14th European Signal Processing Conference (EUSIPCO 2006), Florence, Italy, September 4-8, 2006, copyright by EURASIP

EFFICIENT VECTOR PERTURBATION IN MULTI-ANTENNA MULTI-USER SYSTEMS BASED ON APPROXIMATE INTEGER RELATIONS

Dominik Seethaler and Gerald Matz

Institute of Communications and Radio-Frequency Engineering, Vienna University of Technology
Gusshausstraße 25/389, A-1040 Vienna, Austria
Phone: +43 1 58801 38958, Fax: +43 1 58801 38999, E-mail: dominik.seethaler@tuwien.ac.at

ABSTRACT

Approximate vector perturbation techniques assisted by LLL lattice reduction (LR) can exploit all the diversity that is available in multi-user multi-antenna broadcast systems. However, the required computational complexity of LLL-LR can be quite large. In this paper, we propose a much simpler and much more efficient LR algorithm than LLL. This LR technique is based on Brun's algorithm for finding approximate *integer relations (IRs)*. The link between LR and IRs is established by considering poorly conditioned channels with a single small singular value. Simulation results show that our scheme can achieve large (but bot full) diversity at a fraction of the complexity required for LLL-assisted vector perturbation.

1. INTRODUCTION

Precoding based on vector perturbation is a very promising technique for wireless broadcast scenarios, where a transmitter uses multiple antennas to serve multiple non-cooperating users [1–3]. With vector perturbation, the data (to be transmitted via pre-equalization) is perturbed such that the transmit power is minimized. The optimum perturbation vector can be found via sphere-encoding [2]. In general, this requires exponential complexity (in the worst case and also on average [4]). As a consequence, several efficient approximate vector perturbation techniques have been developed: [2, 3] discussed Tomlinson-Harashima precoding and [5] considered lattice reduction (LR) techniques (using the LLL algorithm [6]) as a preprocessing stage to vector perturbation. Recently, it was shown that approximate vector perturbation preceded by LLL-LR can achieve full diversity [7]. While the LLL algorithm has just polynomial complexity [6, 8] in the number of users, it can still be computationally intensive if the number of users is large and the channel realization is poorly conditioned.

In this paper, we aim at reducing the complexity of the LR preprocessing stage that depends solely on the channel realization (i.e., not on the transmit data) and can often be decisive (e.g. in OFDM-based transmission schemes). To this end, we propose a novel LR algorithm that is much simpler and significantly more efficient than the LLL algorithm. This novel LR technique is then used as preprocessing for approximate vector perturbation techniques. Our LR scheme is based on an algorithm proposed by Brun in 1919 (see [8] and references therein) for finding approximate *integer relations* (IRs) [8, 9]. The link between approximate IRs and LR for vector perturbation is established by considering poorly conditioned channel realizations that have just a single small singular value. We hence refer to our LR technique as IR based LR (IR-LR).

In general, the computational complexity of IR-LR is just a fraction of LLL-LR since it does not require a QR-decomposition of the channel matrix and repeated Givens rotations and size reduction steps (cf. [8, 10]). Although IR-LR incurs a performance penalty compared to LLL-LR, simulation results will reveal that the proposed IR-LR based approximate vector perturbation techniques can still achieve a large part of the available diversity.

This work was supported by the European Union project MASCOT within the Sixth Framework Program under contract IST-26905.

The rest of the paper is organized as follows. In the remainder of this section, we discuss the system model and the basic principle of (LR-assisted) vector perturbation. In Section 2, we present the basic idea of IR-LR. Brun's algorithm and the resulting IR-LR technique is discussed in detail in Section 3. Finally, simulation results are presented in Section 4. Section 5 concludes the paper.

1.1 System Model

We consider a multi-user communications system operating in the downlink (see e.g. [1, 3]). The base station is equipped with M transmit antennas and there are $K \leq M$ users, each with a single receive antenna. We emphasize that in general the user antennas are not co-located and hence cooperation among users is not possible.

Let $\mathbf{x} \triangleq (x_1 \dots x_M)^T$ denote the transmit vector, normalized such that $\mathrm{E}\{\|\mathbf{x}\|^2\} = 1$. Furthermore, collect the values r_k, $k = 1, \dots, K$, received by the K users in a receive vector $\mathbf{r} \triangleq (r_1 \dots r_K)^T$. Under the assumption of a flat-fading channel[1] the mapping from $\mathbf{x}$ to $\mathbf{r}$ is given by

$$\mathbf{r} = \mathbf{H}\mathbf{x} + \mathbf{w}. \tag{1}$$

Here, $\mathbf{H}$ is the $K \times M$ channel matrix, whose elements $h_{k,m} = [\mathbf{H}]_{k,m}$ are the complex fading coefficients between the mth transmit antenna and the kth user. Furthermore, $\mathbf{w} \triangleq (w_1 \dots w_K)^T \sim \mathscr{CN}(\mathbf{0}, \sigma_{\mathbf{w}}^2 \mathbf{I})$ denotes spatially white complex Gaussian noise.

1.2 Precoding and Vector Perturbation

At each time instant, the base station intends to transmit K complex-valued data symbols d_k, $k = 1, \dots, K$, which are picked from a symbol alphabet $\mathscr{A}$. The kth data symbol d_k is intended for user k. With perfect channel state information at the transmitter, interference free transmission to each user is achieved by using zero-forcing precoding/pre-equalization (e.g. [1]). Here, the data vector $\mathbf{d} = (d_1 \dots d_K)^T$ is pre-multiplied with an $M \times K$ precoding matrix $\mathbf{P}$ which equals the pseudo-inverse[2] of the channel:

$$\mathbf{s} = \mathbf{P}\mathbf{d}, \qquad \text{with } \mathbf{P} = \mathbf{H}^H(\mathbf{H}\mathbf{H}^H)^{-1}. \tag{2}$$

The transmit vector is then obtained by a normalization of $\mathbf{s}$, i.e.

$$\mathbf{x} = \frac{\mathbf{s}}{\sqrt{\gamma}}, \qquad \text{with } \gamma \triangleq \|\mathbf{s}\|^2. \tag{3}$$

Inserting (3) and (2) into (1) yields $\mathbf{r} = \frac{1}{\sqrt{\gamma}}\mathbf{d} + \mathbf{w}$, i.e., the received value r_k of the kth user reads

$$r_k = \frac{1}{\sqrt{\gamma}} d_k + w_k. \tag{4}$$

[1] The flat-fading assumption is no serious restriction since any frequency-selective channel can be converted into parallel flat-fading channels via the use of OFDM.

[2] For simplicity, throughout the paper we do not consider any MMSE-based precoding (e.g. [1]), although in general the performance is improved compared to ZF-based precoding.

Hence, each user sees a scaled and noise-corrupted version of his corresponding transmit symbol. Optimum detection can thus be achieved in a non-cooperative fashion by quantizing $\sqrt{\gamma}\, r_k$ with respect to the symbol alphabet $\mathscr{A}$ (denoted as $Q\{\cdot\}$):

$$\hat{d}_k = Q\big\{\sqrt{\gamma}\, r_k\big\}. \tag{5}$$

According to [2], the scaling can be performed with $\sqrt{\mathrm{E}\{\gamma\}}$ instead of $\sqrt{\gamma}$ without significant performance loss.

The major problem of plain ZF precoding is the power enhancement in the unnormalized transmit signal $\mathbf{s}$ in (2) [1, 2]. As can be seen from (4), the receive SNR is inversely proportional to γ, i.e., if γ is very large the SNR is small and hence overall system performance (error probability) degrades significantly.

This performance degradation can be combated very effectively by using the vector perturbation (VP) technique proposed in [2]. Here, the vector $\mathbf{s}$ in (2), (3) is replaced with

$$\mathbf{s}(\mathbf{z}) = \mathbf{P}(\mathbf{d} + \tau\mathbf{z}),$$

where $\mathbf{z} \in \mathbb{C}\mathbb{Z}^K$ is a perturbation vector whose elements are complex integers[3] and τ is an appropriately chosen but fixed real-valued scaling factor. For example, for QAM symbol alphabets the constant τ is usually chosen such that the extended symbol alphabet is again a rectangular lattice (e.g., $\tau = 4$ for 4-QAM $\mathscr{A} = \{1+j, 1-j, -1+j, -1-j\}$). Equivalently to (4) one obtains

$$r_k = \frac{1}{\sqrt{\gamma}}(d_k + \tau z_k) + w_k,$$

which amounts to having an extended symbol alphabet $\mathscr{A} + \tau\mathbb{C}\mathbb{Z}$ (i.e. all τ-scaled complex-valued integer translates of $\mathscr{A}$). Each user can then perform detection by applying a complex-valued modulo operation $M_\tau\{\cdot\}$ to its $\sqrt{\gamma}$-scaled received value,

$$M_\tau\{\sqrt{\gamma} r_k\} \triangleq \big(\mathrm{Re}\{\sqrt{\gamma} r_k\} \mod \tau\big) + j\big(\mathrm{Im}\{\sqrt{\gamma} r_k\} \mod \tau\big),$$

followed by quantization:

$$\hat{d}_k = Q\big\{M_\tau\{\sqrt{\gamma}\, r_k\}\big\}.$$

It is natural to choose the perturbation vector $\mathbf{z}$ such that the receive SNR is maximum. This is achieved by minimizing the scaling factor $\gamma = \|\mathbf{s}(\mathbf{z})\|^2$:

$$\mathbf{z}_{\mathrm{opt}} = \arg\min_{\mathbf{z}\in\mathbb{C}\mathbb{Z}^K} \|\mathbf{s}(\mathbf{z})\|^2 = \arg\min_{\mathbf{z}\in\mathbb{C}\mathbb{Z}^K} \big\|\mathbf{P}(\mathbf{d}+\tau\mathbf{z})\big\|^2. \tag{6}$$

The actual transmit vector is then obtained according to (3) with $\mathbf{s} = \mathbf{s}(\mathbf{z}_{\mathrm{opt}})$. The minimization (6) is an integer least-squares problem whose complexity in general is exponential in the number of users K and thus becomes prohibitively complex even for moderate K. A promising algorithm to solve (6) is the sphere decoding (SD) algorithm [2, 11] (in [2] this is then referred to as *sphere encoding*), which however is still exponentially complex [4] (in the worst case and also on average).

Various precoding techniques can be interpreted as approximations of (6). This includes plain ZF precoding without perturbation ($\mathbf{z} = \mathbf{0}$), Tomlinson-Harashima precoding [3], and LR-assisted vector perturbation [5].

1.3 LR-Assisted Vector Perturbation

For later reference, we briefly review the basic concepts of LR-assisted approximate vector perturbation [5] (see also Babai's approximation [9] and various data detection algorithms [10, 12, 13]).

[3]The set $\mathbb{C}\mathbb{Z}$ of complex integers comprises all complex numbers with integer real and imaginary parts. For brevity, we refer to vectors with complex-valued integer entries as "integer vectors."

For the LR preprocessing, the columns of the precoding matrix $\mathbf{P}$ are viewed as a *basis* for a K-dimensional lattice $\mathscr{L}$ in $\mathbb{C}^M$,

$$\mathscr{L} \triangleq \Big\{\mathbf{P}\mathbf{z}:\ \mathbf{z}\in\mathbb{C}\mathbb{Z}^K\Big\}.$$

(We remain in the complex-valued domain although LR is usually discussed for an equivalent real-valued $2K$-dimensional lattice in $\mathbb{R}^{2M}$ [5, 10].) The volume of a fundamental cell of the lattice $\mathscr{L}$ is defined as $|\mathscr{L}| = \det(\mathbf{P}^H\mathbf{P})$ and is independent of the lattice basis. The goal of LR is to transform the lattice basis $\mathbf{P}$ into a "better" basis $\widetilde{\mathbf{P}}$ for the *same* lattice $\mathscr{L}$. The relation between the original and the reduced bases is given by $\widetilde{\mathbf{P}} = \mathbf{P}\mathbf{B}$, where $\mathbf{B} = [\mathbf{b}_1 \ldots \mathbf{b}_K]$ is a $K\times K$ unimodular matrix (i.e., it has complex-valued integer entries and $\det(\mathbf{B}) = \pm 1$). We note that LR can equivalently be performed on the rows of the channel matrix $\mathbf{H}$ since the resulting reduced channel matrix $\widetilde{\mathbf{H}} = \mathbf{B}^{-1}\mathbf{H}$ entails $\widetilde{\mathbf{P}} = \mathbf{P}\mathbf{B} = \widetilde{\mathbf{H}}^H(\widetilde{\mathbf{H}}\widetilde{\mathbf{H}}^H)^{-1}$.

The *orthogonality defect* of the basis $\widetilde{\mathbf{P}}$ is defined as

$$\delta(\widetilde{\mathbf{P}}) \triangleq \frac{1}{|\mathscr{L}|}\prod_{k=1}^{K}\|\tilde{\mathbf{p}}_k\|^2, \tag{7}$$

where $\tilde{\mathbf{p}}_k = \mathbf{P}\mathbf{b}_k$ denotes the kth column of $\widetilde{\mathbf{P}}$. Since LR aims at finding *short* lattice basis vectors $\tilde{\mathbf{p}}_k$, the orthogonality defect $\delta(\widetilde{\mathbf{P}})$ will be smaller than the orthogonality defect $\delta(\mathbf{P})$ of the original basis $\mathbf{P}$. Hence, $\widetilde{\mathbf{P}}$ is a better basis in that it is more orthogonal than $\mathbf{P}$. This implies that any conventional (approximate) vector perturbation algorithm operating on the transformed basis $\widetilde{\mathbf{P}}$ will in general lead either to better performance results or to a reduced complexity as compared to operating on the original basis $\mathbf{P}$. In particular, the cost function in (6) can equivalently be written as

$$\begin{aligned}\|\mathbf{P}(\mathbf{d}+\tau\mathbf{z})\|^2 &= \|\mathbf{P}\mathbf{B}\mathbf{B}^{-1}(\mathbf{d}+\tau\mathbf{z})\|^2 \\ &= \|\widetilde{\mathbf{P}}(\mathbf{B}^{-1}\mathbf{d}+\tau\mathbf{B}^{-1}\mathbf{z})\|^2 = \|\widetilde{\mathbf{P}}(\tilde{\mathbf{d}}+\tau\tilde{\mathbf{z}})\|^2,\end{aligned}$$

where

$$\tilde{\mathbf{d}} = \mathbf{B}^{-1}\mathbf{d}, \qquad \tilde{\mathbf{z}} = \mathbf{B}^{-1}\mathbf{z}. \tag{8}$$

Due to the unimodularity of $\mathbf{B}$, $\tilde{\mathbf{z}} \in \mathbb{C}\mathbb{Z}^K$. Hence, solving (6) is equivalent to first solving

$$\tilde{\mathbf{z}}_{\mathrm{opt}} = \arg\min_{\tilde{\mathbf{z}}\in\mathbb{C}\mathbb{Z}^K}\Big\{\|\widetilde{\mathbf{P}}(\tilde{\mathbf{d}}+\tau\tilde{\mathbf{z}})\|^2\Big\}, \tag{9}$$

and then calculating

$$\mathbf{z}_{\mathrm{opt}} = \mathbf{B}\tilde{\mathbf{z}}_{\mathrm{opt}}. \tag{10}$$

In general, (9) can be solved much more efficiently than (6) since $\widetilde{\mathbf{P}}$ is more orthogonal than $\mathbf{P}$.

Similarly, one can apply any approximation of optimum vector perturbation (ZF, THP, etc.) to (9) and then obtain an approximate solution of (6) via the right-hand side of (8). This results in better performance than that obtained by applying an approximate vector perturbation algorithm directly to the original problem (8). In fact any approximate vector perturbation approach applied after LLL-LR-preprocessing can indeed achieve full diversity (this is due to certain properties of an LLL reduced basis) [7]. We note that the LLL algorithm has a computational complexity that is polynomial in K [6, 9].

As an example, LR-assisted ZF precoding is performed by replacing the lattice basis $\widetilde{\mathbf{P}}$ in (9) with the identity matrix[4]:

$$\tilde{\mathbf{z}}_{\mathrm{ZF,LR}} = \arg\min_{\tilde{\mathbf{z}}\in\mathbb{C}\mathbb{Z}^K}\|\tilde{\mathbf{d}}+\tau\tilde{\mathbf{z}}\|^2 = \left\lfloor -\frac{1}{\tau}\tilde{\mathbf{d}}\right\rceil$$

[4]Here, $\lfloor\cdot\rceil$ denotes component-wise rounding to complex-valued integers.

and thus

$$\mathbf{z}_{\mathrm{ZF,LR}} = \mathbf{B}\left\lfloor -\frac{1}{\tau}\tilde{\mathbf{d}}\right\rceil. \tag{11}$$

Note that without LR preprocessing, replacing $\mathbf{P}$ in (6) with $\mathbf{I}$ results in $\hat{\mathbf{z}}_{\mathrm{ZF}} = \mathbf{0}$ since all elements of $\mathbf{d}/\tau$ have real and imaginary parts less than 1/2. An equivalent approach can be used for LR-assisted THP [5].

2. BASIC IDEA OF IR-LR

In this section, we present the basic idea behind using integer relations (IR) for the LR-preprocessing stage in vector perturbation. The proposed IR-LR algorithm (described in full detail in Section 3) is much simpler and computationally more efficient than LLL-LR.

In what follows, we will use the singular value decomposition (SVD) $\mathbf{H} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H$ of the channel matrix $\mathbf{H}$ [14]. Here, the diagonal matrix $\mathbf{\Sigma}$ contains the singular values σ_k, $k = 1,\ldots,K$, and the columns of the unitary matrices $\mathbf{U} = [\mathbf{u}_1 \ldots \mathbf{u}_K]$ and $\mathbf{V}$ are the left and right singular vectors, respectively. We assume that the σ_k are sorted in nonincreasing order. The *condition number* of the channel is given by $c_{\mathbf{H}} = \frac{\sigma_1}{\sigma_K} \geq 1$. Channels that are poorly conditioned (i.e., $c_{\mathbf{H}}$ is large) are considered bad.

2.1 LR for Bad Channels

As discussed in Section 1.3, LR aims at finding a unimodular matrix $\mathbf{B}$ such that the lengths of the transformed lattice basis vectors are small. For the original basis $\mathbf{P} = [\mathbf{p}_1 \ldots \mathbf{p}_K]$ we have

$$\begin{aligned}\|\mathbf{p}_k\|^2 &= \left\|\mathbf{H}^H(\mathbf{H}\mathbf{H}^H)^{-1}\mathbf{e}_k\right\|^2 = \mathbf{e}_k^H(\mathbf{H}\mathbf{H}^H)^{-1}\mathbf{e}_k \\ &= \mathbf{e}_k^H\mathbf{U}\mathbf{\Sigma}^{-2}\mathbf{U}^H\mathbf{e}_k = \sum_{j=1}^{K}\frac{1}{\sigma_j^2}|\mathbf{u}_j^H\mathbf{e}_k|^2. \end{aligned}\tag{12}$$

Here, $\mathbf{e}_k$ denotes the kth unit vector. Obviously, $\|\mathbf{p}_k\|$ will be dominated by the terms in (12) corresponding to *small* singular values. If the channel is poorly conditioned, one or more singular values are very small and $\|\mathbf{p}_k\|$ will be very large for some k. Thus, the corresponding orthogonality defect $\delta(\mathbf{P})$ (cf. (7)) will also be large and conventional approximate vector perturbation techniques operating on $\mathbf{P}$ will perform poorly (see [15] for the impact of poorly conditioned channel realizations on the performance of data detection).

Similar to (12), for a reduced basis $\widetilde{\mathbf{P}} = [\tilde{\mathbf{p}}_1 \ldots \tilde{\mathbf{p}}_K]$ with $\tilde{\mathbf{p}}_k = \mathbf{P}\mathbf{b}_k$ we obtain

$$\|\tilde{\mathbf{p}}_k\|^2 = \left\|\mathbf{H}^H(\mathbf{H}\mathbf{H}^H)^{-1}\mathbf{b}_k\right\|^2 = \sum_{j=1}^{K}\frac{1}{\sigma_j^2}|\mathbf{u}_j^H\mathbf{b}_k|^2.$$

Obviously, small $\|\tilde{\mathbf{p}}_k\|^2$ (and thus also small $\delta(\widetilde{\mathbf{P}})$) requires that the integer vectors $\mathbf{b}_k$ be as collinear as possible to left singular vectors $\mathbf{u}_j$ for which σ_j is large. This is in fact achieved by LLL-LR and reduces the degrading influence of bad channels on approximate vector perturbation techniques dramatically. If the channel is well-behaved (close to orthogonal, small $c_{\mathbf{H}}$) all approximate vector perturbation schemes will perform well even without LR. Thus, LR has to take particular care of poorly conditioned channels.

As verified and discussed in [1, 15], iid Gaussian channels with large condition number usually have just a *single* small singular value, which is primarily responsible for the occurrence of long basis vectors in $\mathbf{P}$. Motivated by this fact, we propose to perform LR especially taylored to poorly conditioned channels with a single small singular value. In view of (12), our task is thus to find a unimodular transformation matrix $\mathbf{B}$ with integer columns $\mathbf{b}_k$ that are sufficiently orthogonal to $\mathbf{u}_K$.

2.2 IRs and Diophantine Approximations

In number theory, finding integer vectors $\mathbf{b}_k$ that are (almost) orthogonal to a given vector $\mathbf{u}_K$ is known as the (approximate) IR problem [8, 9]. In general, smaller $|\mathbf{u}_K^H\mathbf{b}_k|$ requires longer vectors $\mathbf{b}_k$. Hence, when searching for approximate IRs one aims at achieving small $|\mathbf{u}_K^H\mathbf{b}_k|$ using vectors $\mathbf{b}_k$ that are as short as possible.

In our setting, the channel determines which of the two targets (small $|\mathbf{u}_K^H\mathbf{b}_k|$ and small $\|\mathbf{b}_k\|$) is more important. In particular, the smaller σ_K is as compared to the other singular values, the more important it is to achieve small values of $|\mathbf{u}_K^H\mathbf{b}_k|$. The resulting increase in the lengths $\|\mathbf{b}_k\|$ leads to an increase of $|\mathbf{u}_j^H\mathbf{b}_k|$, $j < K$, (cf. (12)). However, for bad channels $1/\sigma_j \ll 1/\sigma_K$, $j < K$, and hence this increase will have negligible influence on $\|\tilde{\mathbf{p}}_k\|^2$ as compared to the decrease of $|\mathbf{u}_K^H\mathbf{b}_k|$.

The dual problem to approximate IRs is known as *simultaneous Diophantine approximation* [8, 9]. Here, the goal is finding short integer vectors that are sufficiently collinear with a given straight line (e.g. the straight line given by $\alpha\,\mathbf{u}_K$, $\alpha \in \mathbb{C}$). Good simultaneous Diophantine approximations and/or good approximate IRs can be found using basically the same algorithms. In fact, the original applications of the LLL algorithm were finding simultaneous Diophantine approximations and approximate IRs [6]. Several advanced algorithms for the IR problem have been developed recently, including the so-called PSLQ algorithm [16] and the LLL-based HSLJ algorithm, e.g. [8]. These algorithms (including LLL) are very attractive in that they offer guarantees regarding execution time and approximation quality. However, they are computationally quite intensive, which is in contrast to our goal of developing low-cost LR techniques. Thus, we focus on a extremely simple and very efficient algorithm proposed by Brun (see [8] for a discussion of Brun's algorithm in the context of simultaneous Diophantine approximations).

3. IR-LR: THE ALGORITHM

Before discussing IR-LR in more detail, we state Brun's algorithm (adapted to the complex-valued case) for finding good approximate IRs for $\mathbf{u}_K$ (see [8] and references therein).

3.1 Brun's Algorithm

Set $\mathbf{B} = \mathbf{I}$ and $\boldsymbol{\mu} = \mathbf{u}_K$, and repeat the following steps until a suitable termination condition (see Section 3.2) is satisfied:

1. Find the indices s and t of the two largest $|\mu_k|$, $k = 1,\ldots,K$:

$$s = \arg\max_{k\in\{1,\ldots,K\}}|\mu_k|, \qquad t = \arg\max_{k\in\{1,\ldots,K\}/\{s\}}|\mu_k|.$$

2. Calculate

$$\beta = \left\lfloor\frac{\mu_s}{\mu_t}\right\rceil. \tag{13}$$

3. Perform the updates

$$\mathbf{B} = \left[\mathbf{b}_1 \ldots \mathbf{b}_{s-1}\ \mathbf{b}_s'\ \mathbf{b}_{s+1} \ldots \mathbf{b}_K\right], \tag{14}$$

$$\boldsymbol{\mu} = (\mu_1 \ldots \mu_{s-1}\ \mu_s'\ \mu_{s+1} \ldots \mu_K)^T,$$

with

$$\mu_s' = \mu_s - \beta\,\mu_t, \qquad \mathbf{b}_s' = \mathbf{b}_s - \beta^*\,\mathbf{b}_t. \tag{15}$$

Steps 1–3 correspond to one iteration of Brun's algorithm. It can be easily shown via induction that each iteration preserves the property $\mu_k = \mathbf{b}_k^H\mathbf{u}_K$ (equivalently, $\boldsymbol{\mu} = \mathbf{B}^H\mathbf{u}_K$). We next demonstrate that Brun's algorithm results in a monotonic decrease of

$$|\mu_s| = |\mathbf{u}_K^H\mathbf{b}_s| = \max_{k\in\{1,\ldots,K\}}|\mathbf{u}_K^H\mathbf{b}_k|.$$

To this end, we rewrite (13) as

$$\beta = \left\lfloor\frac{\mu_s}{\mu_t}\right\rceil = \frac{\mu_s}{\mu_t} + \Delta, \qquad \text{with } |\Delta| \leq \frac{1}{\sqrt{2}}.$$

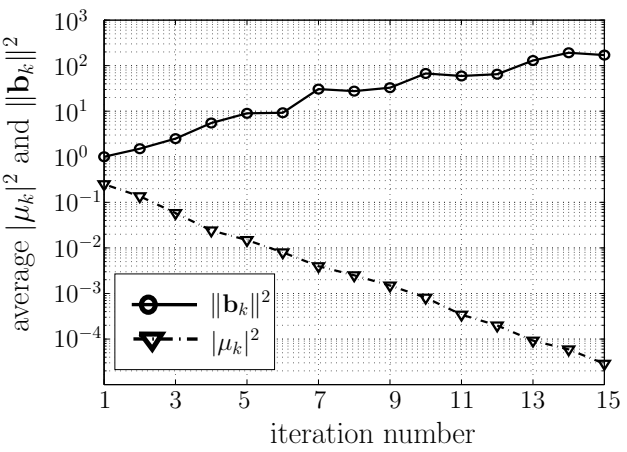

Figure 1: *Behaviour of Brun's algorithm: average* $|\mu_k|^2$ *and* $\|\mathbf{b}_k\|^2$ *versus the number of iterations.*

Inserting this in the μ-update in (15) results in

$$|\mu_s'| = |\mu_s - \beta\,\mu_t| = \left|\mu_s - \left(\frac{\mu_s}{\mu_t} + \Delta\right)\mu_t\right| = |\Delta\,\mu_t| \leq \frac{1}{\sqrt{2}}|\mu_t|.$$

Since by definition $|\mu_t| \leq |\mu_s|$, we finally obtain

$$|\mu_s'| \leq \frac{1}{\sqrt{2}}|\mu_s|.$$

Thus, at every iteration the largest component of $\mathbf{b}_k$, $k = 1,\ldots,K$, in the direction of $\mathbf{u}_K$ is decreased by at least $1/\sqrt{2}$. Hence, the inner products $|\mathbf{u}_K^H\mathbf{b}_k|$, $k = 1,\ldots,K$, are made *arbitrarily* small. On the other hand, a decrease in $|\mu_s|$ will in general result in an increase of the length $\|\mathbf{b}_s\|$ of the corresponding integer vector. This is illustrated in Fig. 1, which shows the average[5] of $|\mu_k|^2$ and $\|\mathbf{b}_k\|^2$ versus the number of iterations in Brun's algorithm for the case $K = 4$.

3.2 LR via Brun's Algorithm

We next show how Brun's algorithm can be used to perform efficient LR for the lattice $\mathscr{L}(\mathbf{P})$ generated by the precoding matrix $\mathbf{P}$.

The matrix $\mathbf{B}$ in Brun's algorithm can be shown to constitute a basis for $\mathbb{C}\mathbb{Z}^K$ in each iteration, which implies that it is unimodular. Thus, $\widetilde{\mathbf{P}} = \mathbf{P}\mathbf{B}$ is a basis for $\mathscr{L}(\mathbf{P})$. Since our goal is to find a reduced basis $\widetilde{\mathbf{P}}$ for $\mathscr{L}(\mathbf{P})$, i.e., the length of the lattice basis vectors $\tilde{\mathbf{p}}_k = \mathbf{P}\mathbf{b}_k$, $k = 1,\ldots,K$, or equivalently the orthogonality defect of $\widetilde{\mathbf{P}}$ (cf. (7)) should be as small as possible, we have to terminate Brun's algorithm properly. We thus propose to repeat Brun's iteration (Steps 1-3) as long as the orthogonality defect $\delta(\widetilde{\mathbf{P}})$ of $\widetilde{\mathbf{P}} = \mathbf{P}\mathbf{B}$ is decreasing. Equivalently, the algorithm will be terminated if

$$\|\mathbf{P}\mathbf{b}_s'\| > \|\mathbf{P}\mathbf{b}_s\|. \tag{16}$$

After termination, the transformation matrix $\mathbf{B}$ is used to perform LR-assisted approximate vector perturbation (cf. Section 1.3).

3.3 Implementation Aspects

To perform IR-LR via Brun's algorithm, we need to determine the left singular vector $\mathbf{u}_K$ of $\mathbf{H}$ corresponding to the smallest singular value. Note that $\mathbf{u}_K$ is also the eigenvector of the inverse Gram-matrix $(\mathbf{H}\mathbf{H}^H)^{-1}$ associated to its largest eigenvalue. For the case in which we are interested (poorly-conditioned channels with a single small singular), $(\mathbf{H}\mathbf{H}^H)^{-1}$ will have a single very large eigenvalue and can thus be approximated by a rank-one matrix, $(\mathbf{H}\mathbf{H}^H)^{-1} \approx \frac{1}{\sigma_K^2}\mathbf{u}_K\mathbf{u}_K^H$. Hence, each column of $(\mathbf{H}\mathbf{H}^H)^{-1}$ is approximately a *scaled* version of $\mathbf{u}_K$. Since such a scaling is irrelevant for IR-LR, we replace $\mathbf{u}_K$ with an *arbitrary* column of $(\mathbf{H}\mathbf{H}^H)^{-1}$, i.e. Brun's algorithm is initialized as $\boldsymbol{\mu} = (\mathbf{H}\mathbf{H}^H)^{-1}\mathbf{e}_k$ with arbitrary k. Note that $(\mathbf{H}\mathbf{H}^H)^{-1}$ has already been calculated for the precoding matrix $\mathbf{P}$.

IR-LR-assisted vector perturbation further requires the inverse of $\mathbf{B}$ (cf. (8)). This inverse can be computed directly within Brun's algorithm by using a matrix $\mathbf{C} = [\mathbf{c}_1\ldots\mathbf{c}_K]^T$ initialized as $\mathbf{C} = \mathbf{I}$. Within each iteration, one performs the additional *row* update

$$\mathbf{C} = [\mathbf{c}_1\ldots\mathbf{c}_{t-1}\ \mathbf{c}_t'\ \mathbf{c}_{t+1}\ldots\mathbf{c}_K]^T$$

with

$$\mathbf{c}_t' = \mathbf{c}_t + \beta^*\,\mathbf{c}_s. \tag{17}$$

It is easily verified that $\mathbf{C}\mathbf{B} = \mathbf{I}$ at every iteration.

Finally, the reduced basis $\widetilde{\mathbf{P}}$ can itself be calculated efficiently during the Brun iterations via the initialization $\widetilde{\mathbf{P}} = \mathbf{P}$ and the update

$$\widetilde{\mathbf{P}} = [\tilde{\mathbf{p}}_1\ldots\tilde{\mathbf{p}}_{s-1}\ \tilde{\mathbf{p}}_s'\ \tilde{\mathbf{p}}_{s+1}\ldots\tilde{\mathbf{p}}_K]^T$$

with (cf. (15))

$$\tilde{\mathbf{p}}_s' = \mathbf{P}\mathbf{b}_s' = \mathbf{P}\mathbf{b}_s - \beta^*\,\mathbf{P}\mathbf{b}_t = \tilde{\mathbf{p}}_s - \beta^*\,\tilde{\mathbf{p}}_t, \tag{18}$$

Note that (18) allows efficient evaluation of the termination condition in (16).

The computational complexity of one iteration of the IR-LR algorithm is governed by the scalar division with quantization in (13), the scalar update for μ_s (15), three vector updates (cf. (15), (17), (18)), and the evaluation of the termination condition (16). Thus, one iteration requires just one scalar division and roughly $4K$ multiplications and additions. In constrast to the LLL algorithm no costly initialization has to be performed and we also observed that the number of iterations required by IR-LR is typically much less than the number of iterations required by the LLL algorithm. IR-LR is thus significantly less complex than LLL-LR (this will be verified numerically in Section 4).

4. SIMULATION RESULTS

We will now assess the symbol-error rate (SER) performance and computational complexity of our method by means of numerical simulations using a 4-QAM symbol alphabet and iid Gaussian channels. We compared ZF precoding and THP (unsorted) without LR, with LLL-LR, and with the proposed IR-LR. As a performance benchmark, we also considered exact solution of (6) via sphere encoding (SE).

4.1 SER Performance

Fig. 2(a) and Fig. 2(b) show the SER versus SNR=$1/\sigma_{\mathbf{w}}^2$ performance obtained using the various precoding schemes for $M = K = 4$ and $M = K = 8$, respectively. These results lead to the following conclusions:

- IR-LR significantly improves the performance of conventional ZF precoding and THP. While ZF and THP without LR just achieve diversity one (cf. [1]), IR-LR assisted ZF and THP achieve a large part of the available diversity. The performance improvements at SER $= 10^{-2}$ are about 8 dB for ZF and 5 dB for THP, both for $M = K = 4$ and $M = K = 8$.
- Since the IR-LR algorithm performs LR using just one singular vector, the IR-LR assisted schemes suffer from a performance loss compared to the LLL-LR assisted versions (which achieve full diversity [7]). For the case $M = K = 4$, the SNR loss at SER $= 10^{-2}$ is about 3 dB for ZF and 1 dB for THP.
- IR-LR assisted THP looses only 2dB compared to the optimal SE performance for $M = K = 4$ and only 3dB for $M = K = 8$ (again at SER $= 10^{-2}$).

[5] The averaging was performed over $k = 1,\ldots,K$ and 1000 randomly picked vectors $\mathbf{u}_K$.

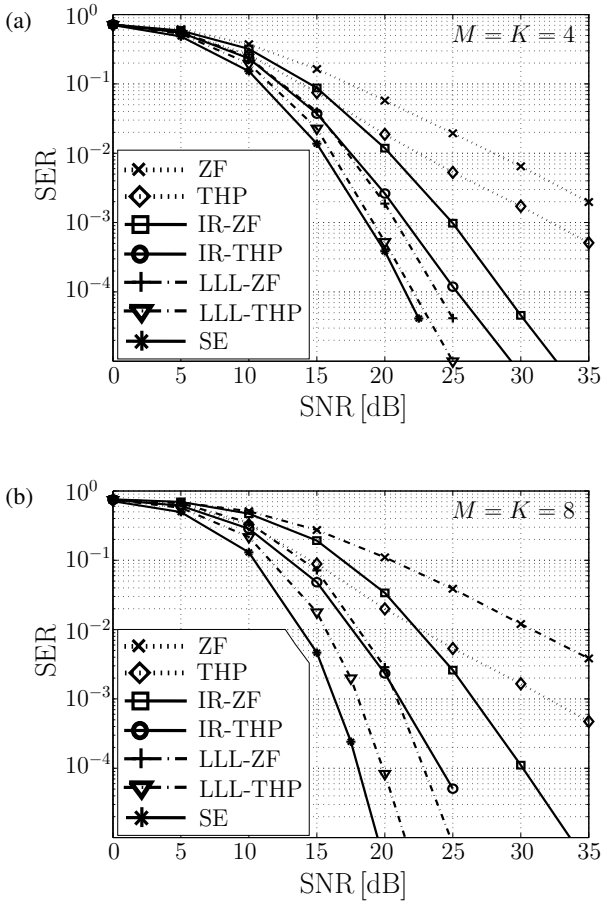

Figure 2: *SER versus SNR achieved for (a) $M = K = 4$, (b) $M = K = 8$, using ZF and THP without LR, with LLL-LR (denoted LLL-ZF and LLL-THP), and with the proposed IR-LR (denoted IR-ZF and IR-THP). Optimum vector perturbation (denoted SE) is shown as a benchmark.*

4.2 Computational Complexity

For a coarse comparison of the computational complexity of IR-LR and LLL-LR, we measured MATLAB kflops measured within 1000 channel realizations. Only the computation of the unimodular transformation matrices was considered since all other steps are identical for the two methods. The number of iterations, and thus the computational complexity, strongly depends on the individual channel realization. For the $M = K = 8$ case, the LLL-LR algorithm required up to 48.3 kflops with an average of 24 kflops. In contrast, with the IR-LR algorithm on average only 2 kflops and at most 8.6 kflops were required. Thus, the IR-LR algorithm is an order of magnitude more efficient than LLL-LR, which amounts roughly to 90% of computational savings. Similar results were obtained for the $M = K = 4$ case.

5. CONCLUSIONS

In this paper, we studied lattice reduction (LR) assisted approximate vector perturbation techniques for multi-antenna broadcast precoding. Previously, LR was performed almost exclusively using the LLL algorithm. We proposed a novel integer relation (IR) based LR algorithm that is based on Brun's algorithm. This was motivated by carefully examining the case of poorly conditioned channel realizations, for which LR is most important. The proposed IR-LR algorithm is extremely simple and essentially performs LR by determining approximate IRs for the singular vector of the channel corresponding to the smallest singular value. Simulation results demonstrated that a large part of the available diversity can be achieved with a computational complexity that is significantly smaller than that of LLL-LR.

REFERENCES

[1] C. B. Peel, B. M. Hochwald, and A. L. Swindlehurst, "A vector-perturbation technique for near-capacity multiantenna multiuser communication - Part I: Channel inversion and regularization," *IEEE Trans. Comm.*, vol. 53, pp. 195–202, Jan. 2005.

[2] B. M. Hochwald, C. B. Peel, and A. L. Swindlehurst, "A vector-perturbation technique for near-capacity multiantenna multiuser communication - Part II: Perturbation," *IEEE Trans. Comm.*, vol. 53, pp. 537–544, March 2005.

[3] C. Windpassinger, R. F. H. Fischer, T. Vencel, and J. B. Huber, "Precoding in multiantenna and multiuser communications," *IEEE Trans. Wireless Comm.*, vol. 3, pp. 1305–1316, July 2004.

[4] J. Jaldén and B. Ottersten, "An exponential lower bound on the expected complexity of sphere decoding," in *Proc. IEEE ICASSP 2004*, vol. IV, (Montreal, Canada), pp. 393–396, May 2004.

[5] C. Windpassinger, R. F. H. Fischer, and J. B. Huber, "Lattice-reduction-aided broadcast precoding," *IEEE Trans. Comm.*, vol. 52, pp. 2057–2060, Dec. 2004.

[6] A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovász, "Factoring polynomials with rational coefficients," *Math. Ann.*, vol. 261, pp. 515–534, 1982.

[7] M. Taherzadeh, A. Mobasher, and A. Khandani, "LLL lattice-basis reduction achieves the maximum diversity in MIMO systems," in *Proc. IEEE ISIT 2005*, (Adelaide, Australia), pp. 1300–1304, Sept. 2005.

[8] V. L. Clarkson, *Approximation of Linear Forms by Lattice Points with Applications to Signal Processing*. PhD thesis, Australian National University, 1997.

[9] M. Grötschel, L. Lovász, and A. Schrijver, *Geometric Algorithms and Combinatorial Optimization*. Berlin: Springer, 2nd ed., 1993.

[10] D. Wübben, R. Böhnke, V. Kühn, and K. Kammeyer, "Near-maximum-likelihood detection of MIMO systems using MMSE-based lattice-reduction," in *Proc. IEEE ICC 2004*, vol. 2, (Paris, France), pp. 798–802, june 2004.

[11] H. Vikalo and B. Hassibi, "On the sphere decoding algorithm II. Generalizations, second-order statistics, and applications to communications," *IEEE Trans. Signal Processing*, vol. 53, pp. 2819–2834, Aug. 2005.

[12] H. Yao and G. W. Wornell, "Lattice-reduction-aided detectors for MIMO communication systems," in *Proc. IEEE Globecom 2002*, vol. 1, (Taipeh, Taiwan), pp. 424–428, Nov. 2002.

[13] C. Windpassinger and R. F. H. Fischer, "Low-complexity near-maximum-likelihood detection and precoding for MIMO systems using lattice reduction," in *Proc. IEEE Information Theory Workshop*, (Paris, France), pp. 345–348, March/April 2003.

[14] G. H. Golub and C. F. Van Loan, *Matrix Computations*. Baltimore: Johns Hopkins University Press, 3rd ed., 1996.

[15] H. Artés, D. Seethaler, and F. Hlawatsch, "Efficient detection algorithms for MIMO channels: A geometrical approach to approximate ML detection," *IEEE Trans. Signal Processing*, vol. 51, pp. 2808–2820, Nov. 2003.

[16] H. Ferguson, D. Bailey, and S. Arno, "Analysis of PSLQ, an integer relation finding algorithm," *Math. Comput.*, vol. 68, no. 10, pp. 351–369, 1999.