Sparsity-Aware Successive Interference Cancellation with Practical Constraints

Benjamin Knoop*, Fabian Monsees**, Carsten Bockelmann**, Dirk Wübben**, Steffen Paul*, Armin Dekorsy**

*Institute of Electrodynamics and Microelectronics (ITEM)
University of Bremen, Germany
Email: {knoop, paul}@me.uni-bremen.de

**Department of Communications Engineering
University of Bremen, Germany
Email: {monsees, bockelmann, wuebben, dekorsy}@ant.uni-bremen.de

***Abstract*—This paper investigates new detection methods for low data-rate Machine Type Communication. Special attention is paid to uplink communication in networks where nodes are only active occasionally, which requires a joint activity and data detection at the receiver. This paper shows that a modified version of Successive Interference Cancellation is a viable approach for the detection of these low data-rate multi-user signals. The analysis is in the context of a hardware implementation, which we address particularly as we investigate the algorithmic complexity of the introduced algorithms.**

I. Introduction

Machine Type Communication (MTC) is a fast growing field and recently gained considerable attention [1]. Applications such as smart metering, medical systems or logistics constantly raise new demands on existing communication systems such as LTE [2]. Integrating these mostly low-data rate services into existing infrastructures is a major concern. Various MTC applications require only low data rate which makes extensive signaling overhead and complex scheduling impracticable. A potential physical layer approach towards reduced signaling is to avoid activity signaling. This requires activity detection of the nodes by the receiver itself. The detection of sporadic MTC was initially adressed in [3]. Here inactive nodes are modeled as transmitting zero symbols and the detector utilizes an augmented modulation alphabet for detection. This approach has been extended further in [4], [5], [6] where the authors investigated the performance of sphere decoding for sparse communication signals in a multi-user framework, which models an uplink Machine to Machine (M2M) scenario.

However, with focus on implementation methods, sphere detection is very costly and various suboptimal methods can nearly approach same performance with decreased complexity. It is well known that Successive Interference Cancellation (SIC) with proper pre-processing such as sorting [7] is a viable candidate for a suboptimal approach.

Aiming at a future hardware implementation, this paper investigates the recovery of sparse communication signals under practical side conditions with a low-complexity SIC. As a major contribution we show that in a sparse communication system SIC nearly approaches the performance of Maximum a posteriori (MAP) detection by utilizing efficient pre-processing algorithms prior to detection. In particular we investigate the application of the well known Sorted QR Decomposition (SQRD) [7] which guarantees that nodes with a high post detection SNR are detected first while aiming at decreased error propagation. Additionally, we introduce a data dependent sorting algorithm, which combines SQRD with the knowledge of sparse source signals. Motivated by ideas from Compresses Sensing [8], this approach outperforms the simple application of SQRD at the cost of additional algorithmic complexity.

Aiming at practical implementations of SIC, we investigate the impact of various side constrains on the performance. We show the impact of quantization to a fixed-point number representation limits the performance of the detector, and we address the minimal necessary quantization word length required for sufficient performance.

The remainder of this paper is organized as follows: Section II describes a general multi-user uplink system model which forms the basis for further investigations. In Section III we introduce the Sparsity-Aware SIC (SA-SIC) as suboptimal detector and investigate the algorithmic complexity. Section IV introduces potential pre-processing steps before SIC detection for increased performance. We also investigate the algorithmic complexity of the introduced pre-processing strategies. In Section V a Code Division Multiple Access (CDMA) multi-user system is introduced to test the performance of the algorithms. We examine the performance under fixed point operations as a step towards a hardware realization in Section VI which we also investigate in detail in Monte-Carlo simulations in Section VII. Section VIII finally concludes the paper.

II. System Model

The main contributions of this work are of general nature and can be applied to a variety of communication systems. We

This work was founded by the German Research Foundation (DFG) under grants PA 438/8-1 and DE 759/3-1.

,however, restrict ourself to the multi-user uplink case where a set of K nodes transmit data to a central aggregation node for further processing. This uplink model is summarized as follows

$$\mathbf{y} = \mathbf{T}\mathbf{x} + \mathbf{w}_n. \tag{1}$$

Here $\mathbf{y} \in \mathbb{C}^K$ is the vector of observations and $\mathbf{T} \in \mathbb{C}^{K\times K}$ summarizes the channels between the nodes and the aggregation node. We assume superposition of additive white Gaussian noise with zero mean and variance σ_n^2. The source vector contains the modulation symbols from the nodes and the kth entry corresponds to the modulation symbol of node k.

In this setup we employ a simple activity model for the nodes which is parametrized by the per node activity probability p_a that we assume equal for all nodes in the system. Each node is active with p_a and transmits symbols from the modulation alphabet $\mathcal{A}$. Inactive nodes do not transmit anything and are modeled as transmitting a zero symbol. The application of this model makes $\Pr(x_k = 0) = 1 - p_a$ and $\Pr(x_k \in \mathcal{A}) = p_a$. If p_a is sufficiently small, the source vector $\mathbf{x} \in \mathcal{A}_0^K$ is a sparse vector containing a considerable number of zero symbols. The main idea is to detect $\mathbf{x}$ with respect to the augmented symbol alphabet $\mathcal{A}_0 = \mathcal{A} \cup \{0\}$, which can be interpreted as a joint activity and data estimation.

III. Sparsity-Aware Successive Interference Cancellation (SA-SIC)

This section provides the formulation of the Sparsity-Aware Maximum a Posteriori (S-MAP) detector for the problem in (1). However, aiming at efficient hardware implementations, we restrict ourselves to suboptimal strategies and take the S-MAP performance as benchmark. In particular, we introduce the Sparsity-Aware SIC (SA-SIC) which we investigate with different types of pre-processing to nearly achieve S-MAP performance.

A. S-MAP Detector

In [3] the authors derived the S-MAP detector for the above described system model which solves

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}\in\mathcal{A}_0^K} \; \|\mathbf{y} - \mathbf{T}\mathbf{x}\|_2^2 + \lambda \sum_{k=0}^{K} \mathbf{1}_{\mathcal{A}}(x_k)\,. \tag{2}$$

The penalty term

$$\lambda = 2\sigma_n^2 \log\left(\frac{1-p_a}{p_a/|\mathcal{A}|}\right) \tag{3}$$

reflects the a priori statistics of the data vector and scales with the noise power σ_n^2 [4], [6]. $\mathbf{1}_{\mathcal{A}}(\cdot)$ is the indicator function which takes the value one if the argument is contained in the set $\mathcal{A}$ and zero otherwise. Problem (2) can efficiently be solved by sphere decoding [9].

```
function SA-SIC( $\tilde{\mathbf{y}}, \tilde{\mathbf{R}}, \sigma_n, p_a$ )
    $\lambda \leftarrow 2\sigma_n^2 \log((1-p_a)/(p_a/|\mathcal{A}|))$
    for $k \leftarrow K, ..., 1$ do
        $z \leftarrow \sum_{j=k+1}^{K} r_{kj}\hat{x}_j$                ▷ residual
        for all $x \in \mathcal{A}_0$ do                  ▷ symbol hypotheses
            $d_x \leftarrow (\tilde{y_k} - \tilde{r}_{kk}x - z)^2 + \lambda|x|$      ▷ metric
        end for
        $\hat{x}_k \leftarrow \arg\min_{\forall x\in\mathcal{A}_0} d_x$               ▷ decision
    end for
    return $\hat{\mathbf{x}}$
end function
```

Fig. 1. Pseudocode of the SA-SIC detection algorithm for real-valued, zero-augmented modulation schemes.

B. Sparsity-Aware SIC (SA-SIC) Detection

To approximately solve the S-MAP problem (2), we introduce SA-SIC as a suboptimal but far less complex solution in this paper. All variables are real-valued, as we focus on Binary Phase-Shift Keying (BPSK) modulation.

We use the well known QR decomposition (QRD), applied to the system matrix $\mathbf{T}$ for decomposing $\mathbf{T} = \mathbf{QR}$, where $\mathbf{Q}$ is column orthonormal and $\mathbf{R}$ is upper triangular [10]. With the help of this, we can rewrite (2) as

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}\in\mathcal{A}_0^K} \; \|\tilde{\mathbf{y}} - \mathbf{R}\mathbf{x}\|_2^2 + \lambda \sum_{k=0}^{K} \mathbf{1}_{\mathcal{A}}(x_k)\;, \tag{4}$$

where $\tilde{\mathbf{y}} = \mathbf{Q}^H\mathbf{y}$. For practical implementations, (4) has to be reformulated with scalar quantities. Thus, $\hat{\mathbf{x}}$ will be the detection result of

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}\in\mathcal{A}_0^K} \left[\sum_{k=1}^{K}\left(\tilde{y}_k - \sum_{j=k}^{K} r_{kj}x_j\right)^2 + \lambda\mathbf{1}_{\mathcal{A}}(x_k)\right]. \tag{5}$$

The sum in front of the of the indicator function could be combined with the sum of the l_2-norm definition.

The application of QRD enables an successive detection approach, which is well-known SIC detection, a back-substitution algorithm. Since (5) demands a minimization of a sum over all k, each summand can be minimized separately. Thus, the problem can be divided into K smaller, partial problems

$$\hat{x}_k = \arg\min_{x\in\mathcal{A}_0} \underbrace{\left[(\tilde{y}_k - r_{kk}x_j - \sum_{j=k+1}^{K} r_{kj}\hat{x}_j)^2 + \lambda\mathbf{1}_{\mathcal{A}}(x)\right]}_{d_x}. \tag{6}$$

Note, that the minimization is dependent on symbols $\hat{x}_j, j > k$. Therefore, the detection process must begin with the highest index, $k := K$, to detect $\hat{x}_K$. Eq. (6) can be evaluated for every possible symbol $x_k \in \mathcal{A}_0$, which leads to “symbol hypotheses” (x_i, d_{x_i}), $i = 1, ..., |\mathcal{A}_0|$, i.e. each symbol is associated with a metric. S-MAP detection rules that the hypothesis with least cost is the most probable one. Hence, the SA-SIC algorithm decides for the corresponding symbol as detection result. Thereupon, the index will be decreased by one, $k := K - 1$. SA-SIC detection continues until all $\hat{x}_i$, $i = K, ..., 1$ are detected.

Fig. 1 lists the SA-SIC algorithm in pseudocode notation. First of all, the S-MAP weight λ is precomputed. Then detection starts with the Kth user symbol and descends successively. The metric for each symbol hypothesis is regularized by the sparsity promoting extension $\lambda|x|$, with $|\cdot|$ being the indicator function $\mathbf{1}_{\mathcal{A}}(\cdot)$ for BPSK. Note, that the sum within the squared round brackets of eq. (5) (see also line 6 in Fig. 1) can be decomposed into $\tilde{r}_{kk}x_k + \sum_{j=k+1}^{K} \tilde{r}_{kj}x_j$, where the first summand is only dependent on the current transmit hypothesis x_k and the second summand only on previously decided symbols. The latter part, named z in the pseudocode, constitutes a residual. The last step of the SA-SIC is a hard decision for the hypothesis with the least cost. After K steps the algorithm terminates and returns the result vector $\hat{\mathbf{x}} \in \mathcal{A}_0^K$.

C. Complexity Comparison with Sparsity Aware Sphere Detection

The execution of the SA-SIC algorithm takes K steps, and in each iteration it computes the partial metric

$$d = (\tilde{y}_k - \tilde{r}_{kk}x_k - z)^2 + \lambda|x_k| \qquad (7)$$

for each possible symbol of the modulation alphabet (see line 6 in Fig. 1).

To evaluate (7) once, two subtractions, a partial vector multiplication for the residual z, and a multiplication for the square are needed. For nonzero symbols one more addition has to be considered to add the S-MAP weight λ. Note, that $\tilde{r}_{kk}x_k$ and $\lambda|x_k|$ are not genuine multiplications since x_k is a constant factor from the very limited set $\mathcal{A}_0$. The sign bit of x_k determines whether these mathematical expressions are added or subtracted and are thus just a simple conversion of the sign bit of $\tilde{r}_{kk}$ or λ respectively.

The partial vector multiplication for the residual takes $[K - (k + 1)]$ multiplications and $[K - k]$ additions. For the same reasons these are not genuine multiplications as well, and we can approximate the necessary operations as

$$C_z \approx \sum_{k=1}^{K} (K - k) C_{\text{Add}} = \frac{K^2 + K}{2} C_{\text{Add}} \,, \qquad (8)$$

where C_{Add} denotes the cost of one addition (or subtraction).

The cost to compute a whole SA-SIC is therefore proportional to

$$C_{\text{SA-SIC}} \propto K|\mathcal{A}_0| \left(C_{\text{Mul}} + C_z + 2C_{\text{Add}} \right) \,, \qquad (9)$$

where C_{Add} denotes the cost of one multiplication. This makes $C_{\text{SIC}} = \mathcal{O}(K^2)$ due to the partial vector multiplication.

Because the number of users and the modulation alphabet are both fixed quantities in a system setup, the algorithmic complexity and the runtime of the SA-SIC is constant. In contrast to this, this is not true for optimal S-MAP detection by a Sphere Decoder (SD). Sphere detection is closely related to SA-SIC, because it operates on the very same problem formulation of eq. (5). The only difference to SA-SIC detection is, that previously made symbol decisions can be dismissed later for an overall better solution. This seemingly minor variation makes the SD result on the one hand optimal, but on the other hand the runtime is not deterministic and varies. A measure of runtime of a SD is the number of processed nodes in the search tree, here denoted as n_{SD}. Fig. 2 shows, that n_{SD} becomes huge for low to medium SNR ranges and exhibits a strong exponential behavior. In distinction to eq. (9), the complexity of SD is

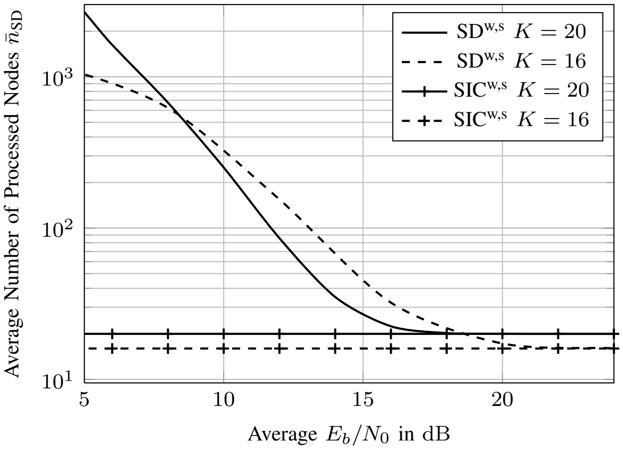

Fig. 2. Sphere decoder complexity in terms of processed nodes n_{SD}. The solid line represents an overdetermined system, whereas the dashed line a fully-loaded system (see section VII).

$$C_{\text{SD}} \propto \frac{\bar{n}_{\text{SD}}(E_b/N_0)}{K} \cdot C_{\text{SA-SIC}} \,. \qquad (10)$$

The detection result (SER) of SD is invariant of sorted or unsorted pre-processing, but of course sorting would decrease n_{SD} in comparison to unsorted QRD. Therefore, Fig. 2 plots the curves for SD with SQRD as pre-processing. Hence, a fair comparison between the complexities of the detectors can be drawn. We will show, that suboptimal SA-SIC detection leads to a drastically reduced algorithmic complexity at small detection performance loss only.

IV. Sorting Strategies

A. Sorted QR Decomposition

In this section two different pre-processing algorithms are designed which significantly enhance the SA-SIC performance. As the focus of this paper is the recovery of sparse source vectors, we exploit this knowledge for enhanced pre-processing. Additionally, we investigate the performance of the well known post detection SNR dependent sorting algorithm based on the application of the Sorted QR Decomposition (SQRD) which we use in both pre-processing algorithms. The Major key point of the SQRD is to decompose the system matrix such that

$$\mathbf{T\Theta} = \mathbf{QR} \Leftrightarrow \mathbf{T} = \mathbf{QR\Theta}^T \qquad (11)$$

Here $\mathbf{\Theta}$ denotes a binary square orthonormal permutation matrix [10] which simply determines the ordering of the detection later on. The major advantage is that detection ordering is optimized such that nodes with a high post detection SNR are detected first [7]. This strategy aims to minimize the probability of error propagation during the successive detection process. Additionally, it is sufficient to re-calculate the SQRD only if $\mathbf{T}$ changes which shows the independence of the sorting scheme from the currently transmitted data. In contrast, we subsequently introduce a data dependent sorting scheme, where the instantaneous vector of observations $\mathbf{y}$ influences the pre-processing.

B. Data-Dependent Sorting and Regularization

In the following we formulate a Data-Dependent Sorting and Regularization (DDS) algorithm which is inspired by Compressed Sensing. We begin with the system description (1) and recall the S-MAP detector stated in (2). The a-priori statistics of the source data $\mathbf{x}$ is reflected by the penalty term λ added to the optimization problem for each element $x_i \in \mathcal{A}$. In the following we restrict ourself to constant modulus data $|x_i| = 1$. Our goal is to adaptively weight the sparsity parameter λ for each element x_i which is derived from the instantaneous received signal $\mathbf{y}$. We thus rewrite the penalty term of (2) as

$$\lambda \sum_{k=0}^{K} \mathbf{1}_{\mathcal{A}}\left(x_i\right) = \lambda \|\mathbf{x}\|_2^2 = \left\|\mathrm{diag}\left(\sqrt{\boldsymbol{\lambda}}\right)\mathbf{x}\right\|_2^2 . \tag{12}$$

In (12) we utilize the definition of the l_2 norm for regularization, moreover, we replace the scalar λ by the vector $\boldsymbol{\lambda}$, where the elements $\lambda_i = \lambda$ have the same value. The diag $(\cdot)$ builds a matrix with the vector on the main diagonal and zero off diagonals. λ_i reflects the a-priori statistics of the source data and is equal for all elements x_i. Our goal is to re-scale λ_i based on the instantaneous activity pattern which we subsequently estimate from the observation $\mathbf{y}$. For getting a rough estimate about the current activity pattern we employ parts of an Orthogonal Matching Pursuit (OMP) [11]. The OMP is a greedy CS algorithm that iteratively estimates the set of active elements and the corresponding values in a CS problem. The OMP uses the correlation between the received signal $\mathbf{y}$ and the columns of the system matrix $\mathbf{T}$ as a measure for the likelihood which columns of $\mathbf{T}$ are contained in $\mathbf{y}$. This can be expressed as

$$\mathbf{c} = \mathbf{T}_n^H \mathbf{y} . \tag{13}$$

The subscript $\mathbf{T}_n$ indicates that the columns of $\mathbf{T}$ are normalized to unit vectors prior to the correlation [12]. The magnitude of the ith element in $\mathbf{c}$ gives an indication whether the ith column of $\mathbf{T}$ is contained in $\mathbf{y}$ or not. If so, it is much likely that the magnitude of c_i is large. Therefore, we sort $\mathbf{c}$ in a descending order, which can be summarized as $\mathbf{c}_s = \mathbf{\Psi}\mathbf{c}$, where $\mathbf{\Psi}$ is the corresponding permutation matrix.

In the following we utilize the sorting obtained by correlation to influence the a-priori assumption contained in $\boldsymbol{\lambda}$. This approach aims to increase λ_i if c_i is low and to decrease λ_i if c_i is high. This heuristic approach ensures that the penalty term is not solely determined by a-priori assumption but also contains information about the instantaneous activity pattern.

```
function DATA-DEPENDENT_SORTING( $\mathbf{y}_w, \mathbf{T}_n, \sigma_n, p_a$ )
    $\lambda \leftarrow 2\sigma_n^2 \log((1-p_a)/(p_a/|\mathcal{A}|))$
    $\mathbf{c} \leftarrow \mathbf{T}_n^H \mathbf{y}_w$                          ▷ correlation
    IndexSet ← sort ($\mathbf{c}$, 'descending')
    $\boldsymbol{\lambda}' \leftarrow$ linspace $(0, 2\lambda, K)$
    $\boldsymbol{\lambda}'_s \leftarrow \boldsymbol{\lambda}'$ (IndexSet)                  ▷ permutation
    $\mathbf{y}_0 \leftarrow [\mathbf{y}; \text{zeros}\,(K,1))]$
    $\mathbf{T}_{\lambda'_s} \leftarrow \left[\mathbf{T}; \text{diag}\left(\sqrt{\boldsymbol{\lambda}'_s}\right)\right]$   ▷ augment system matrix
    $\mathbf{Q}, \mathbf{R}, \mathbf{\Theta} \leftarrow$ sqrd $\left(\mathbf{T}_{\lambda'_s}\right)$
    return $\mathbf{y}_0, \mathbf{Q}, \mathbf{R}, \mathbf{\Theta}$
end function
```

Fig. 3. Matlab-like pseudocode of the Data Dependent Sorting and Regularization (DDS) pre-processing

Subsequently, we introduce $\boldsymbol{\lambda}'$ which contains linearly scaled elements and has the form $\boldsymbol{\lambda}' = [0 \cdots 2\lambda]^T$. We further permute the elements such that $\boldsymbol{\lambda}'_s = \mathbf{\Psi}\boldsymbol{\lambda}'$. As mentioned previously, this guarantees that λ_i is small if the probability that x_i is active is sufficiently high. Additionally, λ_i is high if the activity probability of x_i is low. Note that the scaling and the application of the correlative measure are based on heuristics and do not really reflect a-posteriori probabilities.

We now utilize the sorted vector $\boldsymbol{\lambda}'_s$ to regularize the the S-MAP problem and obtain

$$\begin{aligned}\hat{\mathbf{x}} &= \arg \min_{\mathbf{x} \in \mathcal{A}_0^K} \left\| \begin{bmatrix} \mathbf{y} \\ \mathbf{0}_K \end{bmatrix} - \begin{bmatrix} \mathbf{T} \\ \mathrm{diag}\left(\sqrt{\boldsymbol{\lambda}'_s}\right) \end{bmatrix} \mathbf{x} \right\|_2^2 & (14) \\ &= \arg \min_{\mathbf{x} \in \mathcal{A}_0^K} \left\| \mathbf{y}_0 - \mathbf{T}_{\lambda'_s} \mathbf{x} \right\|_2^2 . & (15)\end{aligned}$$

Furthermore, we apply SQRD on the augmented system matrix, $\mathbf{T}_{\lambda'_s}\mathbf{\Theta} = \mathbf{Q}\mathbf{R}$, which leads to a triangular system description yielding

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x} \in \mathcal{A}_0^K} \left\| \tilde{\mathbf{y}}_0 - \mathbf{R}\mathbf{\Theta}^H \mathbf{x} \right\|_2^2 . \tag{16}$$

By this approach, post detector SNR dependent sorting is improved by the estimation of the current data pattern. This type of pre-processing first collects information about the instantaneous activity pattern and promotes sparsity only at elements in x_i which are most likely inactive according to a correlative measure.

The algorithmic structure of this kind of pre-processing is also listed as Matlab-like pseudocode in Fig. 3. This pre-processing strategy requires to perform sorting and regularization each time a new vector of observations $\mathbf{y}$ is received and the algorithmic complexity is higher compared to the scheme introduced in section IV-A. Moreover, this scheme restricts to constant modulus signals.

C. Complexity Comparison of the Sorting Algorithms

The overall detection scheme of pre-processing and SA-SIC can be examined independently of each other. Section III-C compared the algorithmic complexity of the detection algorithms, whereas in this section we will offer a comparison between the discussed pre-processing strategies.

In [13], the author compares QRD and SQRD of a matrix (here adopted for a square matrix of dimensions $K \times K$) and states their algorithmic complexity as

$$C_{\mathcal{F},\mathrm{QRD}} = 4K^3 - 2K^2 + K\,, \text{ and} \tag{17}$$

$$C_{\mathcal{F},\mathrm{SQRD}} = 4K^3 - \frac{5}{4}K^2 + \frac{1}{4}K\,, \tag{18}$$

where $\mathcal{F}$ denotes a complex floating-point operation (flop), as defined in the cited work. The difference in terms of complexity is therefore with

$$C_{\mathcal{F},\mathrm{SQRD}} - C_{\mathcal{F},\mathrm{QRD}} = \frac{3}{4}K(K-1) \tag{19}$$

only marginally – both algorithms exhibit the same order of complexity, $\mathcal{O}(K^3)$.

According to the pseudocode in Fig. 3, a single SQRD is necessary for DDS as well. Most of the lines describe algorithmically simple operations, like augmentation or permutation, which are neglectable. Line 5 describes in fact the definition of some constants only, which can already incorporate the later occurring square root, $\sqrt{\mathbf{\Lambda}'_s}$. Thus, only the correlation $\mathbf{T}_n^H\mathbf{y}_w$ and the following sorting add additional complexity for the DDS. $\mathbf{T}_n^H\mathbf{y}_w$ constitutes a full matrix-vector multiplication (MVM), which takes

$$\begin{aligned} C_{\mathrm{MWM}} &= n(mC_{\mathrm{Mul}} + (m-1)C_{\mathrm{Add}}) & (20)\\ &\approx K^2(C_{\mathrm{Mul}} + C_{\mathrm{Add}})\,, & (21)\end{aligned}$$

operations, which is equivalent to $C_{\mathcal{F},\mathrm{MWM}} = 2K^2$ flops. Advanced sorting algorithms, e.g. odd-even mergesort, feature an algorithmic complexity of $\mathcal{O}(n\log(n)^2)$ [14]. Therefore, we can approximate the additionally necessary complexity for DDS as

$$C_{\mathcal{F},\mathrm{DDS}} - C_{\mathcal{F},\mathrm{SQRD}} \approx \gamma_1 K^2 + \gamma_2 K\log(K)^2 \tag{22}$$

with some constants γ_1, γ_2. This leads to the conclusion that the introduced pre-processing steps by DDS are asymptotically marginal as well, since $C_{\mathrm{DDS}} = \mathcal{O}(K^3)$ stays the same. A drawback of DDS is that this pre-processing step has to be executed for each received data vector, since it is data-dependent, but for fast varying channels this disadvantage diminishes.

V. Verification with a CDMA System

This section proves the feasibility of the introduced schemes at a Code Division Multiple Access (CDMA) multi-user uplink network. Here K nodes transmit data sporadically to a central aggregation node. With CDMA, each node spreads its modulation symbol to a chip sequence of length N. Those are transmitted to a central aggregation node via a frequency selective Rayleigh fading channel with impulse response length L_h The detector filters the multi-user signal via a matched filter and detects at symbol rate. The node specific spreading and convolution with the underlying channel is summarized in the matrix $\mathbf{A}$ which eases the formulation of the matched filter by simply defining $\mathbf{A}^H$ as the corresponding filter that converts the chip rate signal back to symbol rate. The input-output relation of our multi-user CDMA system can be summarized as

$$\mathbf{y} = \mathbf{A}^H\mathbf{A}\mathbf{x} + \mathbf{A}^H\mathbf{w}_n\,. \tag{23}$$

Note that the CDMA system model (23) and the abstract system (1) are identical except the circumstance of correlated noise in (23). Thus, we introduce a low complex pre-whitening scheme which we efficiently combine with SQRD for keeping the detection ordering optimized at low additional overhead for pre-whitening.

A. Efficiently Combined Pre-Whitening and Sorting

Within this subsection we combine two pre-processing strategies for SA-SIC detection. Considering (23) obviously shows that the matched filter correlates the noise at symbol rate as the noise itself is filtered. It is therefore obvious that the noise has to be de-correlated prior to any further processing or detection. We show how to apply the SQRD [7] for joint pre-whitening and sorting. Additionally we perform sorting and pre-whitening by calculating only a single SQRD which is efficient in terms of algorithmic complexity. The description starts by investigating the symbol rate noise covariance matrix

$$\begin{aligned} \mathbf{\Phi}_{w_n w_n} &= \mathrm{E}\left(\mathbf{A}^H\mathbf{w}_n\mathbf{w}_n^H\mathbf{A}\right) & (24)\\ &= \sigma_n^2\mathbf{A}^H\mathbf{A}\,. & (25)\end{aligned}$$

To de-correlate the noise, we introduce the pre-whitening filter matrix $\mathbf{P}$, which is applied to the symbol rate model (23) yielding

$$\mathbf{y}_w = \mathbf{P}\mathbf{A}^H\mathbf{A}\mathbf{x} + \underbrace{\mathbf{P}\mathbf{A}^H\mathbf{w}_n}_{\tilde{\mathbf{w}}_n}\,. \tag{26}$$

The task of the pre-whitening filter is twofold. First, the filter de-correlates the noise prior to the application of a detector. Second, filtering with $\mathbf{P}$ leads to a sorted triangular system description which allows for optimized successive detection. Using SQRD as defined in (11), i.e.,

$$\mathbf{A}\mathbf{\Theta} = \mathbf{Q}\mathbf{R}\,, \tag{27}$$

we obtain

$$\mathbf{P} = \mathbf{R}^{-H}\mathbf{\Theta}^H. \tag{28}$$

The pre-whitening filter diagonalizes the symbol-clock noise covariance matrix at the output of the filter as

$$\begin{aligned} \mathbf{\Phi}_{\tilde{\mathbf{w}}_n\tilde{\mathbf{w}}_n} &= \sigma_n^2\mathbf{P}\mathbf{A}^H\mathbf{A}\mathbf{P}^H\\ &= \sigma_n^2\mathbf{R}^{-H}\mathbf{\Theta}^H\left(\mathbf{Q}\mathbf{R}\mathbf{\Theta}^H\right)^H\mathbf{Q}\mathbf{R}\mathbf{\Theta}^H\mathbf{\Theta}\mathbf{R}^{-1}\\ &= \sigma_n^2\mathbf{R}^{-H}\mathbf{\Theta}^H\mathbf{\Theta}\mathbf{R}^H\mathbf{Q}^H\mathbf{Q}\mathbf{R}\mathbf{\Theta}^H\mathbf{\Theta}\mathbf{R}^{-1}\\ &= \sigma_n^2\mathbf{I}_K\,. & (29)\end{aligned}$$

Moreover, the application of (28) to the system description in (26) shows that $\mathbf{P}$ even leads to a triangular system description

$$\begin{aligned}\mathbf{y}_w &= \mathbf{P}\mathbf{A}^H\mathbf{A}\mathbf{x} + \tilde{\mathbf{w}}_n \\ &= \mathbf{R}^{-H}\mathbf{\Theta}^H\left(\mathbf{Q}\mathbf{R}\mathbf{\Theta}^H\right)^H\mathbf{Q}\mathbf{R}\mathbf{\Theta}^H\mathbf{x} + \tilde{\mathbf{w}}_n \\ &= \mathbf{R}^{-H}\mathbf{\Theta}^H\mathbf{\Theta}\mathbf{R}^H\mathbf{Q}^H\mathbf{Q}\mathbf{R}\mathbf{\Theta}^H\mathbf{x} + \tilde{\mathbf{w}}_n \\ &= \mathbf{R}\mathbf{\Theta}^H\mathbf{x} + \tilde{\mathbf{w}}_n\,. \end{aligned} \tag{30}$$

Here, $\tilde{\mathbf{w}}_n$ is a vector containing white uncorrelated Gaussian noise samples with variance σ_n^2 and $\mathbf{R}$ is the sorted upper triangular system matrix ensuring optimized successive detection of the permuted source vector $\mathbf{\Theta}^H\mathbf{x}$. Additionally, the matrix inversion in (28) can efficiently be implemented since $\mathbf{R}$ is an upper triangular matrix which can be inverted easily. Another advantage here is that $\mathbf{QR}$ and $\mathbf{\Theta}$ only have to be updated when the underlying channel matrix $\mathbf{A}$ changes which is quite suitable for a hardware realization.

VI. Fixed-Point Operation

Hardware implementations require due to their limited resources the use of fixed-point (FXP) arithmetics. This generally saves area and energy and is thus usually faster compared to floating-point operations, though finite word length effects can lower the accuracy of the calculations. Especially for communications hardware stringent timing requirements apply, which makes the use of a FXP number representation necessary.

We examined the SA-SIC algorithm under finite word length constraints using the Matlab Fixed-Point Toolbox. The input data of the SA-SIC was quantized to the well-known Qn format where each number consists of a sign bit and n fractional bits [15]. Then, the set of representable numbers is a finite ring and is limited to the interval $[-1, 1)$ with a quantization step of 2^{-n}. This number format was chosen, because it is well suited for signal processing applications, since it offers the advantage that number overflows may occur during multiplication without affecting the end result.

Therefore, binary scaling has to be applied to fit the incoming data into the representable number range. The system model, e.g. as given in eq. (30) with the shorthand $\tilde{\mathbf{R}} = \mathbf{R}\mathbf{\Theta}^H$, can be multiplied on both sides with a scaling factor f as

$$f\mathbf{y} = f\left(\tilde{\mathbf{R}}\mathbf{x} + \mathbf{w}_n\right) = f\tilde{\mathbf{R}}\mathbf{x} + f\mathbf{w}_n\,. \tag{31}$$

Quantizing $f\mathbf{y}$, $f\tilde{\mathbf{R}}$ and $f\sigma_n$ leads to input data for the SA-SIC that can be successfully represented by the fractional Qn format. The appropriate sizing of the scaling factor f can be derived from the known statistics of the Gaussian noise, Rayleigh channel and modulation alphabet to avoid underflows and overflows. In practical implementations an automatic gain control would ensure the correct scaling.

Fig. 4 shows the measured input statistics of the pre-whitened y_k's. Unsurprisingly, the distribution resembles a Gauss curve, which is in fact a superposition of several normal distributions, according to

$$p_{Y_k}(y_k) = \sum_{\forall x \in \mathcal{A}_0} \Pr(x)\,\mathcal{N}(\mu = x, \sigma_n)\,, \tag{32}$$

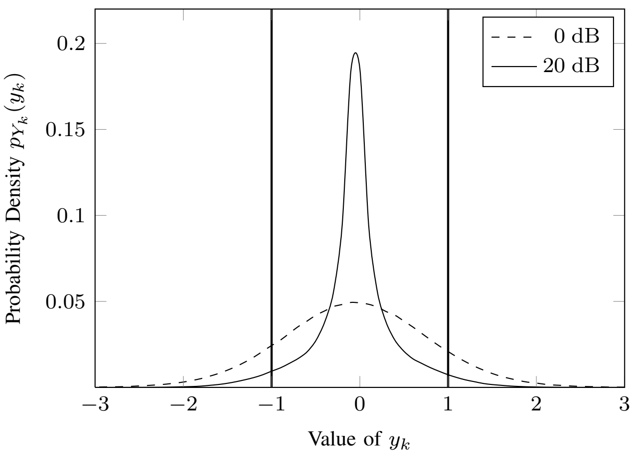

Fig. 4. Input statistics of the elements of the pre-whitened vector y for different E_b/N_0 in linear scale. The interval between the dash-dotted lines signifies the number range of the fractional Qn format. The lesser the channel quality is, the more probable higher magnitudes are, which leads to overflows.

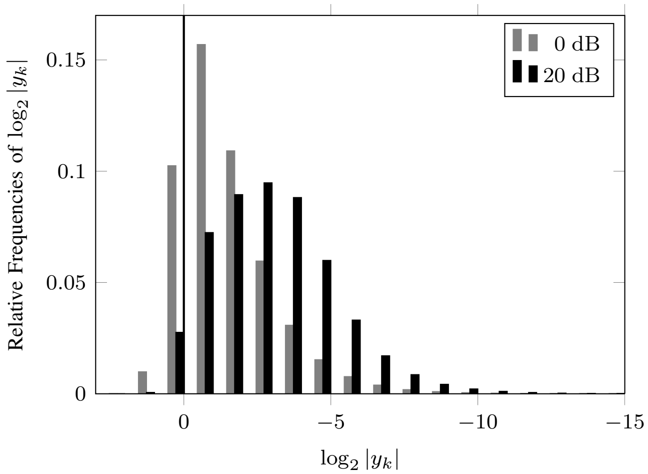

Fig. 5. Input statistics of magnitude of the elements of y for different E_b/N_0 in logarithmic scale with respect to the binary fixed-point representation. Values to the left of the dash-dotted line are outside the number range of the fractional Qn format. For higher SNR, the dynamic range becomes larger, which in turn makes greater bit widths necessary.

when we assume perfect Channel State Information (CSI). To fit y_k with high probability into a Qn representation, the data has to be scaled at least with $f = 1/2$ or better $f = 1/4$, which is algorithmically nothing else than an efficient bit shift to the right by one respectively two bit locations.

One problem is to fit the magnitudes of the input data into the Qn format, but another to consider the dynamic range to avoid number underflows. This means, significant signal changes have to be greater than the quantization error, which asks for a sufficient bit width n. Fig. 5 shows the input statistics for the (unscaled) magnitude of the elements of $\mathbf{y}$ over a $\log_2$ scale. Each bin of the histogram corresponds to one bit position of a binary number. On the left are the most significant bits (MSB) and on the right the least significant ones (LSB). All magnitudes to the right of the dash-dotted line are representable by the fractional Qn binary number format. After sufficient scaling, i.e. bit shifts to the right by at least one or two bit positions, the dynamic range of the input data fits

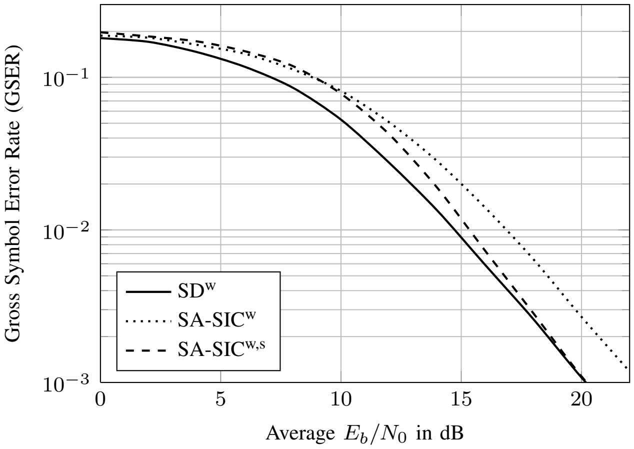

Fig. 6. Gross Symbol Error rate of SA-SIC detection with and without pre-whitening and sorted QR decomposition in comparison to S-MAP sphere decoding.

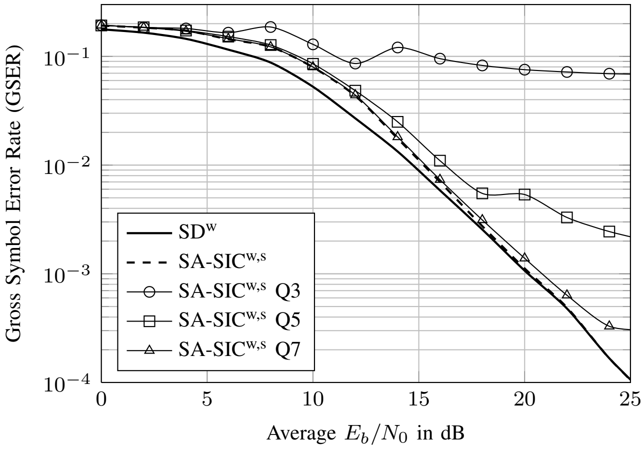

Fig. 7. Performance of fixed-point SIC detection, $K = 20$

with high probability into 16 bits width binary numbers, which enables almost lossless FXP arithmetics. We did not restrict ourselves only to Q15 numbers, but explored the influence of smaller bit widths as well. Especially the Q7 format is of interest, when SA-SIC detection shall be employed on small 8 bit DSP architectures. The results will be discussed in the next section.

As further FXP parameters of the simulation setup we used round to the nearest as rounding method, saturated arithmetics if over- or underflows should occur and we kept the most significant bits after products and sums due to the fractional format.

VII. Simulative Results

We simulated the in section V described system model for two scenarios. The first setup is an overdetermined system with $K = 20$ transmitting nodes or users and with a CDMA spreading sequence length of $N_s = 32$. The second setup is a fully loaded CDMA system with $K = N_s = 16$. We evaluated the gross symbol error rate (GSER) over the augmented symbol alphabet $\mathcal{A}_0$ and examined different combinations of pre-processing methods and quantization effects.

Both simulation scenarios share the remaining properties. The per-node activity probability is homogeneously $p_a = 0.2$. BPSK modulation is used and random Bernoulli sequences are employed as spreading codes. The channel is described by a normalized 4-tap Rayleigh fading impulse response.

A. Overdetermined System

Fig. 6 plots the GSER over the channel quality expressed in E_b/N_0. The relationship between E_b/N_0 and SNR is given (in linear scale) by

$$E_b/N_0 = \frac{\text{SNR}}{p_a \log_2(|\mathcal{A}|)} \,. \tag{33}$$

Optimal performance is achieved by a sparsity-aware sphere decoder with pre-whitening filter (SD$^{\text{w}}$). A performance with nearly constant degradation is achieved by the SA-SIC without sorting but with pre-whitening denoted as SA-SIC$^{\text{w}}$. SA-SIC with pre-whitening and sorting SA-SIC$^{\text{w,s}}$ according to section V-A converges with the performance of S-MAP detection at high SNR. Note that the application of sorted pre-processing makes no difference for the GSER result of the SD, since it always finds the S-MAP solution. In our simulations both algorithms operated on exactly the same (sorted) input data, which makes the runtime of the SD, n_{SD} as plotted in Fig. 2, comparable to the runtime of the SA-SIC, which is independently of the SNR $n_{\text{SA-SIC}} = K$.

A comparison of the detection loss between SD$^{\text{w}}$ and SA-SIC$^{\text{w,s}}$ with the difference in runtime as shown in Fig. 2 underlines that SA-SIC is a very efficient means for the detection of sparse multi-user signals. For high SNR values the performance of the SA-SIC even converges with the SD. The reason for this behaviour lies in the rather quick termination of the SD, which terminates then in average after almost the minimum number of processed nodes, which equates to SA-SIC detection.

Furthermore, we performed simulations for SA-SIC with FXP arithmetics in Q3, Q5, Q7 and larger formats. The GSER curves for Q11 and Q15 are excluded for the sake of clarity, since these bit widths are sufficient to accommodate the dynamic range of the input data, so that the performance is exactly identical to the floating-point simulation SA-SIC$^{\text{w,s}}$ for the range of the shown symbol error rates. Finite word length effects of Q5 and especially Q3 are too dominant to allow for successful symbol detection. But quantizing with 8 bits results in minor performance degradations at high SNR only, which allows for an 8 bit implementation on small DSP architectures. These degradations are due to overflows and underflows, which underlines the fact that the dynamic range of the data must fit into the chosen Qn format, as discussed earlier on Fig. 5 in the previous section.

B. Fully Loaded System

As mentioned above, the second simulation scenario assumed a fully-loaded CDMA system where the number of

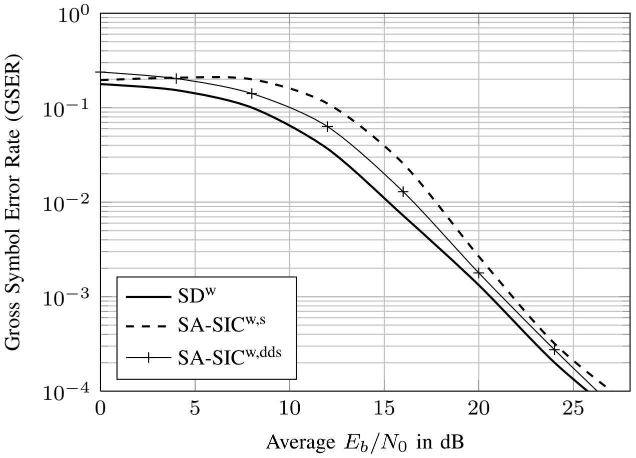

Fig. 8. Gross Symbol Error rate of SA-SIC with data dependent sorting compared to SNR based sorting. S-MAP curve shown for comparison

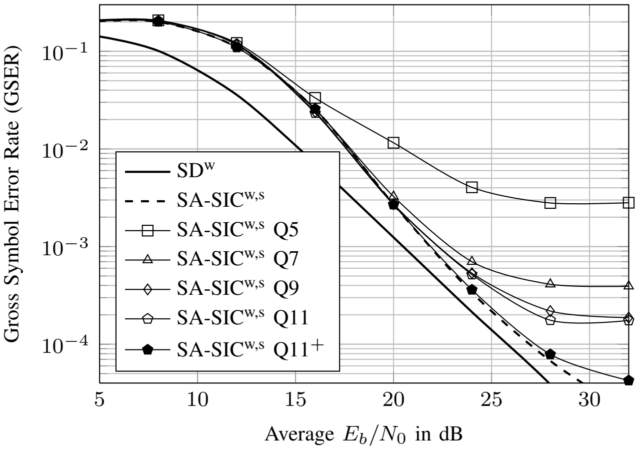

Fig. 9. Performance of fixed-point SA-SIC detection, $K = 16$

users equals the number of available spreading codes. Fig. 8 shows the detection performance of SA-SIC, which is worse than for the overdetermined system. The plot additionally features a SA-SIC with data-dependent pre-sorting as we introduced it in section IV-B, here denoted as SA-SIC$^{\mathrm{w,dds}}$. It performs exceptionally well for this loaded case, thus only shown in this plot, and closes the gap between optimal S-MAP detection and suboptimal SA-SIC detection further. We can therefore conclude that the newly introduced data-depending pre-sorting strategy correctly tunes the S-MAP factor vector $\boldsymbol{\lambda}$ and reduces therewith the probability of SA-SIC error propagation.

In terms of algorithm runtime efficiency, the SA-SIC fares even better in this loaded case. As Fig. 2 shows, the SD has to search longer in this $K = 16$ scenario, whereby the iteration count of the SA-SIC stays naturally constant.

Fig. 9 displays the GSER curves with regard to FXP arithmetics. For high SNR values the FXP SA-SIC runs into an error floor. Notably, the GSER does not really improve for larger bit widths than Q9 at the same scaling factor $1/2$. SA-SIC$^{\mathrm{w,s}}$ Q11$^+$ denotes an additional simulation with a scaling factor of $1/4$ and clearly improved GSER, which emphasizes the necessity of a sufficiently dimensioned scaling as long as the dynamic range of the data allows for it. The error floor of all FXP simulations, hence, must be caused mainly by number range overflows. The plot also shows that 8 bit quantization is a feasible low-complexity choice, with GSER better than 10^{-3} for high SNR, which could be further improved with channel coding techniques.

VIII. Conclusion

We have shown that the Sparsity Aware Successive Interference Cancellation (SA-SIC) can nearly approach the performance of a S-MAP detector with drastically decreased complexity. This gain is made possible by the employment of pre-processing techniques which are in parts inspired by the field of Compressed Sensing. Aiming at practical hardware realizations, we have shown that 8 bit quantization can already be sufficient to preserve the efficiency boost the algorithm introduced, and 16 bit quantization performs ideally. Therefore, this paper substantiates the claim that Sparsity-Aware SIC detection is an excellent alternative to optimal S-MAP detection for sparse multi-user signals.

References

[1] A. Bartoli, M. Dohler, J. Hernndez-Serrano, A. Kountouris, and D. Barthel, "Low-Power Low-Rate goes Long-Range: the case for secure and cooperative Machine-to-Machine communications," in *NETWORKING 2011 Workshops*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, vol. 6827, pp. 219–230.

[2] S. Lien, K. Chen, and Y. Lin, "Toward ubiquitous massive accesses in 3GPP machine-to-machine communications," *Communications Magazine, IEEE*, vol. 49, no. 4, pp. 66 –74, Apr. 2011.

[3] H. Zhu and G. Giannakis, "Exploiting sparse user activity in multiuser detection," *IEEE Transactions on Communications*, vol. 59, no. 2, pp. 454 – 465, Feb. 2011.

[4] F. Monsees, C. Bockelmann, D. Wübben, and A. Dekorsy, "Sparsity aware multiuser detection for machine to machine communication," in *Second International Workshop on Machine-to-Machine Communication at IEEE Globecom 2012*, Anaheim, USA, Dec 2012.

[5] ——, "Compressed sensing bayes risk minimization for underdetermined systems via sphere detection," in *IEEE 77th Vehicular Technology Conference (VTC2013-Spring)*, Dresden, Germany, Jun 2013.

[6] B. Knoop, T. Wiegand, and S. Paul, "Low-Complexity and Approximative Sphere Decoding of Sparse Signals," in *Asolimar Conference on Signals, Systems and Computers*, Pacific Grove USA, Nov 2012.

[7] D. Wübben, R. Böhnke, J. Rinas, K.-D. Kammeyer, and V. Kühn, "Efficient algorithm for decoding layered space-time codes," *IEE Electronic Letters*, vol. 37, no. 22, pp. 1348–1350, Nov 2001.

[8] Y. C. Eldar and G. Kutyniok, *Compressed Sensing: Theory and Applications*. Cambridge, U.K.: Cambridge Univ. Press, May 2012.

[9] M. O. Damen, H. E. Gamal, and G. Caire, "On maximum-likelihood detection and the search for the closest lattice point," *IEEE Transactions on Information Theory*, vol. 49, no. 10, pp. 2389 – 2402, oct. 2003.

[10] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 2nd ed. The Johns Hopkins University Press, 1989.

[11] J. Tropp and A. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Transactions on Information Theory*, vol. 53, no. 12, pp. 4655 –4666, Dec. 2007.

[12] T. Hastie, R. Tibshirani, and J. H. Friedman, *The Elements of Statistical Learning*, 2nd ed. Springer, Jul. 2011.

[13] D. Wübben, "Effiziente Detektionsverfahren für Multilayer-MIMO-Systeme," Ph.D. dissertation, University of Bremen, 2006.

[14] D. E. Knuth, *The Art of Computer Programming Vol. 3: Sorting and Searching*, 2nd ed. Addison-Wesley, 2002.

[15] S. A. Khan, *Digital design of signal processing systems: a practical approach*. Chichester: Wiley, 2011.