} \end{bmatrix} \tag{13}$$

- As $\mathbf{Q}$ is unitary, $\tilde{\mathbf{n}}$ remains independent Gaussian distributed with zero mean and common variance N_0.

- Hence, (6) can be equivalently rewritten as

$$\hat{\mathbf{x}}_{\mathsf{ML}} = \arg \min_{\mathbf{x} \in \mathcal{O}^{N_T}} \parallel \tilde{\mathbf{y}} - \mathbf{R}\mathbf{x} \parallel^2 = \arg \min_{\mathbf{x} \in \mathcal{O}^{N_T}} \sum_{i=1}^{N_T} \left| \tilde{y}_i - \sum_{j=i}^{N_T} r_{i,j} x_j \right|^2 \tag{14}$$

- After the preprocessing stage, the equivalent characterizations of λ^{ML} and $\lambda_{j,b}^{\overline{ML}}$ can be respectively obtained as

$$\lambda^{\mathsf{ML}} = \min_{\mathbf{x} \in \mathcal{O}^{N_T}} \parallel \tilde{\mathbf{y}} - \mathbf{R}\mathbf{x} \parallel^2 \tag{15}$$

and

$$\lambda_{j,b}^{\overline{ML}} = \min_{\mathbf{x} \in \mathcal{X}_{j,b}^{\left(\overline{x_{j,b}^{\mathsf{ML}}}\right)}} \parallel \tilde{\mathbf{y}} - \mathbf{R}\mathbf{x} \parallel^2 . \tag{16}$$

- Define the partial symbol vector (PSV) as $\mathbf{x}^{(k)} = [x_k, \ x_{k+1}, \ \ldots, \ x_{N_T}]^T$, and form a tree with nodes marked by the PSVs.

- The nodes at level k are marked by $\mathbf{x}^{(k)}$ and the dummy root note is conveniently marked as $\mathbf{x}_0$.

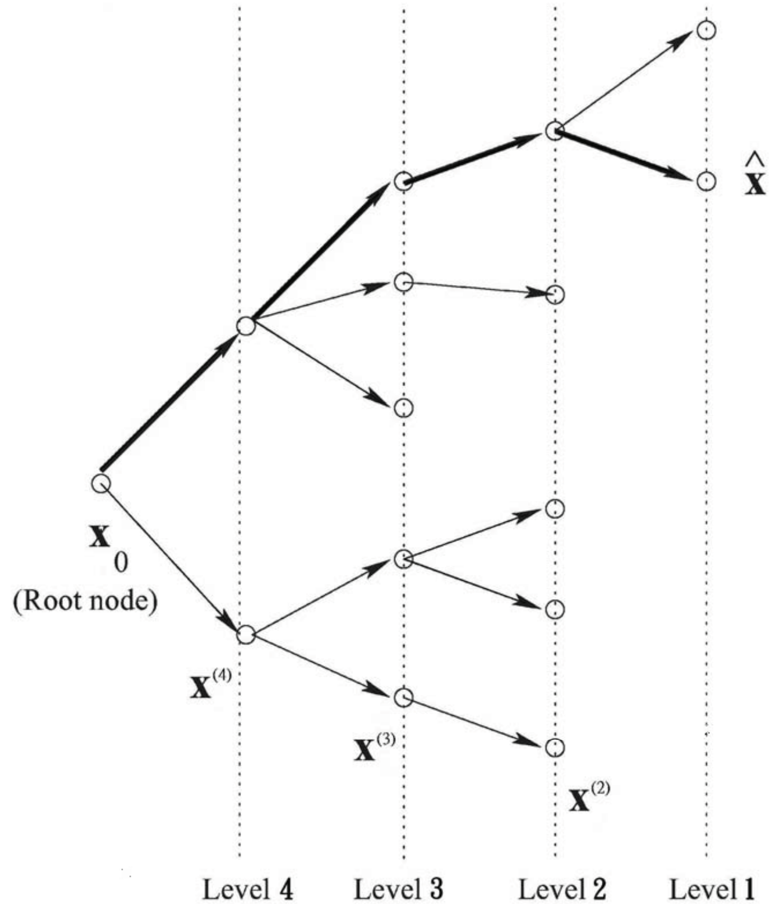


Figure 2: The tree to be searched by the SD decoder. In this example, $N_T = 4$.

- We define recursively the partial Euclidean distance (PED) $d(\mathbf{x}^{(k)})$ to be equal to $d_k = d_k(\mathbf{x}^{(k)})$, where

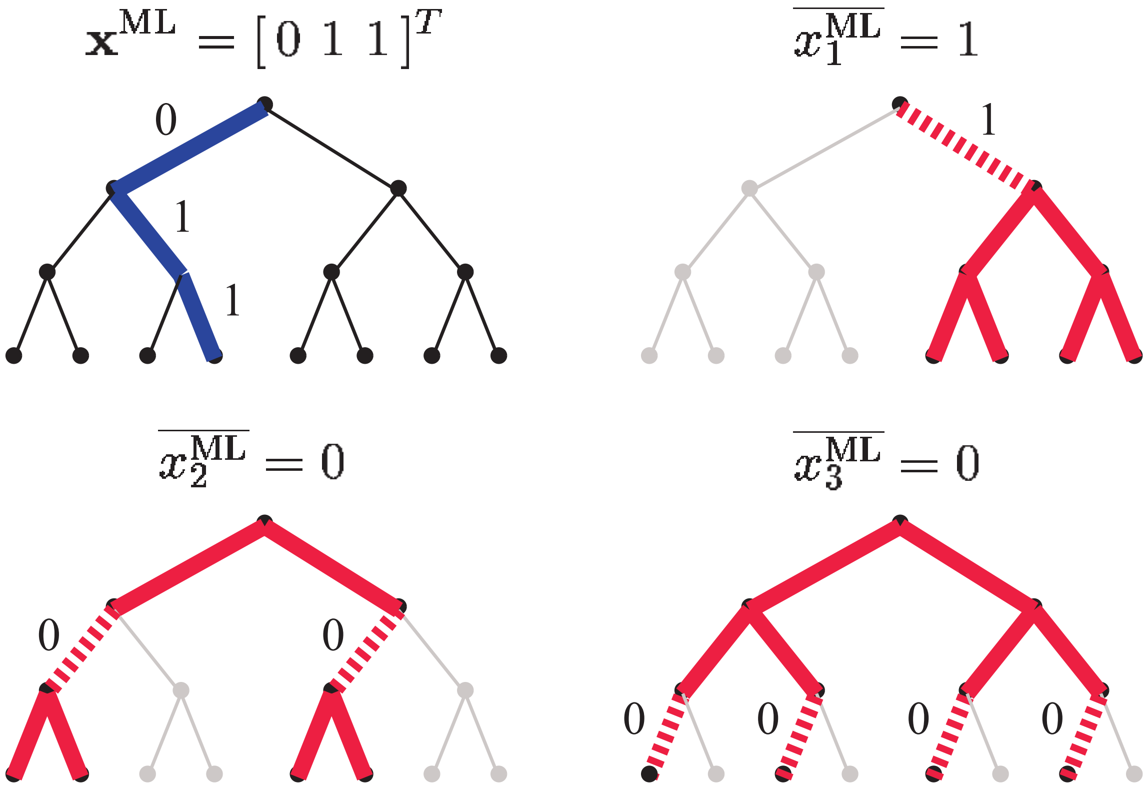
$$d_i = d_{i+1} + |e_i|^2 \quad \text{for } i = N_T,\ N_T - 1 \ldots, k, \tag{17}$$

and d_{N_T+1} is initialized as 0, and the distance increment (DI) $|e_i|^2$ is given by

$$|e_i|^2 = \left| \tilde{y}_i - \sum_{j=i}^{N_T} r_{i,j} x_j \right|^2 \tag{18}$$

- It is then clear from the formulas that the PEDs only depend on the PSVs, and they can be regarded as the branch metrics during the tree search.
- We will introduce two tree traversal strategies for the generation of the LLRs:
  - Repeated Tree Search (RTS) strategy
  - Single Tree Search (STS) strategy

Example of BPSK-constellation in the RTS strategy



Single Tree Search (STS)

- The STS is more efficient than the RTS because it ensures that every node in the tree is visited at most once.
- It searches for the ML solution and all counter-hypotheses concurrently.
- When a leaf is reached, two situations will be considered:
  - If a new ML hypothesis $\mathbf{x}$ is found, i.e., $d(\mathbf{x}) < \lambda^{\mathsf{ML}}$, all $\lambda_{j,b}^{\overline{ML}}$'s, for which their corresponding $x_{j,b} = \overline{x_{j,b}^{\mathsf{ML}}}$, are set to the current λ^{ML}, followed by two new updates:

$$\lambda^{\mathsf{ML}} \leftarrow d(\mathbf{x}) \quad \text{and} \quad \mathbf{x}^{\mathsf{ML}} \leftarrow \mathbf{x}.$$

  - If $d(\mathbf{x}) \geq \lambda^{\mathsf{ML}}$, only the counter-hypotheses have to be checked. In other words, for all j and b such that $x_{j,b} = \overline{x_{j,b}^{\mathsf{ML}}}$ and $d(\mathbf{x}) < \lambda_{j,b}^{\overline{ML}}$, the decoder updates $\lambda_{j,b}^{\overline{ML}} \leftarrow d(\mathbf{x})$.

Outline

- Introduction
- System Model
- Soft-Output Sphere Decoding Algorithm
- Methods for Complexity Reduction
- New Tree Traversal Strategy
- Simulation Results
- Conclusion and Future work

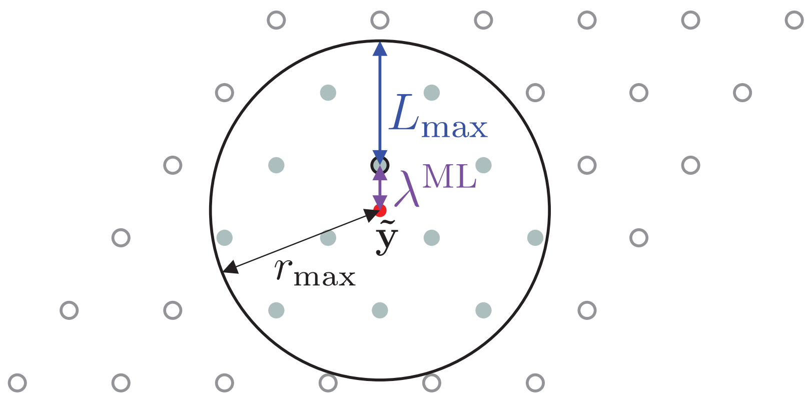
Methods for Complexity Reduction for the STS

- Computing the exact LLR values will require a large amount of computational complexity.
- Three common methods to reduce the computational complexity are:
  - LLR Clipping
  - Sorting and Regularization
  - Run-Time Constraint

LLR Clipping

- $$|L(x_{j,b})| \leq L_{\max}, \ \forall \ 1 \leq j \leq N_T \text{ and } 1 \leq b \leq Q, \tag{19}$$
  where $L_{\max}$ is a chosen maximum LLR value.

- Conceptually, the LLR clipping is to restrict the search domain for lattice points inside the square radius $r_{\max} = \lambda^{\mathsf{ML}} + L_{\max}$.



- It can be anticipated that if $L_{\max} = \infty$, the LLR clipping returns the exact max-log LLRs.

Sorting and Regularization

- *Sorting:*
  - Perform QR-decomposition on $\mathbf{HP}$ instead of $\mathbf{H}$, where $\mathbf{P}$ is an $N_T \times N_T$ column permutation matrix.
  - The idea behind sorting is to let the diagonal entries of the upper triangular matrix $\mathbf{R}$ being sorted in ascending order.
  - Thus, the stronger streams in term of effective signal-to-noise (SNR) ratio are closer to the root.
  - For this reason, it is referred to as sorted QR-decomposition (SQRD).

- *Regularization:*
  - If $\mathbf{H}$ is in a "poor" condition, sorting may still give high search complexity.
  - This problem can be resolved by operating on a regularized channel matrix

$$\begin{bmatrix} \mathbf{H} \\ \alpha \mathbf{I}_{N_T} \end{bmatrix} \mathbf{P} = \mathbf{Q}\mathbf{R} = \begin{bmatrix} \mathbf{Q}_1 \\ \mathbf{Q}_2 \end{bmatrix} \mathbf{R}, \tag{20}$$

    where α is a suitably chosen regularization parameter, $\mathbf{Q}$ is an $(N_R + N_T) \times N_T$ unitary matrix , $\mathbf{R}$ is an $N_T \times N_T$ upper triangular matrix, $\mathbf{I}_{N_T}$ denotes the $N_T \times N_T$ identity matrix, $\mathbf{Q}_1$ is of dimension $N_R \times N_T$ and $\mathbf{Q}_2$ is of dimension $N_T \times N_T$.

– The system model can then be equivalently transformed to

$$\begin{aligned}\tilde{\mathbf{y}} &= \mathbf{Q}_1^H \mathbf{y} \\ &= \mathbf{Q}_1^H \mathbf{H}\mathbf{x} + \mathbf{Q}_1^H \mathbf{n} \\ &= \mathbf{Q}_1^H \mathbf{H}\mathbf{x} + \alpha \mathbf{Q}_2^H \mathbf{x} - \alpha \mathbf{Q}_2^H \mathbf{x} + \mathbf{Q}_1^H \mathbf{n} \\ &= \left(\mathbf{Q}_1^H \mathbf{H} + \alpha \mathbf{Q}_2^H\right) \mathbf{x} - \alpha \mathbf{Q}_2^H \mathbf{x} + \mathbf{Q}_1^H \mathbf{n} \\ &= \mathbf{R}\tilde{\mathbf{x}} + \tilde{\mathbf{n}},\end{aligned}$$

where $\tilde{\mathbf{x}} = \mathbf{P}^{-1}\mathbf{x}$ and $\tilde{\mathbf{n}} = -\alpha \mathbf{Q}_2^H \mathbf{x} + \mathbf{Q}_1^H \mathbf{n}$.

– The Max-Log LLRs in (7) can accordingly be approximated by

$$L(x_{j,b}) \approx \min_{\tilde{\mathbf{x}} \in \mathcal{X}_{j,b}^{(0)}} \| \tilde{\mathbf{y}} - \mathbf{R}\tilde{\mathbf{x}} \|^2 - \min_{\tilde{\mathbf{x}} \in \mathcal{X}_{j,b}^{(1)}} \| \tilde{\mathbf{y}} - \mathbf{R}\tilde{\mathbf{x}} \|^2 \tag{21}$$

by pretending $\tilde{\mathbf{n}}$ to be i.i.d. complex Gaussian distributed.

– Note that the effective noise-plus-(self)-interference (NPI) vector

$$\tilde{\mathbf{n}} = -\alpha \mathbf{Q}_2^H \mathbf{x} + \mathbf{Q}_1^H \mathbf{n} \tag{22}$$

is neither i.i.d. nor complex Gaussian due to the self-interference term $-\alpha \mathbf{Q}_2^H \mathbf{x}$.

– In order to get a good approximation, we compute the covariance matrix of $\tilde{\mathbf{n}}$ as

$$\mathbf{K} = \mathbb{E}[\tilde{\mathbf{n}}\tilde{\mathbf{n}}^H] = \left(\mathbf{R}\mathbf{R}^H\right)^{-1} |\alpha|^2 \left(|\alpha|^2 - N_0\right) + N_0 \mathbf{I}_{N_T} \tag{23}$$

where we assume $\mathbb{E}[\mathbf{x}\mathbf{x}^H] = \mathbf{I}_{N_T}$, then setting $\alpha = \pm\sqrt{N_0}$ corresponds to an MMSE regularization, yielding $\mathbf{K} = N_0 \mathbf{I}_{N_T}$.

– This gives a theoretically conclusive aspect of the so-called MMSE-SQRD with the requirement that N_0 needs to be estimated.

Run-Time Constraint

- We suppose the maximum number of tree nodes allowed to be visited for the decoding of an N-symbol-vector block to be denoted by $N \cdot D_{\text{avg}}$.
- Then, the decoding of the k-th symbol vector can visit at most

$$D_{\max}(k) = N \cdot D_{\text{avg}} - \sum_{i=1}^{k-1} D(i) - (N-k)N_T \tag{24}$$

nodes for $k = 1, 2, \ldots, N$, where $D(i)$ denotes the number of nodes actually visited during the decoding of the i-th symbol vector.

- In concept, (24) said that when a symbol vector is decoded, it is allowed to use all the remaining budget but has to maintain a safety margin $(N-k)N_T$.
- This margin allows the remaining symbols to achieve at least the hard-output SIC performance.

Outline

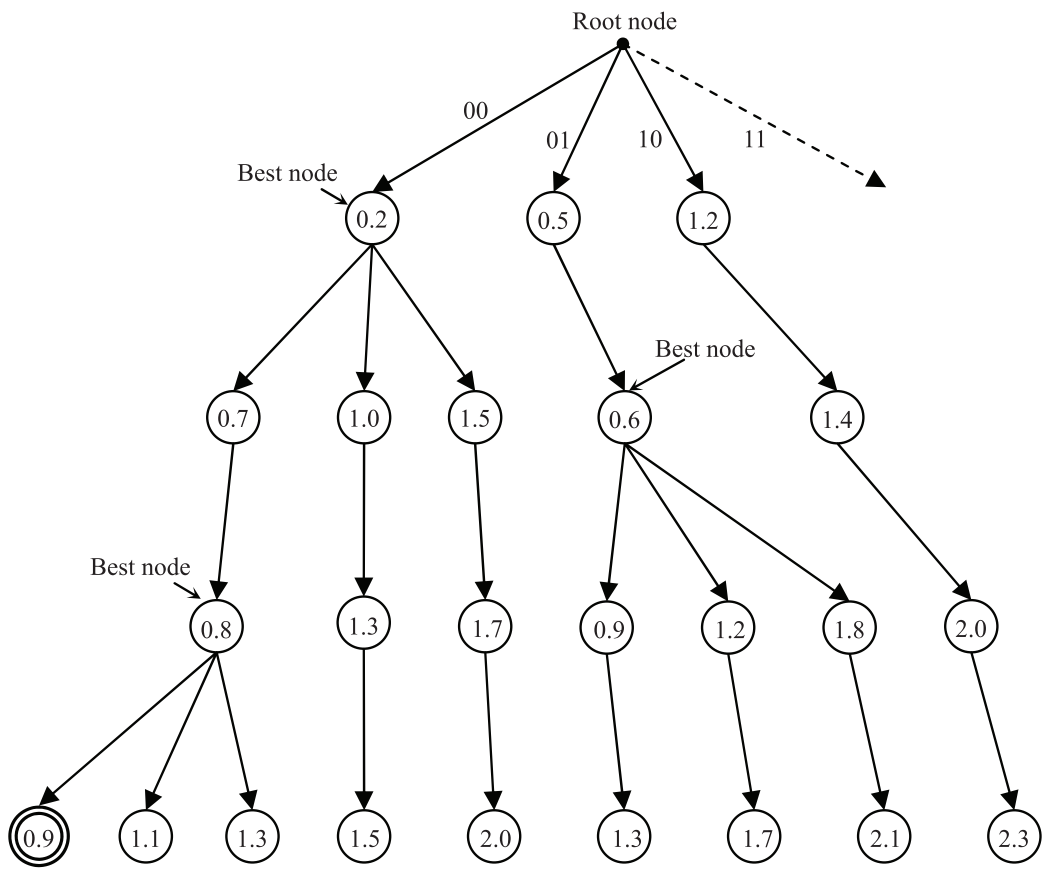
- Introduction
- System Model
- Soft-Output Sphere Decoding Algorithm
- Methods for Complexity Reduction
- New Tree Traversal Strategy
- Simulation Results
- Conclusion and Future work

- We want to design a new tree traversal strategy that yields almost the same computational complexity but has better performance.
- The tree traversal strategy that we newly proposed is basically MMSE-SQRD-based and is a refinement of the Schnorr-Euchner sphere decoder (SESD).
- We will focus on the QPSK and 16-QAM constellation using Gray mapping.
- Two methods respectively for QPSK and 16-QAM will be proposed.

New Tree Search Strategy for QPSK

- Observation: Only the paths that generate the LLR values are necessary to be extended.
- A QPSK symbol only carries two information bits, so once the best path at the current level is decided, only its top three successor paths need to be extended.
- For the paths other than the best path at the current level, only the best successor path needs to be extended.

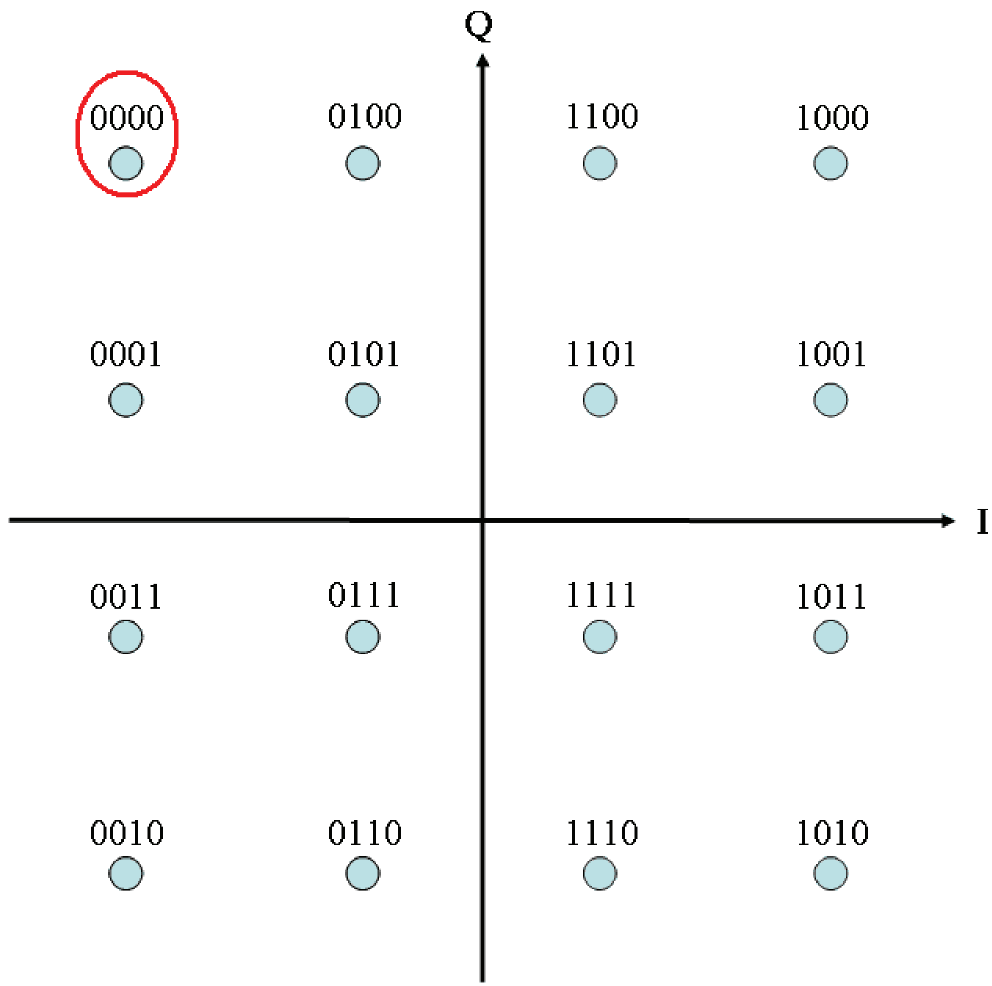
An example of our proposed method



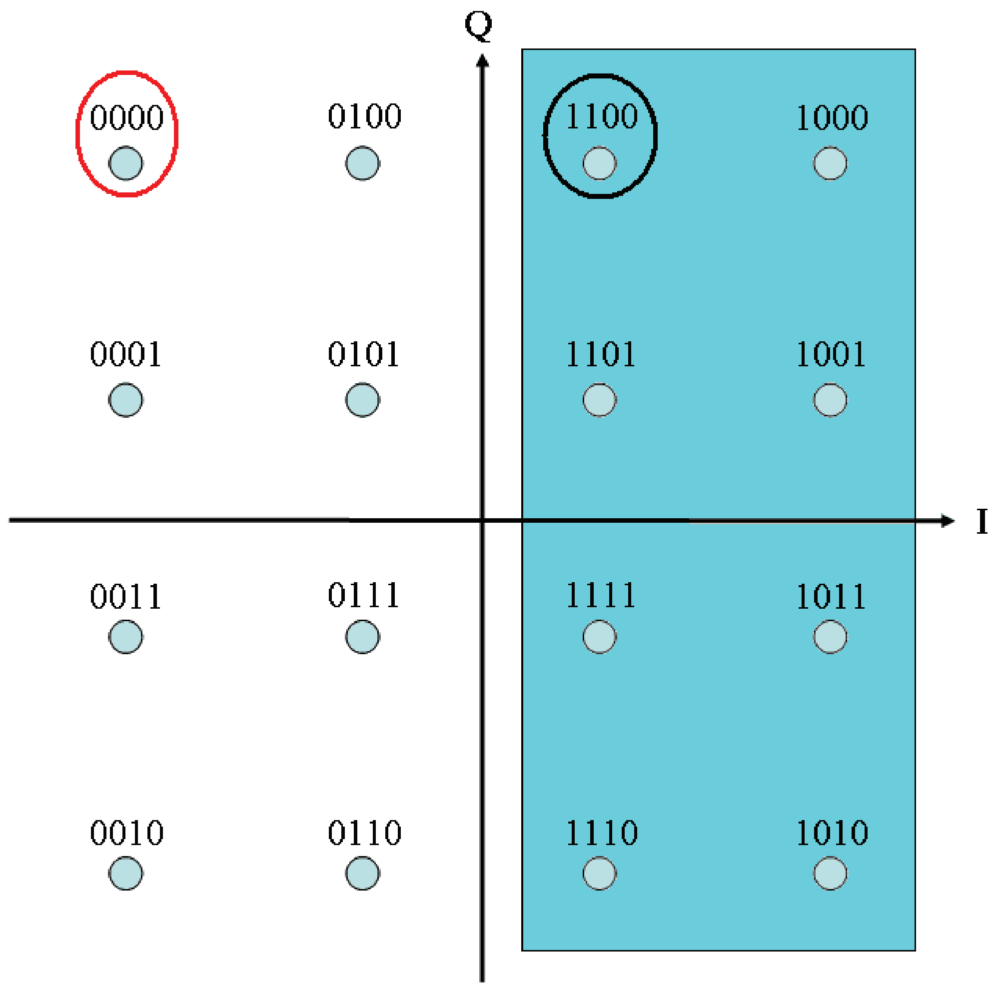
New Tree Search Strategy for 16-QAM using Gray Code Mapping

- Since the previous approach provides no visible improvement for 16-QAM, a new method has been proposed.
- We first find the ML solution by using the hard-output SD algorithm.
- Then, we follow the ML path to decide which paths are relevant to the LLR computations and hence need to be extended.
- Only four successor paths corresponding to the counter-hypotheses (as contrary to the ML path) need to be extended in addition to the expansion of the ML path itself.
- Among the four paths that include all the counter-hypotheses corresponding to the ML symbol, the best path will be further extended.
- Further extension along this path only includes the best path.

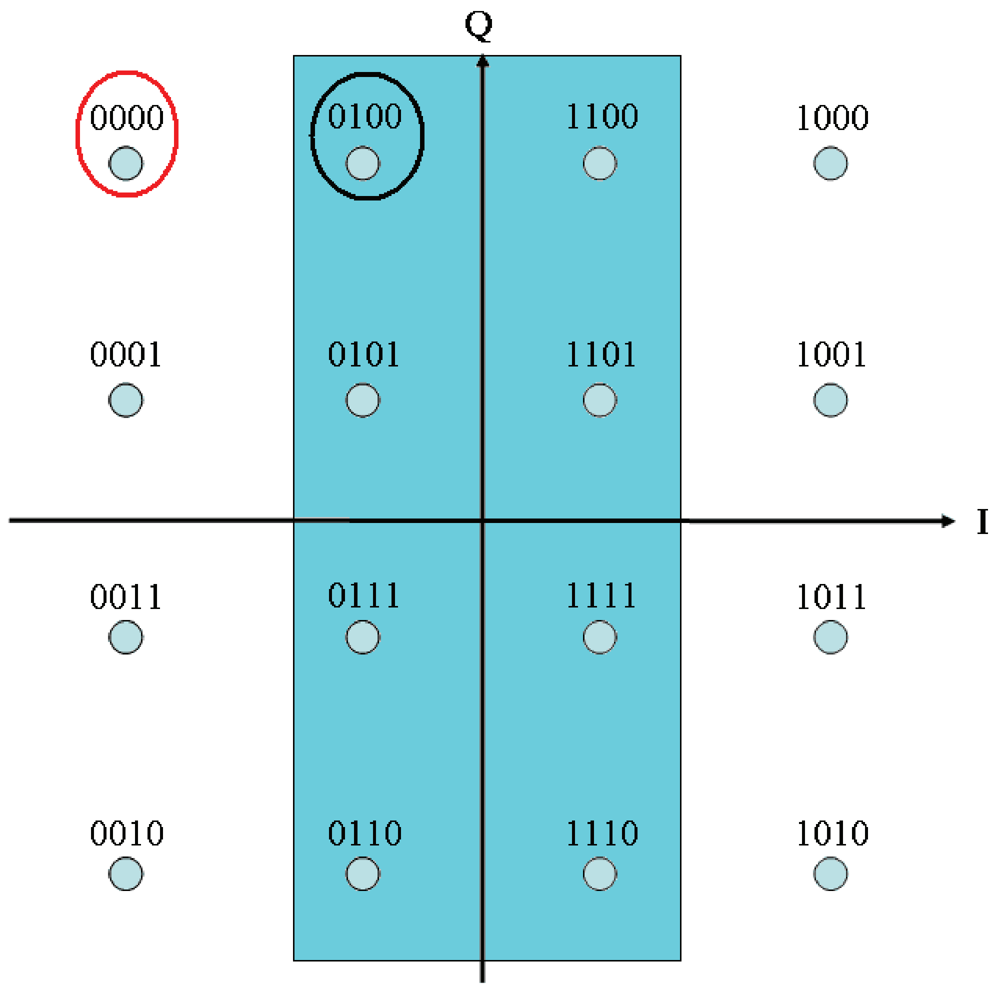
16-QAM constellation



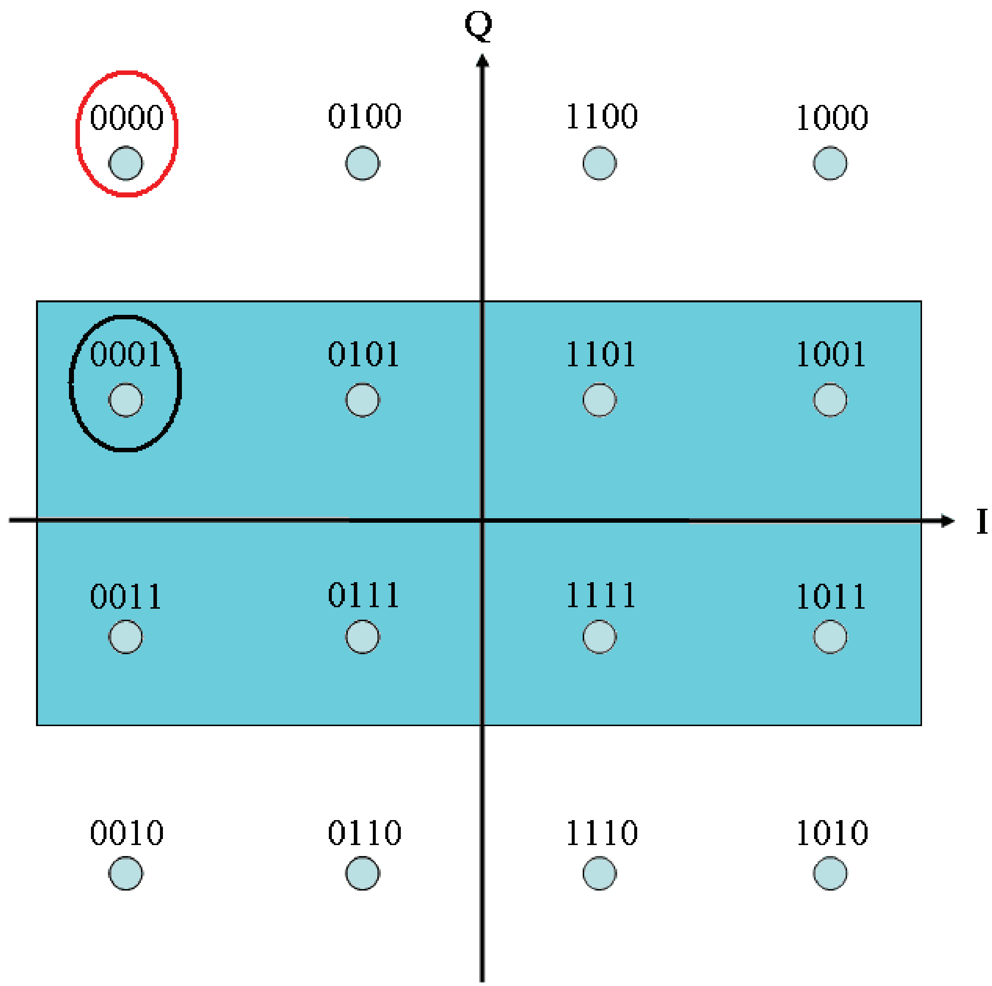
16-QAM constellation



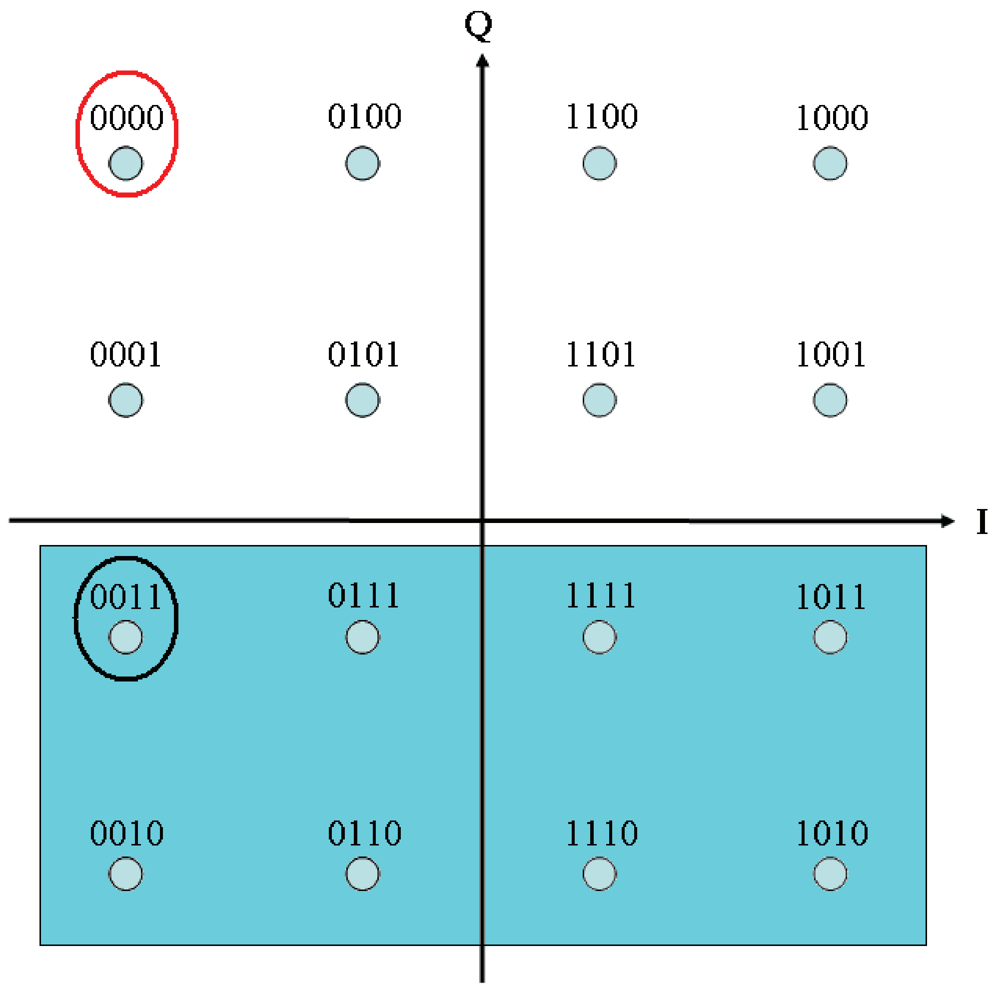
16-QAM constellation



16-QAM constellation



16-QAM constellation



16-QAM constellation

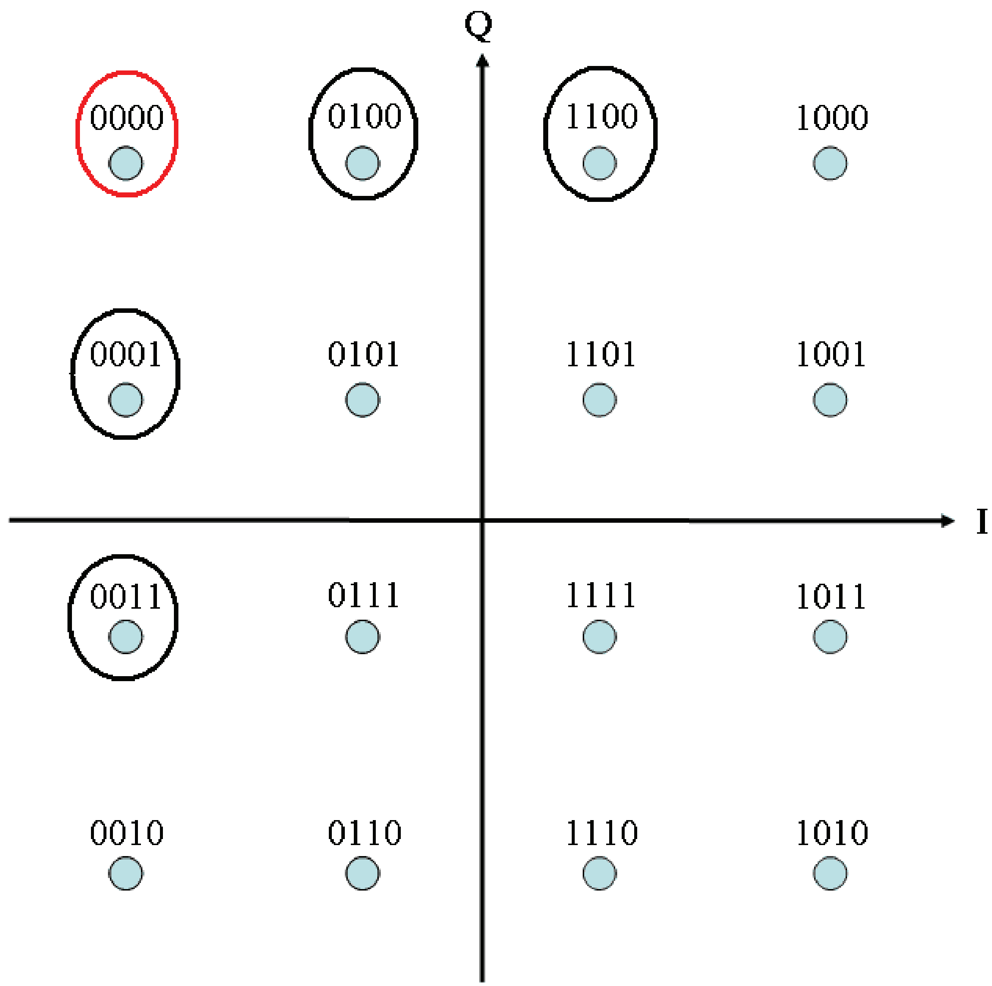
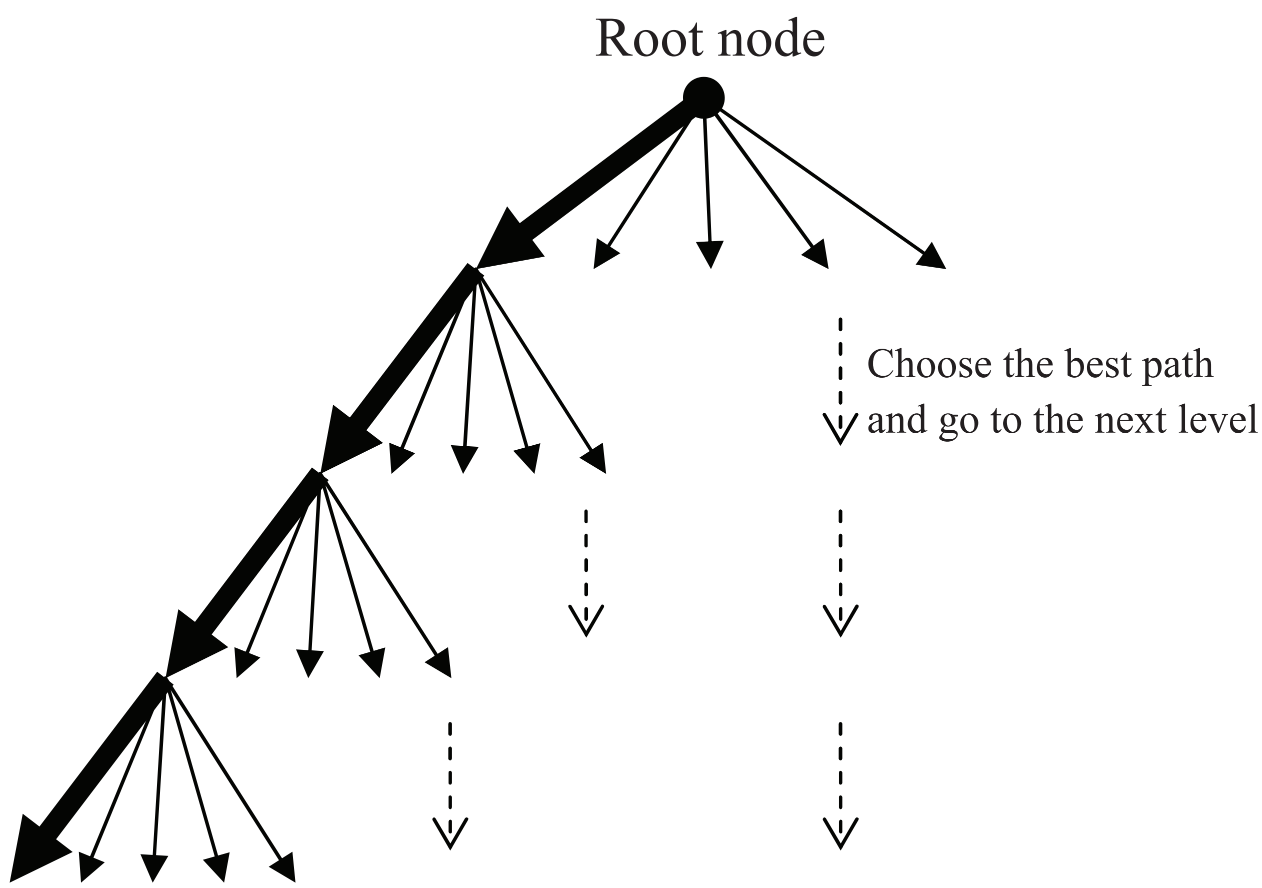


Illustration of the proposed algorithm for the 16-QAM constellation



Outline

- Introduction
- System Model
- Soft-Output Sphere Decoding Algorithm
- Methods for Complexity Reduction
- New Tree Traversal Strategy
- Simulation Results
- Conclusion and Future work

- Flat fading MIMO channels with $N_T = N_R = 4$
- The outer code is a turbo code with rate $R = 1/2$.
- The codeword lengths for the QPSK and 16-QAM constellations are 1000 bits and 2000 bits, respectively, such that 500 symbols are transmitted for both cases for a block of $N = 125$ symbol vectors.

Effect of LLR Clipping

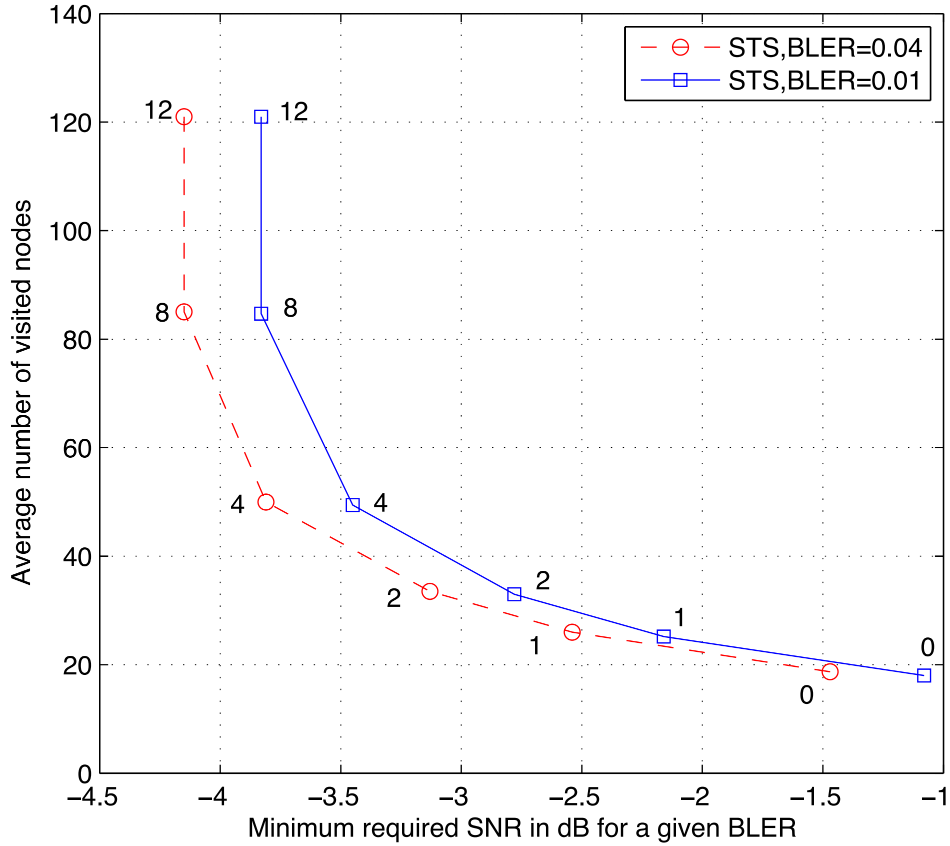


Figure 3: The LLR clipping under the single tree search (STS) strategy for the QPSK modulation. The numbers marked next to the nodes correspond to the used $L_{\max}$ values.

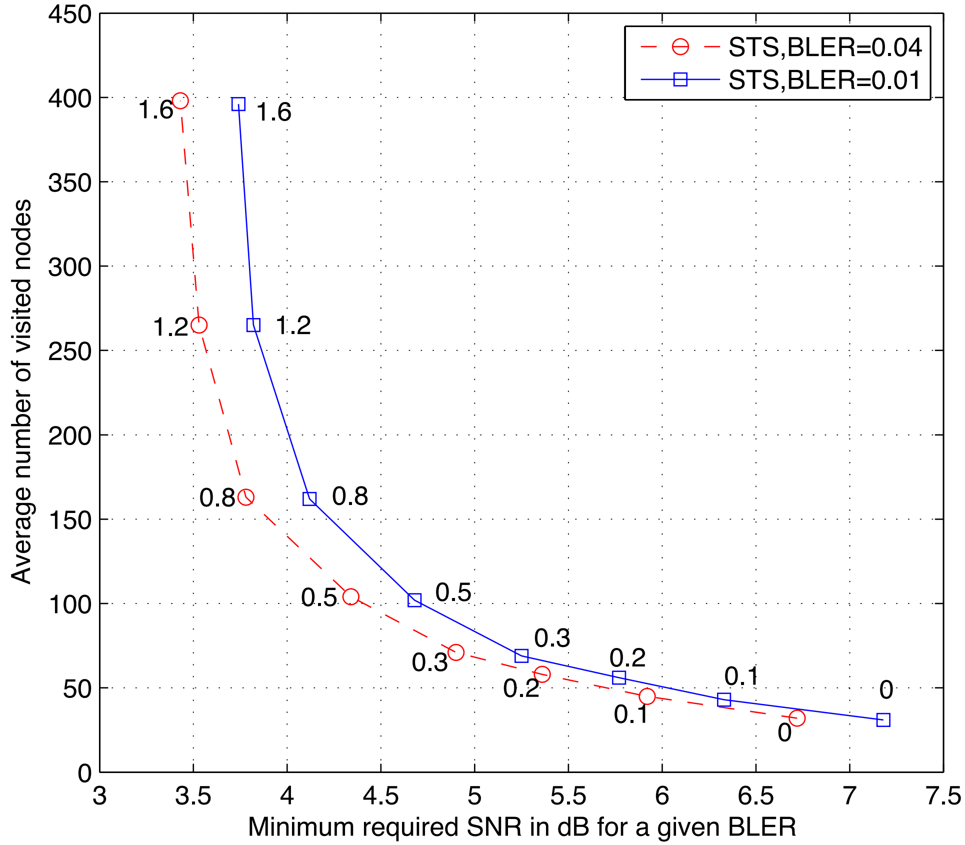


Figure 4: The LLR clipping under the single tree search (STS) strategy for the 16-QAM modulation. The numbers marked next to the nodes correspond to the used $L_{\max}$ values.

Effect of Sorting and Regularization

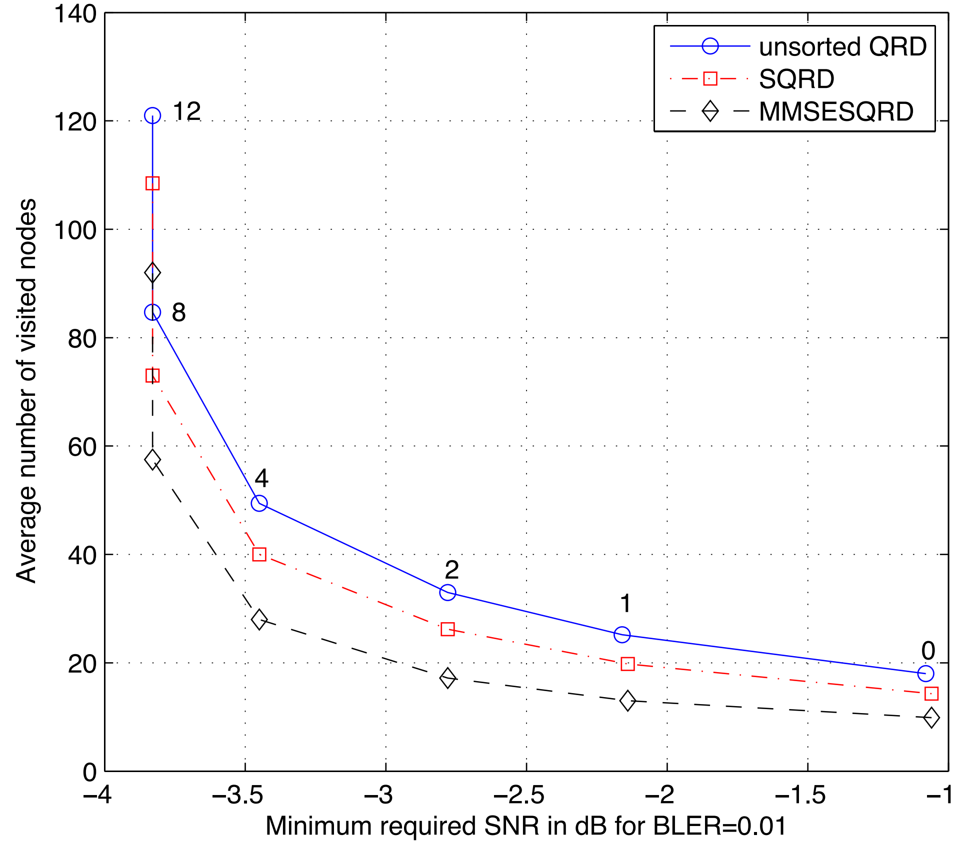


Figure 5: Comparisons of unsorted QRD, SQRD and MMSE-SQRD preprocessing applied to the STS for the QPSK modulation.

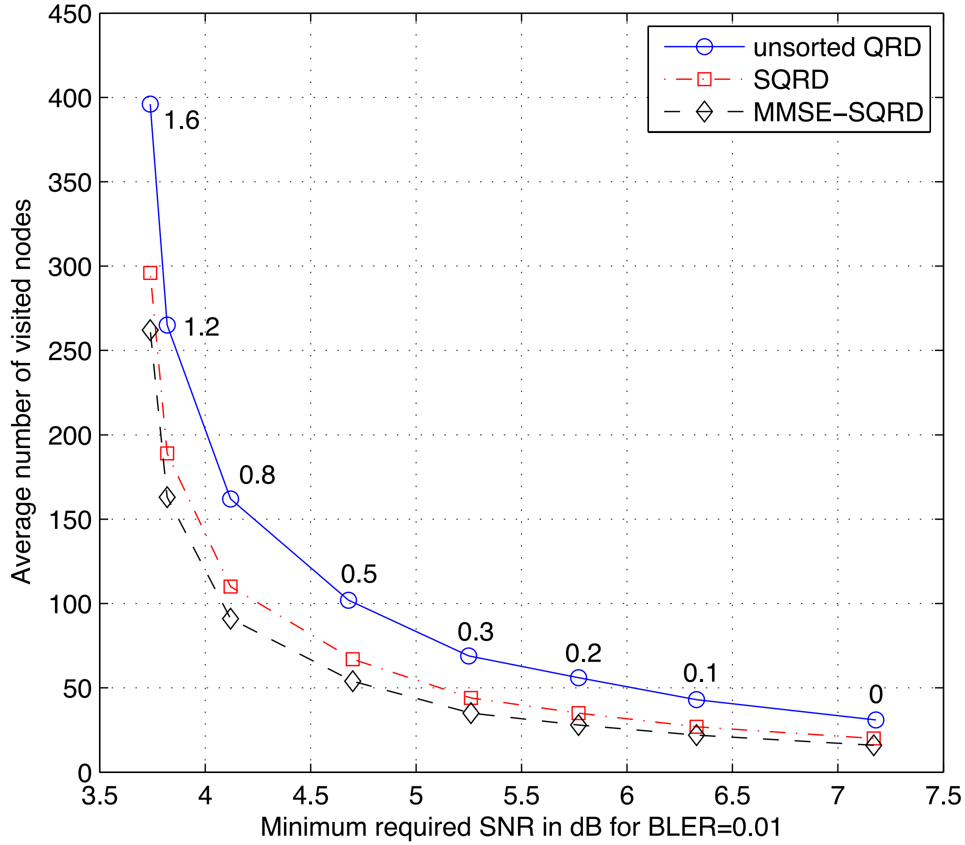


Figure 6: Comparisons of unsorted QRD, SQRD and MMSE-SQRD preprocessing applied to the STS for the 16-QAM modulation.

Effect of Run-time Constraints

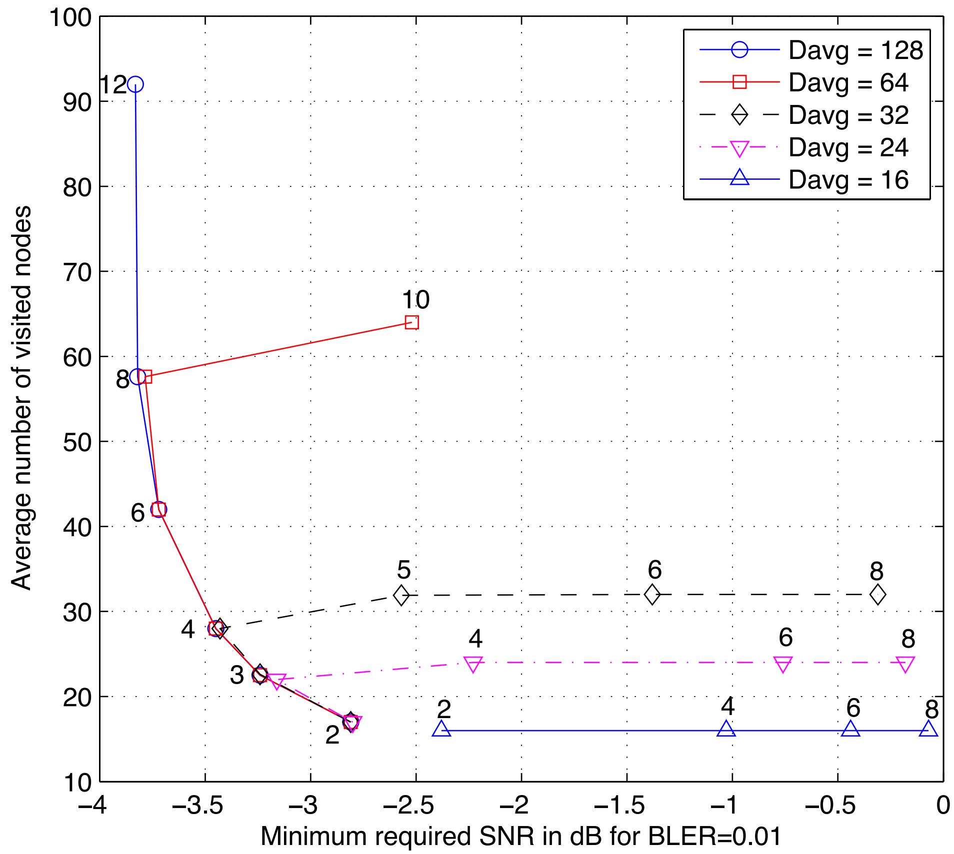


Figure 7: Impact of the run-time constraint on the STS SESD (with MMSE-SQRD preprocessing) for the QPSK modulation.

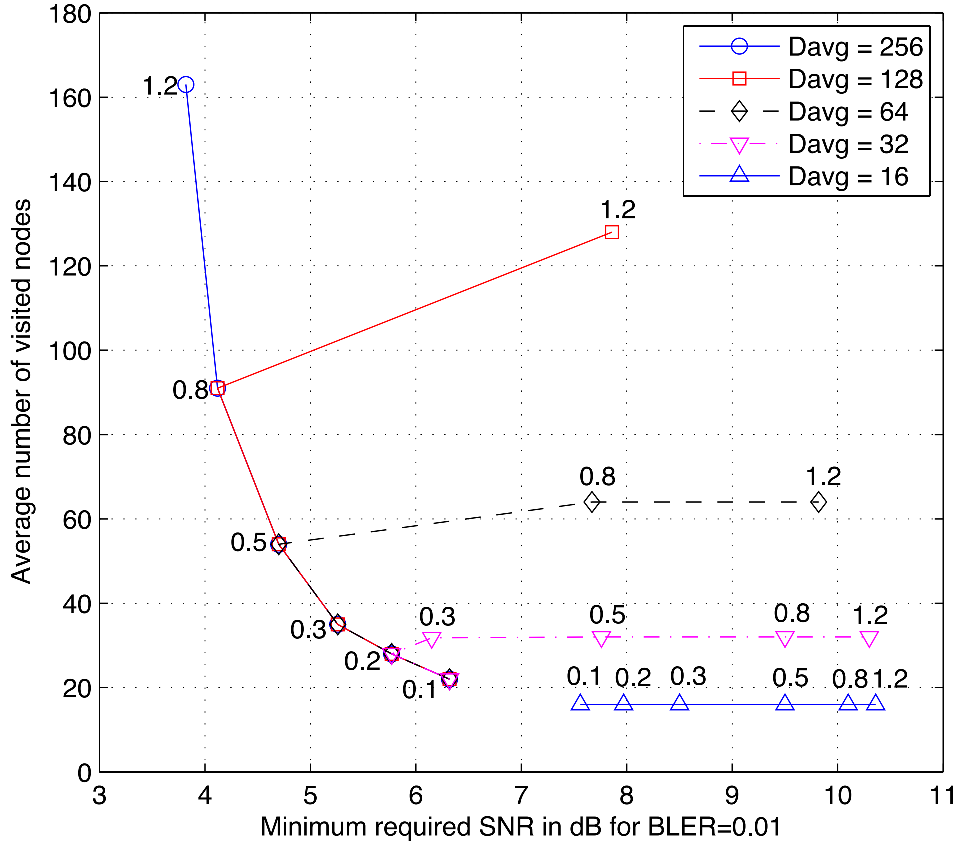


Figure 8: Impact of the run-time constraint on the STS SESD (with MMSE-SQRD preprocessing) for the 16-QAM modulation.

Effect of the New Tree Traversal Strategy

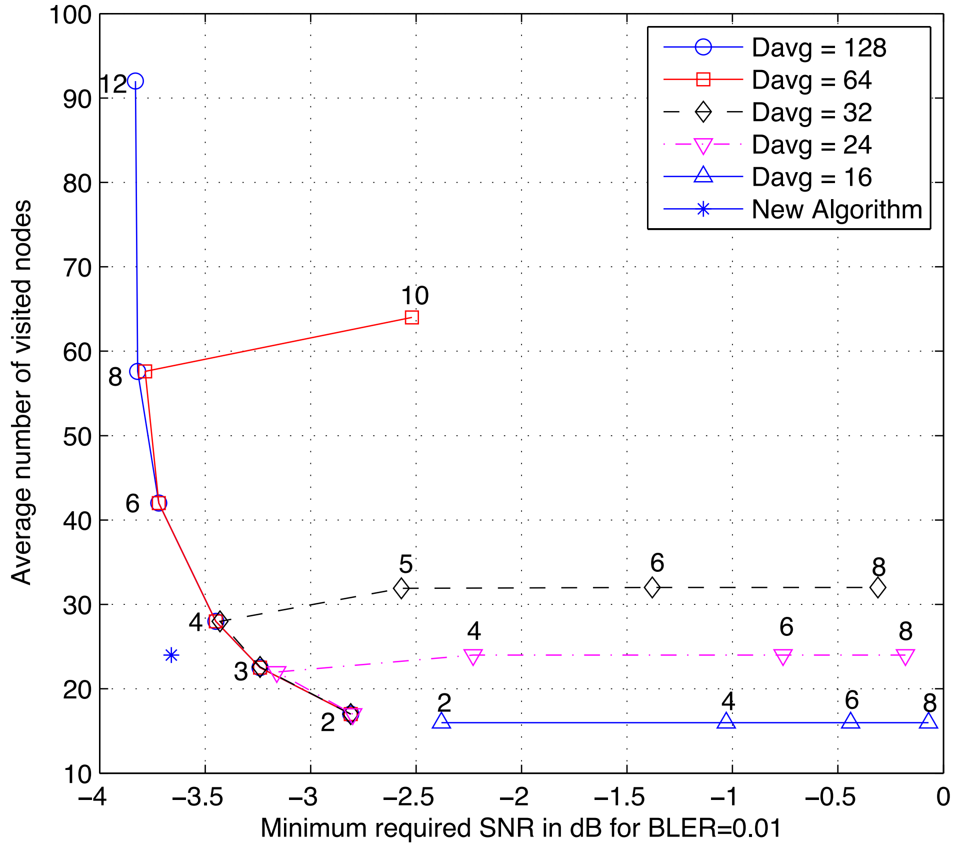


Figure 9: New tree traversal algorithm for the QPSK modulation.

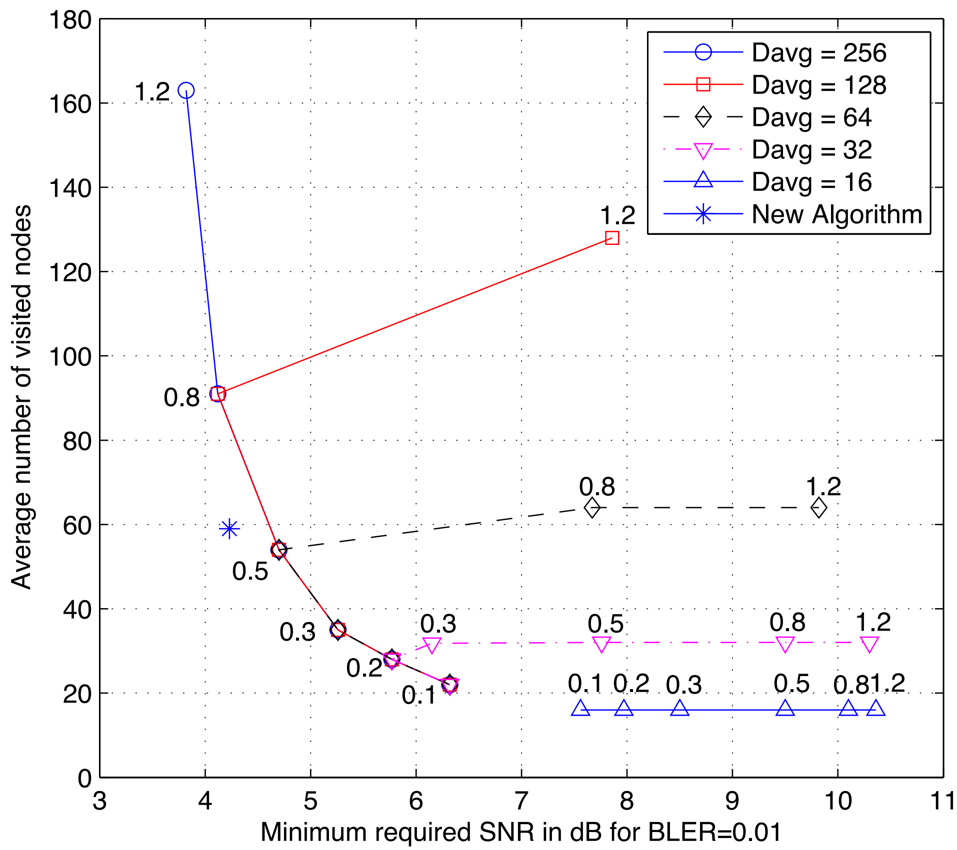


Figure 10: New tree traversal algorithm for the 16-QAM modulation.

Outline

- Introduction
- System Model
- Soft-Output Sphere Decoding Algorithm
- Methods for Complexity Reduction
- New Tree Traversal Strategy
- Simulation Results
- Conclusion and Future work

- This thesis presents new soft-output algorithms for MIMO systems as a support to an outer coder.
- At the current stage, we only deal with the QPSK and 16-QAM modulations. It should be interesting to see whether there exists a systematical method that can be extendably applied to to all square QAM modulations.