SMART CANDIDATE ADDING: A NEW LOW-COMPLEXITY APPROACH TOWARDS NEAR-CAPACITY MIMO DETECTION

P. Marsch, E. Zimmermann, G. Fettweis

Vodafone Chair Mobile Communications Systems, Dresden University of Technology
Department of Electrical Engineering and Information Technology, 01062 Dresden, Germany
email: {marsch,zimmere}@ifn.et.tu-dresden.de

ABSTRACT

Recently, communication systems with multiple transmission and reception antennas (MIMO) have been introduced and proven to be suitable for achieving a high spectral efficiency. Assuming full channel knowledge at the receiver, so-called sphere detectors have been shown to solve the maximum likelihood detection problem at acceptable complexity. In the context of coded transmission, however, the detector has to generate soft output information for every transmitted bit. Existing detectors provide this by observing a high number of hypotheses about the transmitted symbol, which is computationally expensive.

In this paper, we introduce *Smart Candidate Adding*, a new scheme that performs multiple directed searches to obtain only a small set of symbol hypotheses, but with a good representation of all possibly transmitted bit constellations. Simulation shows that the new approach outperforms conventional schemes, both in terms of detection performance and computational complexity.

1. INTRODUCTION

In modern mobile communications systems, it is essential to use bandwidth very efficiently, i.e. to transmit many data bits per channel access. Existing modulation schemes are limited due to power constraints and the achievable signal-to-noise ratio (SNR). An alternative are systems with multiple transmission and reception antennas (MIMO) that use spatial diversity for a parallel data transmission, achieving a higher spectral efficiency for the same SNR.

1.1 System Model

A linear channel with N_T transmission and N_R reception antennas can be defined by a complex matrix $\mathbf{C}_{N_R \times N_T}$. We assume the channel is fast Raleigh fading and ergodic, such that consecutive channel accesses observe a quasi-static channel, but a high number of accesses reveals the statistical properties of the channel, i.e. it is passive, and all elements are independent, identically distributed (i.i.d) Gaussian random variables with mean zero and $E\{|C_{ij}|^2\} = 1$. We also introduce a channel representation with real elements

$$\mathbf{H}_{M\times L} = \begin{bmatrix} \mathrm{Re}\{\mathbf{C}\} & -\mathrm{Im}\{\mathbf{C}\} \\ \mathrm{Im}\{\mathbf{C}\} & \mathrm{Re}\{\mathbf{C}\} \end{bmatrix}$$

with $M = 2N_R$ and $L = 2N_T$. A transmission can be stated as

$$\mathbf{y} = \mathbf{H}\mathbf{s} + \mathbf{n}$$

where $\mathbf{s} \in S$ is one of Q^L possible discrete signal vectors transmitted by the N_T antennas during one channel access. In this paper, we use $Q = 4$ or $Q = 8$ for 16-QAM or 64-QAM modulation, respectively. Transmitted symbols are normalized so that $E\{s_l^2\} = 1$, and $\mathbf{n}$ represents additive white Gaussian noise (AWGN) that appears at the receiver, defined as real i.i.d distributed Gaussian random variables with zero mean and $E\{n_l^2\} = \sigma^2 = \frac{N_0}{2}$. For our simulations, we can then calculate the SNR, based on the code rate R, as

$$\mathrm{SNR\ [dB]} = 10 \cdot \log_{10} \frac{E\{E_b\}}{E\{N_0\}} = 10 \cdot \log_{10} \frac{1}{\log_2 Q \cdot R \cdot N_0}$$

1.2 The Integer Least Squares Problem

A common detection problem is to determine the vector, also referred to as *symbol*, that has been transmitted with the highest *a posteriori* probability, given the received signal. If all symbols are equiprobable, we can use Bayes' theorem to rewrite this problem as a search for the *maximum-likelihood* (ML) symbol, i.e.

$$\hat{\mathbf{s}}_{ML} = \arg\max_{\mathbf{s}\in S} p(\mathbf{s}|\mathbf{y}) \equiv \arg\max_{\mathbf{s}\in S} p(\mathbf{y}|\mathbf{s})$$

From our system model in section 1.1, we can derive

$$p(\mathbf{y}|\mathbf{s}) = \frac{1}{(2\pi\sigma^2)^{\frac{M}{2}}} \exp\left(-\frac{\|\mathbf{y}-\mathbf{H}\mathbf{s}\|^2}{2\sigma^2}\right)$$

and conclude that finding the ML symbol is equivalent to solving the so-called *integer least-squares problem* [1]

$$\hat{\mathbf{s}} = \arg\min_{\mathbf{s}\in S} \|\mathbf{y}-\mathbf{H}\mathbf{s}\|^2 \quad (1)$$

2. SPHERE DETECTION

A solution for equation (1) could be approximated by *linear equalization* [2]. However, the result is not optimal any more after the final quantization. So-called *sphere detectors* alleviate the problem by discretizing every detected real signal component (i.e. every signal *layer*), before proceeding with the next component.

2.1 QR-decomposition

We assume $M \geq L$, and can then mathematically transform the channel matrix by performing a so-called QR-decomposition, so that

$$\mathbf{H} = \mathbf{Q}\begin{bmatrix} \mathbf{R} \\ \emptyset \end{bmatrix} = \begin{bmatrix} \mathbf{Q}_1 & \mathbf{Q}_2 \end{bmatrix}\begin{bmatrix} \mathbf{R} \\ \emptyset \end{bmatrix}$$

where $\mathbf{Q}_{1,M\times L}$ and $\mathbf{Q}_{2,M\times(M-L)}$ are orthonormal matrixes, $\mathbf{R}_{L\times L}$ is upper triangular, and $\emptyset_{(M-L)\times L}$ is a zero matrix. We can then rewrite equation (1) to [3]

$$\begin{aligned} \hat{\mathbf{s}} &= \arg\min_{\mathbf{s}\in S} \left\| \begin{bmatrix} \mathbf{Q}_1^T \\ \mathbf{Q}_2^T \end{bmatrix}\mathbf{y} - \begin{bmatrix} \mathbf{R} \\ \emptyset \end{bmatrix}\mathbf{s} \right\|^2 \\ &= \arg\min_{\mathbf{s}\in S} \left\|\mathbf{Q}_1^T\mathbf{y} - \mathbf{R}\mathbf{s}\right\|^2 + \left\|\mathbf{Q}_2^T\mathbf{y}\right\|^2 \equiv \arg\min_{\mathbf{s}\in S} \left\|\mathbf{y}' - \mathbf{R}\mathbf{s}\right\|^2 \end{aligned} \quad (2)$$

where the term $\left\|\mathbf{Q}_2^T\mathbf{y}\right\|^2$ can be omitted in the last step, as it is independent of $\mathbf{s}$ and has no impact on the *arg min* operation. The upper triangular form of matrix $\mathbf{R}$ now allows us to iteratively calculate estimates for the originally transmitted signals $s_L, s_{L-1}, ..., s_1$ as

$$\tilde{s}_l = \frac{y'_l - \sum_{m=l+1}^{L} r_{lm} \cdot \hat{s}_m}{r_{ll}} \quad (3)$$

and perform a discretization $\hat{s}_l = \lfloor \tilde{s}_l \rceil$, then used for calculating $\tilde{s}_{l-1}$. After processing all L layers we obtain a full signal vector $\hat{\mathbf{s}}$, referred to as a *candidate*. Alternative discrete signals can be chosen in each layer to create a *search tree*, leading to multiple candidates, i.e. different hypotheses on transmitted symbols. We denote the set of candidates as C, the corresponding set of data vectors as X.

2.2 Limiting Sphere Searches through Metrics

From equations (2) and (3), we can derive that [3]

$$\begin{aligned} \|\mathbf{y}-\mathbf{H}\hat{\mathbf{s}}\|^2 &= \left\|\mathbf{Q}_1^T\mathbf{y}-\mathbf{R}\hat{\mathbf{s}}\right\|^2 + \left\|\mathbf{Q}_2^T\mathbf{y}\right\|^2 \\ &= \sum_{l=1}^{L}\left(y'_l - \sum_{m=l}^{L} r_{ml}\hat{s}_m\right)^2 + K = \sum_{l=1}^{L}(\tilde{s}_l-\hat{s}_l)^2\cdot r_{ll}^2 + K \end{aligned} \quad (4)$$

where $K = 0$ if the number of transmission and reception antennas is equal, which we will assume for simplicity. Hence, the spacing $\Delta_l = |\tilde{s}_l - \hat{s}_l|$ between the calculated and discretized signal in every layer can be used to determine the Euclidian distance of the received signal $\mathbf{y}$ to the projection $\mathbf{H}\hat{\mathbf{s}}$ of the obtained candidate. This can be used as a metric to evaluate partial or completed paths in the search tree and to limit the search to those candidates within a predefined Euclidian distance R_0, by assuring in every detection step that

$$M(\hat{\mathbf{s}})_l^L = \sum_{m=l}^{L}(\hat{s}_m-\tilde{s}_m)^2\cdot r_{mm}^2 \leq \|\mathbf{y}-\mathbf{H}\hat{\mathbf{s}}\|^2 \leq R_0^2$$

2.3 Basic Sphere Search Concepts and Terms

We can state three common sphere search algorithms:

- The *Fincke-Pohst* algorithm [4, 5] performs a *depth-first* search and determines the possible discrete signal values $\hat{s}_l$ in each layer that fulfill the overall radius constraint. The search is continued with all of these values, starting with the lowest one.
- The *Schnorr-Euchner* algorithm [6] is similar, but the possible signals are ordered by their spacing to the calculated $\tilde{s}_l$, and the closest is processed first. This is done by selecting possible signals in an alternating fashion around the calculated point.
- In contrast, a *List-Sequential Sphere Detector* (LISS) [7] performs a *breadth-first* search by adding all new nodes to a large list, and always processing the node with the lowest metric, regardless which signal layer it represents. The found candidates will automatically be ordered by ascending metric, i.e. the *best* candidate will always be found first.

The first candidate found by the Schnorr-Euchner algorithm is known as the *Babai point* [8] $\hat{\mathbf{s}}_{BB}$, and is equivalent to the solution found by a detector based on *successive interference cancellation* [9, 10]. The candidate with the lowest metric, i.e. the highest a posteriori probability, is the *maximum-likelihood* point $\hat{\mathbf{s}}_{ML}$.

2.4 Advanced Search Strategies

Based on the algorithms described in the last section, we distinguish between four advanced search strategies:

- **A. Search within a fixed radius**. All candidates within the Euclidian distance R_0 from the received signal are searched.
- **B. Search for the best N candidates**. All candidates are sorted by their metric, and the search radius is always set to the Euclidian distance of the N-th best candidate. After the search, we will have found at least the N candidates with the best metric.
- **C. LISS Search for N candidates**. The LISS algorithm is used, until N candidates are found. Due to the properties of the algorithm, we can again be sure to have found the best N candidates.
- **D. Search for ML point**. The search radius is continuously reduced to that of every found candidate. The last found candidate will then always be the desired ML point.

All strategies in which the search radius is reduced should use the Schnorr-Euchner algorithm, as this outputs candidates with low metrics first and thus strongly reduces the size of the search tree.

2.5 Channel Preprocessing

The search complexity can also be reduced if the channel matrix $\mathbf{H}$ is preprocessed before QR-decomposition. The columns of the matrix should be ordered by increasing SNR, so that the more reliable layers are detected first, which can be done by a *sorted QR-decomposition* (SQRD) [11], ideally succeeded by a *post sorting algorithm* (PSA) [11] for a perfect ordering. Additionally, it is reasonable to consider the noise level during detection so that we achieve a *minimum mean square error* (MMSE) at the receiver. This can be done by extending the channel matrix to [11, 1]

$$\bar{\mathbf{H}}_{(M+L)\times L} = \begin{bmatrix} \mathbf{H} \\ \sigma\mathbf{I}_L \end{bmatrix} \quad \text{and} \quad \bar{\mathbf{y}}_{(M+L)\times 1} = \begin{bmatrix} \mathbf{y} \\ \mathbf{O}_{L\times 1} \end{bmatrix} \quad (5)$$

prior to QR-decomposition. This strongly decreases the complexity of a search for the ML point, but also leads to the fact that the search can output a false result and metrics will be calculated wrongly, as the derivation in (4) is not possible any more. In our simulations, we use sorted QR-decomposition, either with MMSE channel extension (*MMSE-SQRD*) or without (*ZF-SQRD*).

3. SOFT OUTPUT CALCULATION

For coded transmission, a detector has to provide *soft output*, i.e. a posteriori probabilities about each transmitted bit, given the received signal. These are stated as *log-likelihood ratios* (LLR), i.e.

$$L_C(x_k|\mathbf{y}) = \ln\frac{P(x_k=1|\mathbf{y})}{P(x_k=0|\mathbf{y})}$$

It is intractable to fully evaluate this equation, but we can approximate the result using the set X of found candidates

$$L_C(x_k|\mathbf{y}) \approx \ln\frac{\sum\limits_{\hat{\mathbf{x}}\in\{X|\hat{x}_k=1\}} p(\mathbf{y}|\hat{\mathbf{x}})}{\sum\limits_{\hat{\mathbf{x}}\in\{X|\hat{x}_k=0\}} p(\mathbf{y}|\hat{\mathbf{x}})}$$

A problem arises if all candidates in the set have the same value for a certain bit, i.e. $\exists k,b \ : \ \forall\hat{\mathbf{x}}\in X : \hat{x}_k = b$. In this case there is no counter-hypothesis for the bit, leading to an infinite log-likelihood ratio. Different schemes exist in literature to deal with this problem:

- **Bit Flipping** [12]. A new candidate is created by taking $\hat{\mathbf{x}}_{ML}$ and flipping the desired bit value x_k. Its metric then has to be recalculated for all layers below and including the one in which x_k resides. However, due to the correlation between signal layers, there will most likely exist a better candidate fulfilling $x_k \neq b$, so that the probability $P(x_k \neq b)$ will be underestimated.
- **Radius Limitation or LLR Clipping** [13]. Again, a new candidate is created from $\hat{\mathbf{x}}_{ML}$, but now its metric is set to that of a virtual candidate positioned on the search radius. Alternatively, it is possible to limit the LLR values, e.g. $|L_D(x_k|\mathbf{y})| \leq \varepsilon$. In both cases the computational effort is negligible, but the probability $P(x_k \neq b)$ might now be strongly overestimated.
- **Path augmentation** [7]. This scheme uses the information stored in the incomplete paths of the search tree. Any such path, fulfilling $x_k \neq b$, with signal elements $\hat{s}_L, \hat{s}_{L-1}, ..., \hat{s}_m$ can be completed to a full candidate by the unconstrained $(\tilde{s}_l)$ or discretized $(\hat{s}_l)$ linear equalization solution (ZF or MMSE), or by 'soft-mapping' available a priori information. The last two approaches require the calculation of a new metric, and the probability $P(x_k \neq b)$ might again be underestimated due to signal layer correlation. According to [7], metric calculation can be omitted when the unconstrained ZF-solution is used, as then $\forall 1 \leq l \leq m-1 : \Delta l = 0$, hence the metric does not increase

below layer m. This, however, can lead to a new candidate with a strongly overestimated a posteriori probability.

None of the approaches appears to be very accurate, and in some cases the metric calculations are computationally expensive. We thus introduce a new scheme that performs constrained searches to create only a small but very representative set of candidates.

4. SMART CANDIDATE ADDING (SCA)

The new scheme is based on the following steps:

1. We search for the ML point (strategy D in section 2.4). For 16-QAM, this leads to about 3 candidates on average for ZF-SQRD, and 1.5 candidates for MMSE-SQRD.
2. We then select bits that are not represented sufficiently by the obtained set of candidates, e.g. where no counter-hypothesis exists, or where the LLR value exceeds a certain limit.
3. For each of these bits, we perform so-called *constrained searches*, i.e. searches for the ML point using a modified Schnorr-Euchner algorithm, where the investigated bit is fixed to the desired value. We can determine the *constrained Babai point* $\hat{\mathbf{s}}_{BB}[x_k \neq b]$, or search for the *constrained ML point* $\hat{\mathbf{s}}_{ML}[x_k \neq b]$, i.e. the most likely candidate under the constraint $x_k \neq b$. We will refer to these search options as **SCA BB** and **SCA ML**. Other compromises are thinkable, e.g. to search for the constrained Babai point plus a predefined number of better candidates. Thus, the tradeoff between performance and complexity can be adjusted to the application's requirements.
4. For 16-QAM, an average of about 13 (ZF-SQRD) or 14 (MMSE-SQRD) constrained searches are performed, leading to a total of 16 candidates on average for SCA BB, or 34 (ZF-SQRD) or 25 (MMSE-SQRD) candidates for SCA ML.

Figure 1 illustrates a typical search tree after one bit has been investigated. The candidates are displayed as small circles; their distance to the tree center is equivalent to their Euclidian distance to the received signal. The grey candidates are from the original, unconstrained search, all containing $x_k = b$, the black ones have been added through a search constrained to $x_k \neq b$.

A constrained search only yields optimal results if the signal layer containing the investigated bit is detected first. Otherwise layers detected in advance could not be reasonably evaluated by their metric, i.e. a tree path within these layers might appear to have a very bad metric, but finally lead to the constrained ML candidate for the investigated bit. We thus suggest to perform L different QR-decompositions, so that each signal layer has the highest column index once, as done efficiently by the algorithm shown in table 1 and based on [11].

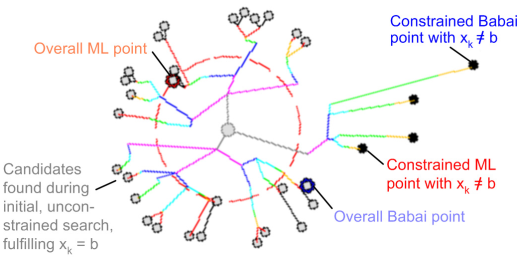

Figure 1: Typical search tree after smart candidate adding.

5. SIMULATION

We will compare the complexity and performance of a sphere search for the 50 or 250 best candidates using strategy B from section 2.4 to that of the new approach, in the versions SCA BB or SCA ML.

Multiple-SQRD()

Input:	Channel matrix $\mathbf{H}_{M\times L}$ (or $\bar{\mathbf{H}}_{(M+L)\times L}$ for MMSE)
Output:	Unitary $\mathbf{Q}_{i,M\times L}$ (or $\bar{\mathbf{Q}}_{i,(M+L)\times L}$), upper-triangular $\mathbf{R}_{i,L\times L}$, $\mathbf{P}_{i,L\times L}$
$\mathbf{R} := 0$, $\mathbf{Q} := \mathbf{H}$	initialize matrixes
for $(i := 1\ldots(L-1))$ {	all columns except last one
$\quad\mathbf{Q}_i := \mathbf{Q}$; $\mathbf{R}_i := \mathbf{R}$; $\mathbf{P}_i := \mathbf{P}$	copy Q, R and P matrixes
$\quad$Partial-SQRD($\mathbf{Q}_i$, $\mathbf{R}_i$, $\mathbf{P}_i$, i, L, true)	do special QR decomposition
$\quad$Partial-SQRD($\mathbf{Q}$, $\mathbf{R}$, $\mathbf{P}$, i, i, false)	do next step of QR decomp.
}	
Partial-SQRD($\mathbf{Q}$, $\mathbf{R}$, $\mathbf{P}$, L, L, false)	do last step of QR decomp.
$\mathbf{Q}_L := \mathbf{Q}$; $\mathbf{R}_L := \mathbf{R}$; $\mathbf{P}_L := \mathbf{P}$	copy Q, R and P matrixes

Partial-SQRD()

Input:	$\mathbf{Q}$, $\mathbf{R}$ (or MMSE equiv.), $\mathbf{P}$, column indices i_1, i_2, boolean b_{toback}
Output:	Output is written back into input matrixes
for $(i := i_1 \ldots i_2)$ {	loop through desired columns
$\quad$if $(i = i_1 \vee \neg b_{toback})$	find column with smallest norm
$\quad\quad k := \arg\min_{k'=i_1\ldots L} \mathbf{q}_{k'}^T \mathbf{q}_{k'}$	by searching all columns
$\quad$else $k := \arg\min_{k'=i_1\ldots L-1} \mathbf{q}_{k'}^T \mathbf{q}_{k'}$	or all columns except last one
$\quad$if $(i = i_1 \wedge b_{toback})$ swap cols k, L in $\mathbf{P}$, $\mathbf{R}$ and first $L+i-1$ rows of $\mathbf{Q}$	
$\quad$else swap cols k, i in $\mathbf{P}$, $\mathbf{R}$ and first $L+i-1$ rows of $\mathbf{Q}$	
$\quad r_{ii} := \sqrt{\mathbf{q}_i^T \mathbf{q}_i}$	set entry to column norm
$\quad \mathbf{q}_i := \mathbf{q}_i / r_{ii}$	normalize column
$\quad$for $(k := i+1\ldots L)$ {	loop through remain. columns
$\quad\quad r_{ik} := \mathbf{q}_i^T \mathbf{q}_k$	determine column correlation
$\quad\quad \mathbf{q}_k := \mathbf{q}_k - r_{ik} \cdot \mathbf{q}_i$	remove correlation
$\quad$}	
}	

Table 1: Algorithm for multiple sorted QR-decompositions.

5.1 Setup

All simulations are based on Turbo coded transmission, using a standard PCCC with $(7_R, 5)$ constituent convolutional codes. The block length is 8920 bits and a code rate of $R = 1/2$ is obtained by alternately puncturing parity bits at the output of the constituent encoders. We use two different setups:

- **Decoder iterations**. After an initial detection process, the soft output is sent through 8 internal decoder iterations.
- **Detector iterations**. Again, 8 turbo decoder iterations are performed, but the first 4 are succeeded by additional detection processes based on a priori information from the last decoding step (see [7]). Thus, the initial detection is followed by one internal decoder loop, then the next detection is performed, again followed by one internal decoder loop, etc. Note that this setup deviates slightly from that of other papers, e.g. [13].

5.2 Performance

Figure 2 shows the BER of the approaches as a function of SNR. The respective Shannon limits have been added according to [13]. For 16-QAM and ZF-SQRD, a search for the best 250 candidates has its waterfall region at 8.2dB or 7.1dB, for pure decoder or detector iterations, respectively. This corresponds to [13], considering that we are using less candidates and a different setup.

In general, we can observe that conventional schemes can be improved by about 1.5dB if detector iterations are used. The new schemes, however, improve by about 3dB, as the constrained searches explore the signal space in a more sophisticated way, providing a higher amount of information to the decoder. When MMSE-SQRD preprocessing is employed instead of ZF-SQRD, the performance of conventional schemes is degraded by about 0.5dB, due to the mentioned falsely calculated metrics, whereas SCA ML performance remains fairly unchanged. SCA BB even improves, as preprocessing causes constrained Babai and ML points to often be equivalent, so that the difference between SCA BB and SCA ML diminishes. SCA can strongly outperform conventional sphere searches, especially with MMSE-SQRD preprocessing. The performance gap increases for 64-QAM modulation, as here a symbol contains more bits, and many of these will be under-represented in the set of candidates provided by a conventional search scheme.

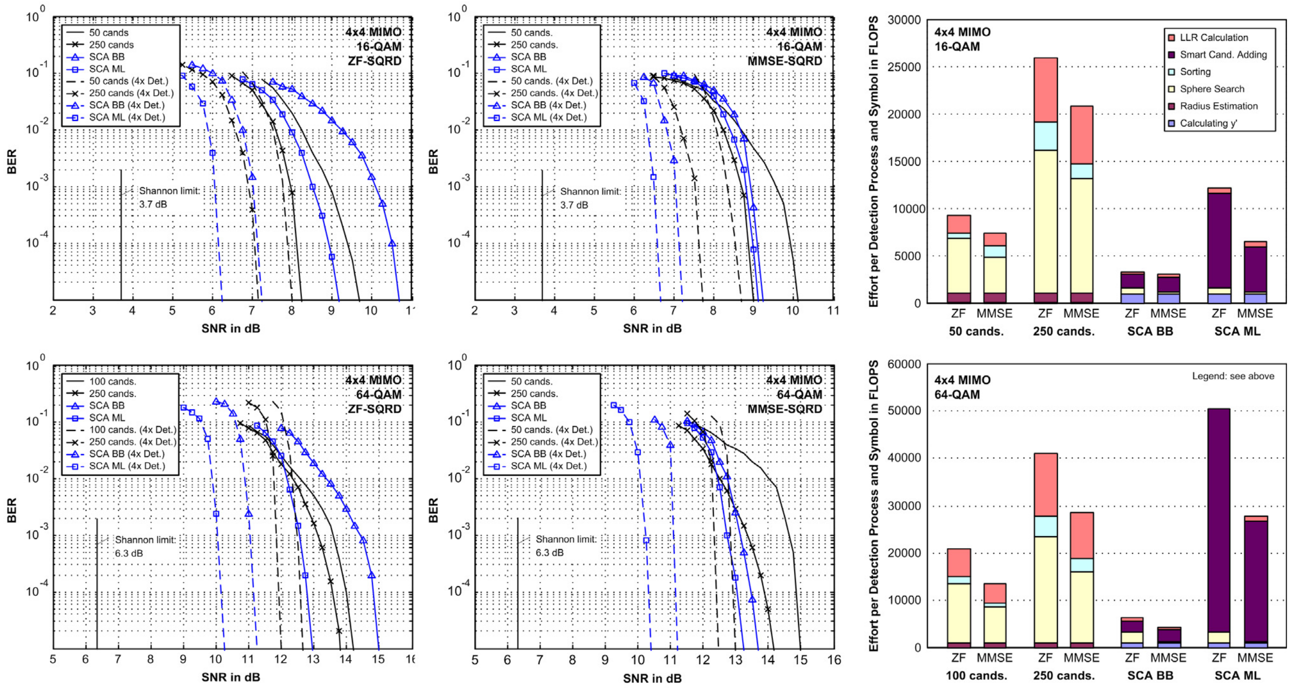

Figure 2: Performance and complexity measurements, for 16-QAM or 64-QAM modulation and ZF-SQRD or MMSE-SQRD preprocessing.

5.3 Complexity

Figure 2 also shows the average number of floating point operations needed for one detection process of one symbol, including data manipulation and comparison, but excluding data copying. The measurements have been performed at 8dB SNR (16-QAM) and 12dB SNR (64-QAM), i.e. signal-to-noise ratios close to the average waterfall region of the approaches. Note that for detector iterations, the stated effort has to be multiplied by 5. Performing sorted or multiple QR-decomposition requires 3052 or 15673 FLOPS, respectively, for a 4x4 MIMO system using MMSE-SQRD, and should thus be negligible if the channel remains constant over multiple symbols.

We can see that MMSE-SQRD preprocessing decreases the complexity of all approaches. The SCA approaches generally appear to be very compact, even though the performance is often comparable to or even exceeds that of conventional searches with a higher complexity. SCA ML appears to be unsuitable for 64-QAM modulation, as the number of investigated bits is high and complex constrained searches have to be performed. However, the SCA BB approach using MMSE-SQRD preprocessing attracts our attention, as it is very compact and predictable, as a search for a constrained Babai point has a determinable complexity per investigated bit. It reaches (16-QAM) or even outperforms (64-QAM) a conventional search for 250 candidates, even though it requires less than 15% of the respective computational effort.

6. CONCLUSIONS

The simulations have shown that *smart candidate adding* can achieve a better ratio of performance per computational effort for the detection of complex symbols within applications using coded transmission. Especially the scheme SCA BB has proven to be very compact and determinable and can nevertheless compete in performance with conventional sphere searches involving a much higher complexity. A strong performance improvement is observable when detector iterations are used, due to the higher amount of information provided by an SCA detector compared to conventional schemes. The additional preprocessing effort required for *smart candidate adding*, to supply multiple QR-decompositions, is low compared to the general effort needed for detection and decoding, especially if the channel remains constant over a long period of time, as can for example be expected in short range scenarios with low mobility.

REFERENCES

[1] B. Hassibi and H. Vikalo, *On the expected Complexity of Sphere Decoding*, Record of the Thirty-Fifth Asilomar Conf. on Signals, Systems and Computers (2001), 1051–1055.

[2] J. G. Proakis, *Digital communications*, 4 ed., McGraw-Hill, 2000.

[3] M. O. Damen, H. El Gamal, and G. Caire, *On Maximum-Likelihood Detection and the Search for the closest Lattice Point*, IEEE Transactions on Information Theory **49** (2003), no. 10, 2389–2402.

[4] M. Pohst, *On the Computation of Lattice Vectors of minimal Length, successiv Minima and reduced Bases with Applications*, ACM SIGSAM Bulletin **15** (1981), 37–44.

[5] U. Fincke and M. Pohst, *Improved Methods for calculating Vectors of short Length in a Lattice, including a Complexity Analysis*, Math. of Comput. **44** (1985), 463–471.

[6] C. P. Schnorr and M. Euchner, *Lattice Basis Reduction: Improved practical Algorithms and solving Subset Sum Problems*, Math. Prog. **66** (1994), 181–191.

[7] S. Bäro, J. Hagenauer, and M. Witzke, *Iterative Detection of MIMO Transmission using a List-Sequential (LISS) Detector*, IEEE International Conference on Communications **4** (2003), 2653–2657.

[8] L. Babai, *On Lovasz' Lattice Reduction and the nearest Lattice Point Problem*, Combinatorica **6** (1986), no. 1, 1–13.

[9] A. J. Viterbi, *Very Low Rate Convolutional Codes for Maximum Theoretical Performance of Spread-Spectrum Multiple-Access Channels*, IEEE JSAC **8** (1990), no. 4, 641–649.

[10] R. Kohno et al., *Combination of an Adaptive Array Antenna and a Canceller of Interference for Direct-Sequence Spread-Spectrum Multiple-Access System*, IEEE JSAC **8** (1990), no. 4, 675–682.

[11] D. Wübben, R. Böhnke, V. Kühn, and K.-D. Kammeyer, *MMSE Extension of V-BLAST based on Sorted QR Decomposition*, IEEE Semiannual Vehicular Technology Conference **1** (2003), 508–512.

[12] R. Wang and G. B. Giannakis, *Approaching MIMO Channel Capacity with Reduced-Complexity Soft Sphere Decoding*, subm. July 2003, rev. Oct. 2004.

[13] B. M. Hochwald and S. ten Brink, *Achieving Near-Capacity on a Multiple-Antenna Channel*, IEEE Trans. on Communications **51** (2003), no. 3, 389–399.