From Gram–Schmidt Orthogonalization via Sorting and Quantization to Lattice Reduction

Robert F.H. Fischer
Lehrstuhl für Informationsübertragung, Friedrich–Alexander–Universität Erlangen–Nürnberg,
Cauerstrasse 7/LIT, 91058 Erlangen, Germany, Email: fischer@LNT.de

I. Introduction

Over the last years, numerous equalization schemes for *multiple-input/multiple-output (MIMO) channels* have been studied in literature. In particular, techniques known from intersymbol-interference channels have been transferred to the MIMO setting, such as linear equalization, decision-feedback equalization (DFE, also known as successive interference cancellation, SIC, and the main ingredient of the BLAST approach), and maximum-likelihood detection, cf. [3, Table E.1].

Besides them, new approaches based on *lattice basis reduction*, e.g., [16], [11], are of special interest. Using these *lattice-reduction-aided (LRA) techniques*, low-complexity equalization achieving the optimum diversity behavior [9] is enabled.

In this contribution, the connection of the *Gram–Schmidt procedure*—well-known from linear algebra for calculating an orthogonal/orthonormal basis for a vector space—to decision-feedback equalization and to lattice reduction and lattice-reduction-aided equalization, respectively, is enlightened. It is shown that the operations *quantization* and *sorting* play an important role. Their consequences for the universal tool Gram–Schmidt procedure and the connection to the LLL algorithm with deep insertions [8] are explained.

II. Gram–Schmidt Orthogonalization

Assume we are operating on $\mathbb{R}^N$ and let K ($K \leq N$) linearly independent vectors $\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_K$ be given.[1] These vectors span some K-dimensional subspace of $\mathbb{R}^N$, i.e., are a basis for this subspace.

Given the $\boldsymbol{h}_i$'s, the *Gram–Schmidt (GS)* orthogonalization procedure[2] result in new vectors $\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots, \boldsymbol{q}_K$, such that they are mutually orthogonal ($\boldsymbol{q}_k^{\mathsf{T}}\boldsymbol{q}_l = 0$, $k \neq l$) and span the same K-dimensional subspace, e.g. [4]. If desired, the new basis may also be *normalized*, i.e., $\boldsymbol{q}_k^{\mathsf{T}}\boldsymbol{q}_k = ||\boldsymbol{q}_k||^2 = 1$, $\forall k$—*orthonormal vectors*, Gram–Schmidt orthonormalization.

A possible implementation of Gram–Schmidt orthogonalization (steps for orthonormalization are shown in square brackets) is given in Algorithm 1. For convenience, we combine the vectors and scalars, respectively, in matrices as follows

$$\boldsymbol{H} \stackrel{\text{def}}{=} \left[h_{ij}\right] = \left[\boldsymbol{h}_1 \ldots \boldsymbol{h}_K\right] \tag{1}$$

$$\boldsymbol{Q} \stackrel{\text{def}}{=} \left[q_{ij}\right] = \left[\boldsymbol{q}_1 \ldots \boldsymbol{q}_K\right] \tag{2}$$

$$\boldsymbol{R} \stackrel{\text{def}}{=} \left[r_{ij}\right] = \left[\boldsymbol{r}_1 \ldots \boldsymbol{r}_K\right] . \tag{3}$$

The coefficients r_{ij} describe the basis change; the initial basis vectors $\boldsymbol{h}_k$ are given as a linear combination of the new vectors $\boldsymbol{q}_k$, i.e., $\boldsymbol{h}_k = \sum_{l=1}^{k} r_{lk}\boldsymbol{q}_l$. Due to construction, $\boldsymbol{Q}$ is an orthogonal matrix ($\boldsymbol{Q}^{\mathsf{T}}\boldsymbol{Q} = \mathbf{diag}\left(\boldsymbol{q}_1^{\mathsf{T}}\boldsymbol{q}_1, \ldots, \boldsymbol{q}_K^{\mathsf{T}}\boldsymbol{q}_K\right)$) or an orthonormal matrix ($\boldsymbol{Q}^{\mathsf{T}}\boldsymbol{Q} = \boldsymbol{I}$), and $\boldsymbol{R}$ is upper triangular. Hence, we have

$$\boldsymbol{H} = \boldsymbol{Q}\boldsymbol{R} \,, \tag{4}$$

i.e., the Gram–Schmidt process induces a *QR decomposition* of the initial basis matrix $\boldsymbol{H}$. Subsequently, we only consider orthogonalization.

In linear algebra, usually the basis $\boldsymbol{Q}$ is of main interest and the matrix $\boldsymbol{R}$ of less importance. Moreover, the order of the vectors $\boldsymbol{q}_i$ is often irrelevant, as a permutation of the vectors does not change the spanned subspace.

Besides linear algebra, the Gram–Schmidt procedure also plays an important role in digital communications. Given the signal elements of a general digital modulation scheme, an orthonormal basis spanning the signal space can be obtained via the Gram–Schmidt orthogonalization, e.g. [14]. Using this basis, modulator and, in particular, the optimum receiver can be derived immediately.

Algorithm 1 Gram–Schmidt orthogonal-/orthonormalization.

$[\boldsymbol{Q}, \boldsymbol{R}] = \texttt{GramSchmidt}(\boldsymbol{H})$

- $\boldsymbol{Q} = \boldsymbol{H}, \;\; \boldsymbol{R} = \boldsymbol{0}$
- $k = 1$
- **while** $k \leq K$ {
  - $r_{kk} = 1$ $\quad\left[\, r_{kk} = \sqrt{\boldsymbol{q}_k^{\mathsf{T}}\boldsymbol{q}_k} \,\right]$
  - $\quad\left[\, \boldsymbol{q}_k = \boldsymbol{q}_k / r_{kk} \,\right]$
  - **for** $i = k+1, \ldots, K$ {
    - $r_{ki} = \boldsymbol{q}_k^{\mathsf{T}}\boldsymbol{q}_i / (\boldsymbol{q}_k^{\mathsf{T}}\boldsymbol{q}_k)$ $\quad\left[\, r_{ki} = \boldsymbol{q}_k^{\mathsf{T}}\boldsymbol{q}_i \,\right]$
    - $\boldsymbol{q}_i = \boldsymbol{q}_i - r_{ki}\boldsymbol{q}_k$
  - }
  - $k = k + 1$
- }

[1] Notation: matrices are denoted by bold uppercase letters, column vectors by bold lower case letters. $\boldsymbol{A}^{\mathsf{T}}$, $\boldsymbol{a}^{\mathsf{T}}$: transpose of a matrix or a vector; $\boldsymbol{A}^{-1}$: inverse of the square matrix; $\boldsymbol{I}$: identity matrix; $\boldsymbol{0}$: null matrix; $||\boldsymbol{a}|| = \sqrt{\boldsymbol{a}^{\mathsf{T}}\boldsymbol{a}}$: Euclidean norm of the vector $\boldsymbol{a}$. For brevity, the basic principles are described for real-valued vectors. However, almost all statements are equally valid for vectors from $\mathbb{C}^N$. In this case, transposition $\cdot^{\mathsf{T}}$ has to be replaced by conjugate transposition, i.e., hermitian operation $\cdot^{\mathsf{H}}$.

[2] Named after the Danish mathematician Jørgen Pedersen Gram (1850–1916) and the German mathematician Erhard Schmidt (1876–1959).

III. Decision-Feedback Equalization

Next, we consider tranmission over a flat fading channel with K transmit and N receive signals. The fading coefficients h_{ij} from transmit antenna j to receive antenna i are collected in the channel matrix $\boldsymbol{H} = \left[h_{ij} \right]$. The column vectors $\boldsymbol{h}_j$ of $\boldsymbol{H}$ contain the transmission factors from transmit antenna j to all receive antennas; they span the space, the receive signal lies. Furthermore, additive (spatially white) Gaussian noise (variance σ_n^2 per component) is present. Denoting the vector of data symbols by $\boldsymbol{a}$, the classical input/output relation

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{a} + \boldsymbol{n} \tag{5}$$

holds.

The interference between the parallel data streams has to be eliminated by means of equalization. Using linear equalization (either optimized according to the zero-forcing (ZF) or the minimum mean-squared error (MMSE) criterion) via[3] $\boldsymbol{r} = \boldsymbol{H}_{\mathrm{R}}\boldsymbol{y}$ a decision vector is generated, on which individual (per component) threshold decision can be performed to recover data.

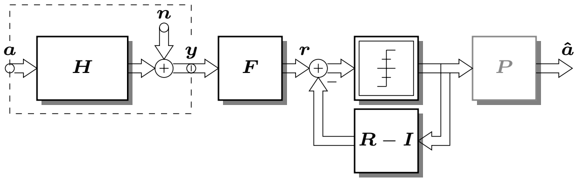

Fig. 1. MIMO transmission systems using decision-feedback equalization.

Some gains over linear equalization can be achieved by using *decision-feedback equalization*, also known as BLAST or SIC, see Figure 1. Key ingredient for ZF DFE[4] is the QR decomposition of the channel matrix, such that (cf. (4))

$$\boldsymbol{H} = \boldsymbol{Q}\boldsymbol{R}\,, \tag{6}$$

where $\boldsymbol{Q}$ is orthogonal and $\boldsymbol{R}$ is upper triangular with unit main diagonal. The factorization can simply be obtained applying the Gram–Schmidt procedure. However, in the present case, the upper triangular matrix $\boldsymbol{R}$ is of more interest. Calculating via feedforward processing (feedforward matrix $\boldsymbol{F} \stackrel{\text{def}}{=} (\boldsymbol{Q}^\mathsf{T}\boldsymbol{Q})^{-1}\boldsymbol{Q}^\mathsf{T}$)

$$\begin{aligned} \boldsymbol{r} &= \boldsymbol{F}\boldsymbol{y} \\ &= (\boldsymbol{Q}^\mathsf{T}\boldsymbol{Q})^{-1}\boldsymbol{Q}^\mathsf{T}\boldsymbol{H}\boldsymbol{a} + (\boldsymbol{Q}^\mathsf{T}\boldsymbol{Q})^{-1}\boldsymbol{Q}^\mathsf{T}\boldsymbol{n} \\ &= (\boldsymbol{Q}^\mathsf{T}\boldsymbol{Q})^{-1}\boldsymbol{Q}^\mathsf{T}\boldsymbol{Q}\boldsymbol{R}\boldsymbol{a} + (\boldsymbol{Q}^\mathsf{T}\boldsymbol{Q})^{-1}\boldsymbol{Q}^\mathsf{T}\boldsymbol{n} \\ &= \boldsymbol{R}\boldsymbol{a} + \tilde{\boldsymbol{n}}\,, \end{aligned} \tag{7}$$

we see that the effective transmission matrix is $\boldsymbol{R}$, hence has triangular form. Thereby *spatially causality* is achieved; tranmission symbols with larger index disturb those with smaller one but not the other way round. Due to the orthogonal transformation, the new noise vector $\tilde{\boldsymbol{n}}$ is also Gaussian and white but with variances $\sigma_{\tilde{n}_k}^2 = \sigma_n^2(\boldsymbol{q}_k^\mathsf{T}\boldsymbol{q}_k)^{-1}$.

Due to causality, the data symbols can be detected in a successive manner. Having decisions on symbols $K, K-1, \ldots, k{+}2, k{+}1$, their interference into data symbol k (given by the factors r_{ki}) can be eliminated, and the kth symbol is decided. This procedure is repeated until all K transmit symbols are known (detection in reverse natural order).

The signal-to-noise ratio in component k is proportional to $||\boldsymbol{q}_k||^2$. Hence, for best performance (most reliable decisions), the norms of the vectors $\boldsymbol{q}_k$ should be as large as possible. However, since decisions are taken (quantization is a non-linear operation) within the equalization procedure, the decision order influences performance. Since the receiver operates on the entire receive vector it is not necessary to detect the data symbols in (reverse) natural order but an optimized one can be used. Defining a criteria of optimality—usually the symbol to be detected next among the not yet detected one is that one with the largest signal-to-noise ratio, cf. the BLAST ordering [13]—among all possible $K!$ detection orders the best one can be identified.

The optimum detection order can be described by a (virtual) relabeling of the transmit antennas, corresponding to a column-wise permutation of the channel matrix $\boldsymbol{H}$. Instead of $\boldsymbol{H}$, the matrix $\boldsymbol{C} \stackrel{\text{def}}{=} \boldsymbol{H}\boldsymbol{P}$, where $\boldsymbol{P}$ is a permutation matrix (a single one in each row and column), is considered. Decisions are still taken in the reverse natural order ($K, K{-}1, \ldots, 1$); having all estimates, via the permutation matrix $\boldsymbol{P}$ the original sequence is re-established, cf. Figure 1.

Hence, we end up with a *sorted QR decomposition* of the channel matrix according to

$$\boldsymbol{C} \stackrel{\text{def}}{=} \boldsymbol{H}\boldsymbol{P} = \boldsymbol{Q}\boldsymbol{R}\,. \tag{8}$$

In [13] an algorithm for finding the optimum permutation and feedforward and feedback matrices for DFE has been given. Later on, more efficient variants, e.g., [5], [1], have been derived in literature.

Essentially, the sorted QR decomposition is nothing else than a *Gram–Schmidt procedure with pivoting*. In [15] a simple pivoting has been proposed which, however, shows some loss compared to the BLAST algorithm. Basically, instead of maximizing $||\boldsymbol{q}_k||^2$ in sequence of detection ($k = K, K-1, \ldots, 1$), it is minimized for $k = 1, 2, \ldots, K$.

In the above Algorithm 1, only few new lines have to be additionally introduced, see Algorithm 2. Lines already present in Algorithm 1 are printed in gray. Instead of a column exchange, here an insertion of column i between columns $k{-}1$ and k (thereby columns k through $i-1$ are shifted by one position to the right) is used.

In summary, quantization of the involved signal induces the need to look for a suited ordering for performing these non-linear operations. In other words, as long as only linear operations are carried out, sorting is of no importance; as soon as non-linear operations are present, performance can be improved by sorting.

[3] $\boldsymbol{H}_{\mathrm{R}}$ is a (regularized) pseudo inverse of $\boldsymbol{H}$.

[4] Note that MMSE DFE can be obtained as for the ZF case when using augmented matrices and vectors, cf. [5], [12].

Algorithm 2 GS with pivoting.

$[\boldsymbol{Q}, \boldsymbol{R}, \boldsymbol{P}, \boldsymbol{C}] = \texttt{GramSchmidtSort}(\boldsymbol{H})$

$\boldsymbol{Q} = \boldsymbol{H}, \;\; \boldsymbol{R} = \boldsymbol{0}, \;\; \boldsymbol{C} = \boldsymbol{H}, \;\; \boldsymbol{P} = \boldsymbol{I}$
$k = 1$
while $k \leq K$ {
  $i = \operatorname{argmin}_{j=k,\ldots,K} ||\boldsymbol{q}_j||^2$
  if $i \neq k$ {
    $\texttt{Insert}(i, k)$
  }
  $r_{kk} = 1$
  for $i = k + 1, \ldots, K$ {
    $r_{ki} = \boldsymbol{q}_k^\top \boldsymbol{q}_i / (\boldsymbol{q}_k^\top \boldsymbol{q}_k)$
    $\boldsymbol{q}_i = \boldsymbol{q}_i - r_{ki} \boldsymbol{q}_k$
  }
  $k = k + 1$
}

$\texttt{Insert}(i, k)$

in $\boldsymbol{C}$, $\boldsymbol{Q}$, $\boldsymbol{R}$, and $\boldsymbol{P}$: insert column i between columns $k - 1$ and k; delete old column i

IV. Lattice-Reduction-Aided Decision-Feedback Equalization

A. Basic Operation

The idea of *lattice-reduction-aided equalization* [16], [11] is to interpret the (noise-less) signal at the output of the MIMO channel as points taken from the lattice spanned by the columns $\boldsymbol{h}_k$ of the channel matrix $\boldsymbol{H}$. Choosing a "more suited" representation of this lattice, a so-called *reduced basis*, within the DFE loop not the data symbols itself but integer linear combinations thereof are recovered. After having these decisions, the change of basis is undone. In other words, equalization is carried out with respect to the new basis, which is desired to be close to orthogonal,[5] hence less noise enhancement results. Meanwhile it has been proven that lattice-reduction-aided equalization achieves full diversity provided by the MIMO channel [9].

Given the $N \times K$ channel matrix $\boldsymbol{H}$, in the first step lattice basis reduction—e.g., by using the LLL algorithm [6]—is performed. For obtaining equalization optimized according to the zero-forcing criterion,[6] the factorization of the channel matrix reads

$$\boldsymbol{H} = \boldsymbol{H}_{\mathrm{red}} \boldsymbol{T} \,. \tag{9}$$

Thereby, $\boldsymbol{T}$ is an unimodular matrix (only integer coefficients and $|\det(\boldsymbol{T})| = 1$) and the columns of $\boldsymbol{H}_{\mathrm{red}}$ should be close to orthogonal and should have small norms.

[5] A quantitative measure is the so-called *orthogonality defect*. For a given matrix $\boldsymbol{H}$ with columns $\boldsymbol{h}_k$ it is defined as $\delta = \prod_{k=1}^{K} ||\boldsymbol{h}_k|| / |\det(\boldsymbol{H})|$. It is easy to see that $\delta \geq 1$, $\forall \boldsymbol{H}$, and $\delta = 1$, iff all columns of $\boldsymbol{H}$ are pairwise orthogonal.

[6] As above, optimization according to the minimum mean-squared error criterion may be obtained by replacing the matrices $\boldsymbol{H}$ and $\boldsymbol{Q}$ by their augmented versions.

In the second step, for performing *LRA decision-feedback equalization* $\boldsymbol{H}_{\mathrm{red}}$ is further decomposed according to

$$\boldsymbol{H}_{\mathrm{red}} \boldsymbol{P} = \boldsymbol{Q}\boldsymbol{R} \,, \tag{10}$$

i.e., again a sorted QR decomposition has to be performed.

The transmission system employing LRA DFE is depicted in Fig. 2. Feedforward and feedback matrices are thereby given as above (for more details see [12]). Since via DFE only $\boldsymbol{H}_{\mathrm{red}}$ is equalized, estimates of the actual data symbols are finally generated using $\boldsymbol{T}^{-1}$.

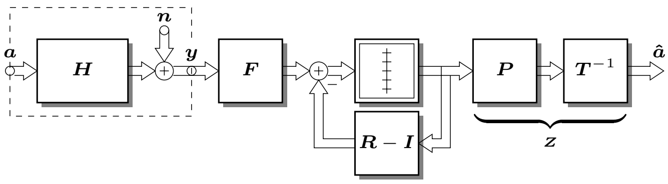

Fig. 2. MIMO transmission systems using lattice-basis-reduction-aided decision-feedback equalization.

B. Combined Lattice Reduction and QR Decomposition

The obvious procedure for LRA DFE would be to perform lattice reduction and QR decomposition of the reduced channel matrix in sequence. However, it has been recognized, e.g., [10], [7], that parts of the complexity can be saved as there are interactions between both steps.

Combining (9) and (10), and defining $\boldsymbol{Z} \stackrel{\mathrm{def}}{=} \boldsymbol{T}^{-1}\boldsymbol{P}$, which is also unimodular, we arrive at

$$\boldsymbol{C} \stackrel{\mathrm{def}}{=} \boldsymbol{H}\boldsymbol{Z} = \boldsymbol{Q}\boldsymbol{R} \,. \tag{11}$$

Hence, for performing LRA DFE a suited unimodular matrix $\boldsymbol{Z}$ (replacing the permutation matrix $\boldsymbol{P}$ in conventional DFE) has to be found such that $\boldsymbol{C}$ is a (sorted and) reduced basis. At the same time, the QR decomposition of $\boldsymbol{C}$ has to be calculated.

Fortunately, the famous lattice reduction algorithm proposed by *Lenstra, Lenstra, and Lovász (LLL)* [6] relies on the QR decomposition. The basic idea of the LLL algorithm is to calculate $\boldsymbol{Q}$ and $\boldsymbol{R}$ and then to quantize the coefficients of $\boldsymbol{R}$ to the set of integers. The inverse of this integer matrix—which is upper triangular with unit main diagonal, hence unimodular—is a candidate for the desired unimodular matrix $\boldsymbol{Z}$. If $\boldsymbol{R}$ would be integer, the reduced basis would be given by $\boldsymbol{Q}$; an orthogonal basis would be present. Unfortunately, this will usually be not the case and the columns of $\boldsymbol{C} \stackrel{\mathrm{def}}{=} [c_{ij}] = [\boldsymbol{c}_1 \ldots \boldsymbol{c}_K]$ will only be approximately orthogonal.

Basically, the LLL can be obtained from the above Gram–Schmidt procedure by incorporating a *size reduction step*. This multiplication of all terms in (11) from the right by the inverse of the quantized version of $\boldsymbol{R}$ is done successively. Each time (step k) a new vector $\boldsymbol{q}_k$ is processed not only its projection—which is a *real-valued multiple* of $\boldsymbol{q}_k$—onto the remaining vectors $\boldsymbol{q}_{k+1}, \ldots, \boldsymbol{q}_K$ is subtracted, but also an *integer multiple*, say r, of the already processed vectors $\boldsymbol{c}_i$, $i = k-1, \ldots, 1$, is subtracted from $\boldsymbol{c}_k$ (this operation

Algorithm 3 GS with pivoting and reduction.

$[\boldsymbol{Q}, \boldsymbol{R}, \boldsymbol{Z}, \boldsymbol{C}] = \texttt{GramSchmidtRed}(\boldsymbol{H})$

$\boldsymbol{Q} = \boldsymbol{H}, \;\; \boldsymbol{R} = 0, \;\; \boldsymbol{C} = \boldsymbol{H}, \;\; \boldsymbol{Z} = \boldsymbol{I}$
$k = 1$
while $k \leq K$ {
  $i = \operatorname{argmin}_{j=k,\ldots,K} ||\boldsymbol{q}_j||^2$
  if $i \neq k$ {
    $\texttt{Insert}(i, k)$
  }
  $r_{kk} = 1$
  for $i = k+1, \ldots, K$ {
    $r_{ki} = \boldsymbol{q}_k^{\top} \boldsymbol{q}_i / (\boldsymbol{q}_k^{\top} \boldsymbol{q}_k)$
    $\boldsymbol{q}_i = \boldsymbol{q}_i - r_{ki} \boldsymbol{q}_k$
  }
  for $i = k-1, k-2, \ldots, 1$ {
    $\texttt{Red}(k, i)$
  }
  $k = k + 1$
}

$\texttt{Insert}(i, k)$

in $\boldsymbol{C}$, $\boldsymbol{Q}$, $\boldsymbol{R}$, and $\boldsymbol{Z}$: insert column i between columns $k-1$ and k; delete old column i

$\texttt{Red}(k, i)$

$r = \lfloor r_{ik} \rceil$
$\boldsymbol{c}_k = \boldsymbol{c}_k - r \cdot \boldsymbol{c}_i$
$\boldsymbol{r}_k = \boldsymbol{r}_k - r \cdot \boldsymbol{r}_i$
$\boldsymbol{z}_k = \boldsymbol{z}_k - r \cdot \boldsymbol{z}_i$

is kept track in $\boldsymbol{Z}$). Choosing[7] $r = \lfloor r_{ik} \rceil$, i.e., calculating $\boldsymbol{c}_k = \boldsymbol{c}_k - \lfloor r_{ik} \rceil \boldsymbol{c}_i$, the off-diagonal elements of the final $\boldsymbol{R}$ have magnitude not exceeding $1/2$. This is one of the conditions for $\boldsymbol{C}$ being *"LLL reduced"*, cf. [6].

A first approach to combined Gram–Schmidt procedure and lattice reduction is summarized in Algorithm 3. Again, the lines printed in gray are already present in Algorithm 2.

C. LLL with Deep Insertions and Pivoting

The above algorithm achieves gains with respect to the orthogonality defect of $\boldsymbol{C}$ but as $\boldsymbol{Q}$ remains unchanged, no performance gains in DFE are possible, cf. [10]. Hence there is still room for improvement.

In the Gram–Schmidt procedure no quantization at all is present. Performing (conventional) DFE, quantization of the signals, *outside the GS algorithm*, takes place, leading to the necessity of finding a suited permutation of the columns of the channel matrix. The quantization results (decisions) influence the decisions of the not yet detected symbols. Hence, quantization acts in a causal manner and the greedy approach of BLAST or sorted QR decomposition to find a suited sorting is justified.

[7] $\lfloor x \rceil$: quantization (rounding) to the nearest integer.

When additionally including size reduction, *quantization within the GS algorithm* is done. Now, this operation not only influences the not yet processed vectors but also has an effect on the already calculated ones. In turn, for obtaining optimum results, the need for revisiting prior processed vectors arises. However, as in the case of sorting/pivoting a criterion for optimality is required—here a criterion of when and how far going back.

The LLL algorithm requires that $|r_{ik}| \leq 1/2, \; \forall i < k$. This reflects the degree of orthogonality of the vectors $\boldsymbol{c}_k$ of the reduced basis. Additionally,

$$||\boldsymbol{q}_k||^2 + r_{k-1,k}^2 ||\boldsymbol{q}_{k-1}||^2 \geq \frac{3}{4} ||\boldsymbol{q}_{k-1}||^2 \,, \qquad 1 < k \leq K \tag{12}$$

has to hold, which gives a criterion on the sorting of the basis vectors with respect to their length. In the basic LLL algorithm adjacent basis vectors are compared and if (12) is not fulfilled they are swapped. Then, k is decremented, hence, the algorithm goes back one step in the GS procedure.

In [8] Schnorr and Euchner proposed a different approach. Instead of swapping adjacent vectors, the vector $\boldsymbol{c}_k$ may be inserted between vectors $\boldsymbol{c}_{i-1}$ and $\boldsymbol{c}_i$. When doing so, and recalculating the Gram–Schmidt factorization, the new vector at position i, say $\boldsymbol{q}'_i$, is given by $\boldsymbol{c}_k$ minus the projections of the already known basis vectors $\boldsymbol{q}_j$, $j = 1, \ldots, i-1$. Hence, as the projections r_{jk} are already known, it calculates to

$$\boldsymbol{q}'_i = \boldsymbol{c}_k - \sum_{j=1}^{i-1} r_{jk} \boldsymbol{q}_j \tag{13}$$

and using $\boldsymbol{c}_k = \sum_{j=1}^{k} r_{jk} \boldsymbol{q}_j$, with $r_{kk} = 1$, we arrive at

$$= \boldsymbol{q}_k + \sum_{j=i}^{k-1} r_{jk} \boldsymbol{q}_j \,. \tag{14}$$

Taking the pair-wise orthogonality of the vectors $\boldsymbol{q}_k$ into account, the squared length of the new vector $\boldsymbol{q}'_i$ is hence given as [2]

$$||\boldsymbol{q}'_i||^2 = ||\boldsymbol{c}_k||^2 - \sum_{j=1}^{i-1} r_{jk}^2 ||\boldsymbol{q}_j||^2 \tag{15}$$

$$= ||\boldsymbol{q}_k||^2 + \sum_{j=i}^{k-1} r_{jk}^2 ||\boldsymbol{q}_j||^2 \,. \tag{16}$$

If the length of the new vector $\boldsymbol{q}'_i$ is significantly smaller than the length of the old vector $\boldsymbol{q}_i$, say,

$$||\boldsymbol{q}'_i||^2 \leq \frac{3}{4} ||\boldsymbol{q}_i||^2 \,, \tag{17}$$

then it is reasonable to do the insertion and obtain a Gram–Schmidt basis with shorter vectors and in turn a "more reduced" basis of the lattice. After the insertion, k is reset to i and the Gram–Schmidt procedure is restarted from this point.

Algorithm 4 LLL with deep insertions.

$[\boldsymbol{Q}, \boldsymbol{R}, \boldsymbol{Z}, \boldsymbol{C}] = \texttt{LLLdeep}(\boldsymbol{H})$

```
Q = H,  R = 0,  C = H,  Z = I
k = 1
while k ≤ K {
    i = argmin_{j=k,...,K} ||q_j||^2
    if i ≠ k {
        Insert(i, k)
    }
    r_kk = 1
    for i = k+1, ..., K {
        r_ki = q_k^T q_i / (q_k^T q_k)
        q_i = q_i − r_ki q_k
    }
    for i = k−1, k−2, ..., 1 {
        Red(k, i)
    }
    find i such that ||c_k||^2 − Σ_{j=1}^{i−1} r_jk^2 ||q_j||^2 ≤ 3/4 ||q_i||^2
    if i ≠ k {
        Insert(k, i)
        recalculate q_i, ..., q_K and r_ij, j = i, ..., K
        k = i
    }
    k = k + 1
}
```

$\texttt{Insert}(i, k)$

```
in C, Q, R, and Z: insert column i between columns
    k − 1 and k; delete old column i
```

$\texttt{Red}(k, i)$

```
r = ⌊r_ik⌉
c_k = c_k − r · c_i
r_k = r_k − r · r_i
z_k = z_k − r · z_i
```

Due to (15), the test can be done successively: starting with $L_1 = ||\boldsymbol{c}_k||^2$ it is tested against $||\boldsymbol{q}_1||^2$. At each step $L_{i+1} = L_i - r_{ik}^2||\boldsymbol{q}_i||^2$ is calculated and tested against $||\boldsymbol{q}_{i+1}||^2$, $i = 1, \ldots, k-2$. It should be noted that from (16) and (17) it can be seen that for comparing adjacent vectors the same condition (12) as in the basic LLL results.

A possible implementation of the LLL algorithm with deep insertions—described as an extended Gram–Schmidt procedure—is summarized in Algorithm 4. As above, the lines printed in gray are already present in Algorithm 3. The main ingredients are i) pivoting (insert vector to be processed next at the current position), ii) size reduction, and iii) back insertion (insert current vector at position within the already processed ones).

It should be noted, that the LLL with deep insertions (e.g., [2]) is usually described without the pivoting (initial sorting) step. However, introducing this operation, at no additional cost (assuming that it is kept track of the length of the vectors $\boldsymbol{q}_k$ anyway) a significant reduction in complexity (e.g., how many times the while loop is carried out) is possible.

V. Conclusions

To summarize, in this paper, the connection of the sorted Gram–Schmidt procedure and its use in decision-feedback equalization schemes on the one side and the LLL algorithm and its use in lattice-reduction-aided equalization schemes on the other side has been studied. It has been shown that the LLL with pivoting and deep insertions is a logical development staring from the basic Gram–Schmidt procedure.

Acknowledgment

This work was supported in parts by Deutsche Forschungsgemeinschaft (DFG) within the framework COIN under grant FI 982/4-1.

References

[1] J. Benesty, Y. Huang, J. Chen. A Fast Recursive Algorithm for Optimum Sequential Signal Detection in a BLAST System. *IEEE Transactions on Signal Processing*, Vol. 51, No. 7, pp. 1722–1730, July 2003.

[2] H. Cohen. *A Course in Computational Algebraic Number Theory*, Springer, Berlin, second printing, 1995.

[3] R.F.H. Fischer. *Precoding and Signal Shaping for Digital Transmission*, John Wiley & Sons, New York, 2002.

[4] G.H. Golub, C.F. van Loan. *Matrix Computations*, The Johns Hopkins University Press, Baltimore, third edition, 1996.

[5] B. Hassibi. An Efficient Square-Root Algorithm for BLAST. *IEEE International Conference on Acoustics, Speech, and Signal Processing* pp. 737–740, Istanbul, Turkey, June 2000.

[6] A.K. Lenstra, H.W. Lenstra, L. Lovász. Factoring polynomials with rational coefficients, *Mathematische Annalen*, pp. 515–534, 1982.

[7] C. Ling, W.H. Mow. A Unified View of Sorting in Lattice Reduction: From V-BLAST to LLL and Beyond. *IEEE Information Theory Workshop*, pp. 529–533, Taorminy, Italy, Oct. 2009.

[8] C.P. Schnorr, M. Euchner. Lattice Basis Reduction: Improved Practical Algorithms and Solving Subset Sum Problems. *Mathematical Programming*, pp. 181–199, 1994.

[9] M. Taherzadeh, A. Mobasher, A.K. Khandani. LLL Reduction Achieves the Receive Diversity in MIMO Decoding. *IEEE Transactions on Information Theory*, Vol. 53, No. 12, pp. 4801–4805, Dec. 2007.

[10] D.W. Waters, J.R. Berry. A Reduced-Complexity Lattice-Aided Decision-Feedback Detector. *IEEE International Conference on Wireless Networks, Communications and Mobile Computing*, pp. 845–850, Maui, HI, USA, June 2005.

[11] C. Windpassinger, R.F.H. Fischer. Low-Complexity Near-Maximum-Likelihood Detection and Precoding for MIMO Systems using Lattice Reduction. *IEEE Information Theory Workshop*, pp. 345–348, Paris, France, Mar./Apr. 2003.

[12] C. Windpassinger. *Detection and Precoding for Multiple Input Multiple Output Channels*. Dissertation, Erlangen, June 2004.

[13] P. Wolniansky, G. Foschini, G. Golden, R. Valenzuela. V-BLAST: An Architecture for Realizing Very High Data Rates Over the Rich-Scattering Wireless Channel. *URSI International Symposium on Signals Systems and Electronics*, Pisa, Italy, Sep. 1998.

[14] J.M. Wozencraft, I.M. Jacobs. *Principles of Communication Engineering*, John Wily & Sons, New York, 1965.

[15] D. Wübben, R. Böhnke, J. Rinas, V. Kühn, K.D. Kammeyer. Efficient Algorithm for Decoding Layered Space-Time Codes. *IEE Electronics Letters*, Vol. 37, No. 22, pp. 1348–1350, Oct. 2001.

[16] H. Yao, G.W. Wornell. Lattice-Reduction-Aided Detectors for MIMO Communication Systems. *IEEE Global Communications Conference*, Taipei, Taiwan, Nov. 2002.