ETH zürich

ETH Library

Multilevel Codes in Lattice-Reduction-Aided Equalization

Conference Paper

Author(s):
Fischer, Robert F.H.; Huber, Johannes B.; Stern, Sebastian; Guter, Paulus M.

Publication date:
2018-02

Permanent link:
https://doi.org/10.3929/ethz-b-000245081

Rights / license:
In Copyright - Non-Commercial Use Permitted

This page was generated automatically upon download from the ETH Zurich Research Collection.
For more information, please consult the Terms of use.

International Zurich Seminar on Information and Communication (IZS), February 21 – 23, 2018

Multilevel Codes in Lattice-Reduction-Aided Equalization

Robert F.H. Fischer[1], Johannes B. Huber[2], Sebastian Stern[1], Paulus M. Guter[2]

[1]Institut für Nachrichtentechnik, Universität Ulm, Ulm, Germany, Email: {robert.fischer,sebastian.stern}@uni-ulm.de
[2]Lehrstuhl für Informationsübertragung, Universität Erlangen-Nürnberg, Erlangen, Germany, Email: johannes.huber@fau.de

***Abstract*—The application of multilevel codes in lattice-reduction-aided equalization is considered, i.e., the decoding of multilevel codes when (Gaussian) integer linear combinations of the codewords in signal space are present. Typically, multilevel codes do not generate lattice codes, hence arbitrary integer linear combinations are not directly decodable. We show that this lattice property is not required, which relaxes the constraints on the component codes significantly. A generalized version of multistage decoding which incorporates a "carry correction" is proposed; it circumvents the lattice property and any integer linear combinations are decodable. Numerical simulations are given to cover the performance of the proposed method.**

I. Introduction

In the literature, low-complexity but well-performing approaches for the equalization in *multiple-input/multiple-output (MIMO) multi-user uplink scenarios* are discussed for more than one decade. *Lattice-reduction-aided* (LRA) techniques [16], [15] and the tightly related concept of *integer-forcing* (IF) receivers [10], [17] are of special interest. In both schemes, the main idea is not to decode the transmitted signals, but *integer linear combinations* thereof; LRA and IF receivers differ in the way the (residual) integer interference is handled, cf. [4].

Right from the start, IF schemes were proposed as coded schemes—a strong coupling between integer equalization and decoding/code constraints is present. In contrast, LRA schemes are usually treated uncoded as a combination with coded modulation schemes seems to be easier. Here, the actual demand is that the code is linear in signal space; any integer linear combination of codewords has to be a codeword. Lattice codes obviously fulfill this demand.

In this paper, we are interested in the combination of *multilevel codes (MLC)* and LRA equalization, because multilevel coding together with *multistage decoding (MSD)* is in principle a capacity-achieving strategy [14]. Typically, multilevel codes do not generate lattice codes [9] and are not linear. However, we show that this property is indeed not required. We study the relevant design criteria on the component codes and propose a generalized version of multistage decoding which incorporates a "*carry correction*" such that integer linear combinations are decodable.[1]

The paper is organized as follows: In Sec. II the system model is introduced. The consequences of integer linear combinations on multilevel codes are studied in Sec. III and a new decoding scheme with carry correction is proposed. Sec. IV presents numerical examples; the paper is briefly summarized in Sec. V.

II. System Model

Throughout the paper, we assume K non-cooperating (single-antenna) users k, $k = 1, \ldots, K$, communicating their binary source symbols[2] $\mathfrak{q}_k \in \mathbb{F}_2$ to a central receiver with $N_\mathrm{R} \geq K$ antennas. To that end, the symbols are encoded and mapped to complex-valued transmit symbols x_k, drawn from some signal constellation $\mathcal{A}$ with variance σ_x^2.

Denoting the K-dimensional transmit vector as $\boldsymbol{x}$, the $N_\mathrm{R} \times K$ (flat-fading) channel matrix as $\boldsymbol{H}$, and the N_R-dimensional noise vector (with zero-mean Gaussian noise components with variance σ_n^2 per dimension) as $\boldsymbol{n}$, the receive vector $\boldsymbol{y}$ (complex baseband notation) reads as usual

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{n} \,. \tag{1}$$

We assume joint processing of the N_R components of the receive vector $\boldsymbol{y}$. The common, practicable approach is to first perform some form of joint equalization, followed by individual decoding of the codes.

Lattice-reduction-aided and integer-forcing equalization are low-complexity, well-performing approaches. In both variants, the main idea is to factorize the channel matrix as

$$\boldsymbol{H} = \boldsymbol{W}\,\boldsymbol{Z} \,, \tag{2}$$

where $\boldsymbol{Z} \in \mathbb{G}^{K\times K}$, $\mathbb{G} = \mathbb{Z} + \mathrm{j}\mathbb{Z}$, is a full-rank (Gaussian) integer matrix. Then, only the non-integer part $\boldsymbol{W}$ is equalized (here via MMSE linear equalization) [3], [17], [4]. The different factorization criteria (lattice or dual-lattice approach, ZF or MMSE criterion), constraints (unimodular vs. full-rank), and algorithms (shortest basis problem vs. shortest independent vector problem) are irrelevant for the scope of the paper; for an overview see, e.g., [4].

Including the linear (ZF or MMSE) equalizer frontend, the remaining part of the (LRA/IF) receiver has to deal with

$$\boldsymbol{r} = \boldsymbol{Z}\boldsymbol{x} + \bar{\boldsymbol{n}} \stackrel{\mathrm{def}}{=} \bar{\boldsymbol{x}} + \bar{\boldsymbol{n}} \,, \tag{3}$$

where $\bar{\boldsymbol{n}}$ is the effective disturbance after equalization—here, for brevity we assume that all (complex-valued) components have the same variance $\sigma_{\bar{n}}^2$. Fig. 1 shows the effective end-to-end channel model.

The main difference between LRA and IF equalization is how the integer interference is resolved. In LRA schemes, the

[1]Shortly after the submission of this paper we became aware of a similar work [1], however, treating only coding over ASK constellations.

[2]The notation distinguishes quantities over the complex numbers (typeset as $\boldsymbol{x}$, $\boldsymbol{Z}$, …), and over finite fields (typeset in Fraktur font; $\mathfrak{q}$, $\mathfrak{c}$, $\mathfrak{Z}_0$, …).

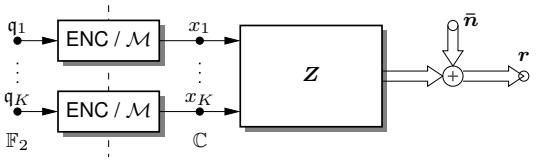

Fig. 1. End-to-end integer channel model (including linear equalization).

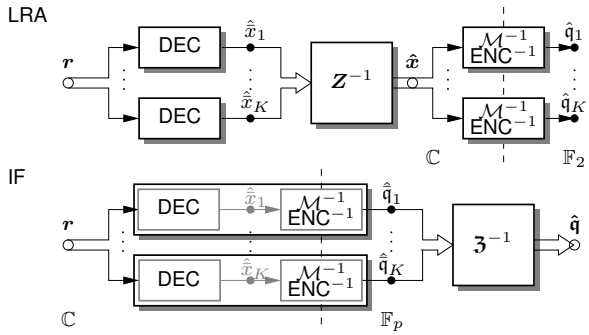

Fig. 2. System model of the receiver over end-to-end integer channels. Top: lattice-reduction-aided equalization; Bottom: integer-forcing receiver.

linear combinations $\bar{\boldsymbol{x}}$ of the transmit symbols in signal space are estimated by the decoders and the interference is undone via $\boldsymbol{Z}^{-1}$ over the complex numbers, cf. Fig. 2, top. Finally, via the encoder inverses, estimates $\hat{\mathsf{q}}_k$ are produced.

In IF schemes (Fig. 2, bottom), the source symbols are drawn from a finite field $\mathbb{F}_p$, p a prime, and encoding is done over $\mathbb{F}_p$. At the receiver, linear combinations $\bar{\mathsf{q}}_k$ of the source symbols are delivered by the decoders and the integer matrix $\boldsymbol{\mathcal{Z}}$ (the $\mathbb{F}_p$-equivalent to $\boldsymbol{Z}$) is inverted over $\mathbb{F}_p$.

These different orders of encoder inverse and inverse of $\boldsymbol{Z}$ impose different constraints on the codes. In LRA schemes, integer linear combinations in signal space have to be decodable. When lattice codes are used, this is directly fulfilled.

III. Multilevel Codes and Integer Interference

A particular strategy to coded modulation is *multilevel coding* [7], [14]. Via a set of *binary component codes* $\mathfrak{C}_l$, $l = 0, \ldots, m-1$, and a mapping from binary address labels of $m = \log_2(M)$ digits to M signal points (constituent constellation), a code in signal space is generated.

A. Mapping

We are interested in one-dimensional (amplitude-shift keying, ASK) and (square) two-dimensional (quadrature-amplitude modulation, QAM) constituent constellations and restrict ourselves to mapping according to the *set partitioning rule* [13]. In both cases, the mapping can be written as [5]

$$\mathcal{M}(\mathfrak{b}_{m-1} \ldots \mathfrak{b}_1 \mathfrak{b}_0) = \mathrm{mod}_B \Big(\sum\nolimits_{l=0}^{m-1} \psi(\mathfrak{b}_l) \phi^l \Big) - O \,, \quad (4)$$

where $\psi(\cdot)$ is the common mapping from the finite field ($\mathbb{F}_2$) elements "0" and "1" to integers "0" and "1", i.e., $\psi(\mathsf{0}) = 0$ and $\psi(\mathsf{1}) = 1$. $B = M$ for ASK and $B = \sqrt{M}$ for QAM defines the boundary region[3] and $O = \frac{M-1}{2}$ for ASK and $O = (1+\mathrm{j})\frac{\sqrt{M}-1}{2}$ for QAM is the offset for zero-mean constellations. Ignoring the modulo reduction and the offset, the point in signal space is given by its binary expansion w.r.t. the *base* ϕ. For ASK we have $\phi = 2$ and the usual binary expansion of an integer is present. For QAM we choose $\phi = -1 + \mathrm{j}$. This is due to the fact that the Gaussian integers $\mathbb{G}$ can be uniquely given via a binary representation where the base (radix) is the complex number $\pm 1 \pm \mathrm{j}$ [11], [6], cf. also [5].

B. Multilevel Codes and Lattices

Using this mapping, the multilevel code is defined by

$$\mathcal{C}_{\mathrm{MLC}} = \mathrm{mod}_B \Big(\sum\nolimits_{l=0}^{m-1} \psi(\mathfrak{C}_l) \phi^l \Big) - O \,, \quad (5)$$

where $\psi(\cdot)$, $\mathrm{mod}(\cdot)$, and the offset O are applied componentwise and $\mathfrak{C}_l$, $l = 0, \ldots, m-1$, are the binary component codes (at level l). Here, to simplify exposition, they are assumed to be block codes of equal length N.

Such a multilevel code is never a lattice. However, we can consider the following construction ("construction by code formula" [9], [5])

$$\mathcal{C} = \sum\nolimits_{l=0}^{m-1} \psi(\mathfrak{C}_l) \phi^l + \phi^m \mathbb{G}^N \,, \quad (6)$$

which differs from $\mathcal{C}_{\mathrm{MLC}}$ by i) eliminating the offset O and ii) inherently assuming an infinite number of "uncoded levels" via the addition of $\phi^m \mathbb{G}^N$ ($\phi^m \mathbb{Z}^N$ in case of ASK), i.e., periodic extension by adding integer multiples of ϕ^m to the coordinates of the codewords (note that $|\phi^m| = B$ is the size of the support per dimension of the multilevel code).

In [9] it is shown that $\mathcal{C}$ is a (real) lattice, i.e., has group structure under ordinary vector addition (over $\mathbb{R}$), if

a) $\mathfrak{C}_l$ are linear codes
b) $\mathfrak{C}_0 \subseteq \mathfrak{C}_1 \subseteq \cdots \subseteq \mathfrak{C}_{m-1}$ (7)
c) for all l and $\mathfrak{c}_i, \mathfrak{c}_j \in \mathfrak{C}_l$ it has to hold $\mathfrak{c}_i \odot \mathfrak{c}_j \in \mathfrak{C}_{l+1}$

where $\odot$ denotes the element-wise multiplication (in the arithmetic of $\mathbb{F}_2$, i.e., AND operation) of the words $\mathfrak{c}_i$ and $\mathfrak{c}_j$. The intuition behind the third demand is that the addition of two mapped words in real space (ASK) can be written as [9] ($\oplus$: element-wise addition in the arithmetic of $\mathbb{F}_2$)

$$\psi(\mathfrak{c}_i) + \psi(\mathfrak{c}_j) = \psi(\mathfrak{c}_i \oplus \mathfrak{c}_j) + 2\psi(\mathfrak{c}_i \odot \mathfrak{c}_j) \,, \quad (8)$$

where $\mathfrak{c}_i \odot \mathfrak{c}_j$ is the "carry" since[4] $1 + 1 = 2 = [\mathsf{1}\,\mathsf{0}]_2$.

For QAM signaling we have to resort to the arithmetic w.r.t. the base $-1 + \mathrm{j}$. Here, $2 = [\mathsf{1}\,\mathsf{1}\,\mathsf{0}\,\mathsf{0}]_{-1+\mathrm{j}}$ which gives

$$\psi(\mathfrak{c}_i) + \psi(\mathfrak{c}_j) = \psi(\mathfrak{c}_i \oplus \mathfrak{c}_j) + (\phi^2 + \phi^3)\psi(\mathfrak{c}_i \odot \mathfrak{c}_j) \,, \quad (9)$$

i.e., the "carry" is 110 [6] and $\mathfrak{c}_i \odot \mathfrak{c}_j \in \mathfrak{C}_{l+2} \subseteq \mathfrak{C}_{l+3}$ is the respective constraint ((7)c) is sufficient but not required).

[3] $\mathrm{mod}_B(x) \in 0, \ldots, B-1$ if x is real-valued; mod is applied separately to real and imaginary part when x is complex-valued.

[4] The sum of two codewords in signal space is only a valid codeword if the sum of two codewords in Hamming space is also in the code (linear codes) and if the "carry" is in the code of the next higher level.

If in ASK only the lowest level is coded (identical to construction A [2]), Constraints b) and c) are automatically fulfilled and a lattice is present. However, the (gross) coding gain is then limited to 6 dB by the first uncoded level. In QAM the levels are spaced only by $|\phi| = \sqrt{2}$ (3 dB) and not 2 as in ASK. Here, no carry to the next level but only the second and third next level is caused. Hence, the lower two levels can be coded and still all constraints are fulfilled and a lattice is present. In this case, the (gross) coding gain is also limited to 6 dB.

C. Multilevel Codes and Integer Linear Combinations

We are now interested in (Gaussian) integer linear combinations of multilevel codewords. To this end, we consider $\mathcal{C}$ from (6), i.e., ignore the boundary and the offset.

Let the MLC codewords $\boldsymbol{c}^{(k)}$ in signal space be obtained via the m codewords $\mathfrak{c}_l^{(k)}$, $l = 0, \ldots, m-1$, $k = 1, 2, \ldots$, in Hamming space, i.e.,

$$\boldsymbol{c}^{(k)} = \sum\nolimits_{l=0}^{m-1} \psi(\mathfrak{c}_l^{(k)}) \phi^l \,. \tag{10}$$

The action of integer linear combinations of these codewords is determined by the binary expansion (w.r.t. the base ϕ, $\mathfrak{z}_i^{(k)} \in \mathbb{F}_2$) of the respective (Gaussian) integer z_k

$$z_k = \left[\ldots \mathfrak{z}_2^{(k)} \mathfrak{z}_1^{(k)} \mathfrak{z}_0^{(k)} \right]_\phi = \sum\nolimits_{i \geq 0} \psi(\mathfrak{z}_i^{(k)}) \, \phi^i \,. \tag{11}$$

Using (10) and (11) we have ($\psi(\mathfrak{a})\psi(\mathfrak{b}) = \psi(\mathfrak{a}\mathfrak{b})$, $\mathfrak{a}, \mathfrak{b} \in \mathbb{F}_2$)

$$\begin{aligned} \sum\nolimits_k z_k \boldsymbol{c}^{(k)} &= \sum\nolimits_k z_k \sum\nolimits_{l=0}^{m-1} \psi(\mathfrak{c}_l^{(k)}) \, \phi^l \\ &= \sum\nolimits_{l=0}^{m-1} \sum\nolimits_{i \geq 0} \sum\nolimits_k \psi(\mathfrak{z}_i^{(k)}) \, \phi^i \, \psi(\mathfrak{c}_l^{(k)}) \, \phi^l \\ &= \sum\nolimits_{l=0}^{m-1} \sum\nolimits_{i \geq 0} \sum\nolimits_k \psi(\mathfrak{z}_i^{(k)} \mathfrak{c}_l^{(k)}) \, \phi^{i+l} \\ &= \sum\nolimits_\ell \psi(\mathfrak{c}_\ell^{\text{eff}}) \, \phi^\ell \,, \end{aligned} \tag{12}$$

where $\mathfrak{c}_\ell^{\text{eff}}$ is the effective codeword at level ℓ. These words can be calculated by applying (8) or (9) (ASK or QAM) recursively. For example, for two codewords and QAM signaling we have

$$\begin{aligned} \mathfrak{c}_0^{\text{eff}} &= \mathfrak{z}_0^{(1)} \mathfrak{c}_0^{(1)} \oplus \mathfrak{z}_0^{(2)} \mathfrak{c}_0^{(2)} \\ \mathfrak{c}_1^{\text{eff}} &= \mathfrak{z}_0^{(1)} \mathfrak{c}_1^{(1)} \oplus \mathfrak{z}_0^{(2)} \mathfrak{c}_1^{(2)} \oplus \mathfrak{z}_1^{(1)} \mathfrak{c}_0^{(1)} \oplus \mathfrak{z}_1^{(2)} \mathfrak{c}_0^{(2)} \\ \mathfrak{c}_2^{\text{eff}} &= \mathfrak{z}_0^{(1)} \mathfrak{c}_2^{(1)} \oplus \mathfrak{z}_0^{(2)} \mathfrak{c}_2^{(2)} \oplus \mathfrak{z}_1^{(1)} \mathfrak{c}_1^{(1)} \oplus \mathfrak{z}_1^{(2)} \mathfrak{c}_1^{(2)} \\ &\quad \vdots \qquad \oplus \mathfrak{z}_2^{(1)} \mathfrak{c}_0^{(1)} \oplus \mathfrak{z}_2^{(2)} \mathfrak{c}_0^{(2)} \;\oplus\; \mathfrak{z}_0^{(1)} \mathfrak{c}_0^{(1)} \odot \mathfrak{z}_0^{(2)} \mathfrak{c}_0^{(2)} \end{aligned} \tag{13}$$

Having a look at the appearing terms and keeping in mind that $\mathfrak{c}_\ell^{\text{eff}}$ has to be a valid codeword of code $\mathfrak{C}_\ell$ the three above demands (7) on the codes are directly clear.

D. Multistage Decoding

Usually, multilevel codes are decoded via *multistage decoding (MSD)*, cf. Alg. 1. Thereby, given $\boldsymbol{r} = \boldsymbol{c} + \boldsymbol{n}$ with $\boldsymbol{c} = \sum_{l=0}^{m-1} \psi(\mathfrak{c}_l)\phi^l \in \mathcal{C}$, one component code (estimate $\hat{\mathfrak{c}}_l$) after the other is decoded, taking into account the decoding results of lower levels but ignoring the codes of higher levels (i.e., treating them as uncoded). Each decoding step is hence identical to decoding a lattice constructed via lattice construction A [2] (multilevel code where only the lowest level is coded). Thus, in a multistage decoder it is immaterial whether the entire code forms (a translate of a subset of) a lattice.

Alg. 1 Multistage Decoding.

function $\hat{\boldsymbol{c}} = \texttt{MSD}(\boldsymbol{r})$

1 $\ell = 0$; $\boldsymbol{r}_\ell = \boldsymbol{r}$ // init
2 `while` $\ell < m$ {
3 $\quad \hat{\mathfrak{c}}_\ell = \text{DEC}_{\mathfrak{C}_\ell}\{\boldsymbol{r}_\ell\}$ // decode level ℓ
4 $\quad \boldsymbol{r}_{\ell+1} = (\boldsymbol{r}_\ell - \psi(\hat{\mathfrak{c}}_\ell))/\phi$ // eliminate known level
5 $\quad \ell = \ell + 1$ }
6 $\hat{\boldsymbol{c}} = \sum_{l=0}^{m-1} \psi(\hat{\mathfrak{c}}_l)\phi^l$ // codeword estimate

However, it is essential that the effective word at level ℓ is a valid codeword from $\mathfrak{C}_\ell$. If integer linear combinations of codewords are to be decoded and Constraint c) of (7) is violated, carries from lower levels destroy this property and MSD does not work any more.

E. Carry Correction

In the following, we assume that Constraints a) and b) on the component codes are fulfilled but Constraint c) is violated. We present a generalization of multistage decoding which decodes the K linear combinations and eliminates the effect of the carries.

Let $\boldsymbol{c}^{(1)}$, $\boldsymbol{c}^{(2)}$, …, $\boldsymbol{c}^{(K)}$ be the codewords of the K users in signal space (component codewords $\mathfrak{c}_0^{(k)}$, $\mathfrak{c}_1^{(k)}$, …, $\mathfrak{c}_{m-1}^{(k)}$, $k = 1, \ldots, K$, in Hamming space) and $\boldsymbol{Z}$ the integer matrix. Ignoring the noise for the moment we have

$$\begin{bmatrix} \boldsymbol{r}_1 \\ \vdots \\ \boldsymbol{r}_K \end{bmatrix} = \boldsymbol{Z} \begin{bmatrix} \boldsymbol{c}^{(1)} \\ \vdots \\ \boldsymbol{c}^{(K)} \end{bmatrix} . \tag{14}$$

From the discussion above (cf. (12) and (13)), we see that the effective codewords at level $l = 0$ are obtained from the component codewords at this levels via

$$\begin{bmatrix} \mathfrak{c}_{1,0}^{\text{eff}} \\ \vdots \\ \mathfrak{c}_{K,0}^{\text{eff}} \end{bmatrix} = \boldsymbol{\mathfrak{Z}}_0 \begin{bmatrix} \mathfrak{c}_0^{(1)} \\ \vdots \\ \mathfrak{c}_0^{(K)} \end{bmatrix} \tag{15}$$

where $\mathfrak{z}_0^{(i,j)}$ is the least significant bit (LSB) of $z_{i,j}$ w.r.t. to the basis ϕ and $\boldsymbol{\mathfrak{Z}}_0 = [\mathfrak{z}_0^{(i,j)}]$. Having estimates for $\mathfrak{c}_{k,0}^{\text{eff}}$, $k = 1, \ldots, K$, the original codewords at level 0 can hence be obtained by solving (over $\mathbb{F}_2$) the set of linear equations (15).

Once the codewords at level 0 are known, the contributions (over $\mathbb{C}$) of the superposition of these levels into the higher levels (carries) can be calculated via[5]

$$\begin{bmatrix} \boldsymbol{s}_1 \\ \vdots \\ \boldsymbol{s}_K \end{bmatrix} = \mathcal{Q}_\phi \left\{ \boldsymbol{Z} \begin{bmatrix} \psi(\mathfrak{c}_0^{(1)}) \\ \vdots \\ \psi(\mathfrak{c}_0^{(K)}) \end{bmatrix} \right\} . \tag{16}$$

[5] $\mathcal{Q}_\phi\{z\}$ denotes the quantization operation which nulls the LSB in the binary expansion w.r.t. ϕ, i.e., with $z = [\ldots \mathfrak{z}_2 \mathfrak{z}_1 \mathfrak{z}_0]_\phi$ we obtain $\mathcal{Q}_\phi\{z\} = [\ldots \mathfrak{z}_2 \mathfrak{z}_1 \, 0]_\phi$. This nulling is required since only the contribution to the higher levels but not the current level is of interest.

Alg. 2 Multistage Decoding with Carry Correction.

function $[\hat{\boldsymbol{c}}^{(1)}, \ldots, \hat{\boldsymbol{c}}^{(K)}] = \texttt{MSD}(\boldsymbol{r})$

1 $\ell = 0$; $\boldsymbol{r}_{k,\ell} = \boldsymbol{r}_k$, $k = 1, \ldots, K$ // init
2 `while` $\ell < m$ {
3 $\hat{\mathfrak{c}}_{k,\ell} = \mathrm{DEC}_{\mathfrak{C}_\ell}\left\{\boldsymbol{r}_{k,\ell}\right\}$, $k = 1, \ldots, K$ // decode level ℓ
4 solve (15) and calculate $\boldsymbol{s}_k$ via (16) // calculate carries
5 $\boldsymbol{r}_{k,\ell+1} = (\boldsymbol{r}_{k,\ell} - \psi(\hat{\mathfrak{c}}_{k,\ell}) - \boldsymbol{s}_{k,\ell})/\phi$ // eliminate known interference
6 $\ell = \ell + 1$ }
7 $\hat{\boldsymbol{c}}^{(k)} = \sum_{l=0}^{m-1} \psi(\hat{\mathfrak{c}}_{k,l})\phi^l$ // codeword estimates

This known "interference" caused by the codewords at level 0 can now be eliminated. Hence, in a generalized version of MSD, i) the influence of the effective codeword at level 0 is eliminated by subtraction of $\psi(\mathfrak{c}_{i,0}^{\mathrm{eff}})$ and ii) the influence (carries) of the linear combinations of the codewords onto higher levels is eliminated by subtraction of $\boldsymbol{s}_k$. This procedure is iterated over the levels. In Alg. 2, a pseudo-code description of this generalized version of multistage decoding is given.

The additional complexity for the calculation of the correction terms $\boldsymbol{s}_k$ is negligible compared to the decoding operations.

F. Conditions on $\boldsymbol{Z}$

The final question is under which conditions does the carry correction work. Since the effective codewords are generated via the binary matrix $\boldsymbol{\mathfrak{Z}}_0$ (the matrix of LSBs of the (Gaussian) integer entries of the matrix $\boldsymbol{Z}$), this matrix has to have full rank.[6]

In order that $\boldsymbol{\mathfrak{Z}}_0$ has full rank, its determinant (over $\mathbb{F}_2$) has to be non-zero. We have $\psi(\det(\boldsymbol{\mathfrak{Z}}_0)) = \mathrm{mod}_2(\det(\psi(\boldsymbol{\mathfrak{Z}}_0)))$ and using the Leibniz formula for the expansion of the determinant, we obtain $\det(\psi(\boldsymbol{\mathfrak{Z}}_0)) = \det(\boldsymbol{Z}) + \phi\mathbb{G}$.

If $\det(\boldsymbol{Z}) \in \mathbb{G}$ is "odd" (LSB w.r.t. base ϕ equal to one) then $\det(\psi(\boldsymbol{\mathfrak{Z}}_0))$ is also odd and hence non-zero. Unimodular matrices have an "odd" determinant. If $\det(\boldsymbol{Z})$ is "even" (LSB equal to zero) then $\det(\psi(\boldsymbol{\mathfrak{Z}}_0))$ is also even and thus $\det(\boldsymbol{\mathfrak{Z}}_0) = 0$ and $\boldsymbol{\mathfrak{Z}}_0$ is not invertible.

Thus it is proven that for all matrices for which $\det(\boldsymbol{Z})$ is "odd" ($\in 1 + \phi\mathbb{G}$), hence in particular for unimodular matrices, carry correction works. This imposes restrictions on $\boldsymbol{Z}$. However, it has been shown that using a suited lattice reduction algorithm (Minkowski reduction) for i.i.d. Gaussian channel matrices even the restriction to unimodular matrices causes almost no loss compared to the set of full-rank matrices [12].

IV. Numerical Results

In oder to study the effect of integer linear combinations and carry correction, numerical simulations have been conducted. We assume a 16 QAM constellation; the code has four levels.

First, a "toy example" is presented. Linear block codes of length $N = 13$ are employed, the generator matrices of the

[6] If $\boldsymbol{\mathfrak{Z}}_0$ has columns which are the all-zero vector, the corresponding codewords do not influence the effective codewords and do not contribute to carry generation. Hence, only the relevant part (matrix $\boldsymbol{\mathfrak{Z}}_0$ with all-zero columns removed) has to have full rank.

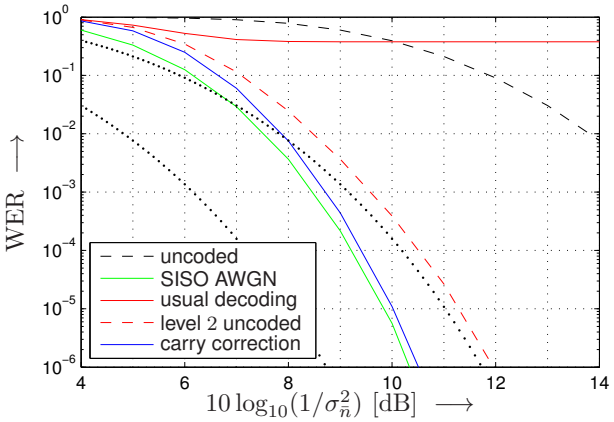

Fig. 3. Word error rate over the inverse noise power (in dB). 16 QAM. Component codes: linear block codes of length $N = 13$. Rate $31/13$ bit/symbol. $K = 3$ users. Fixed integer matrix $\boldsymbol{Z} = [2{-}2\mathrm{j}\ \ 1\ \ {-}2{-}\mathrm{j};\ \ 1\ \ {-}1{-}\mathrm{j}\ \ 0;\ {-}1\ \ \mathrm{j}\ \ 0]$; $\det(\boldsymbol{Z}) = 2 + \mathrm{j}$. Dotted: asymptotic behavior.

$(13, 2, 8)$ code at level 0 and that of the $(13, 4, 4)$ code level 1 are

$$\mathfrak{G}_0 = \begin{bmatrix} 1111111100000 \\ 0000011111111 \end{bmatrix}, \quad \mathfrak{G}_1 = \begin{bmatrix} 1111111100000 \\ 1100000111111 \\ 0011111100011 \\ 0000011111111 \end{bmatrix}. \quad (17)$$

At level 2 a $(13, 12, 2)$ single-parity-check code is used and level 3 is uncoded ($(13, 13, 1)$ code). This construction does not fulfill Condition c) of (7), hence no lattice is present and carry correction is required for correct decoding.

Fig. 3 shows the results (word error rate averaged over the users) for $K = 3$ and the arbitrarily selected integer matrix $\boldsymbol{Z} = [2{-}2\mathrm{j}\ \ 1\ \ {-}2{-}\mathrm{j};\ 1\ \ {-}1{-}\mathrm{j}\ \ 0;\ \ {-}1\ \ \mathrm{j}\ \ 0]$ with $\det(\boldsymbol{Z}) = 2 + \mathrm{j}$. Uncoded transmission and coded transmission over the single-input/single-output (SISO) AWGN channel are shown for reference. The code has an asymptotic (gross) coding gain of 9 dB (dotted; curve for uncoded transmission shifted by the asymptotic gain). If no carry correction is applied, decoding of the multilevel code fails completely.[7] With the proposed carry correction the performance of the code over the SISO channel can almost be achieved (some error multiplication is present). If level 2 is (treated as) uncoded, Constraint c) is fulfilled and no carry correction is required. However, the (asymptotic, gross) coding gain is then limited to 6 dB. Via carry correction we can break this 6 dB barrier.

Next, a multilevel code with low-density parity-check (LDPC) codes as component codes is considered. The rates of the codes are adjusted according to the capacity design rule for multilevel codes [14]. For a target rate of 3 bits per QAM symbol, the rates of the component codes have to be selected as $R_0/R_1/R_2/R_3 = .282/.753/.964/1$. We use a code length $N = 5000$, which gives code dimensions $\kappa_0/\kappa_1/\kappa_2/\kappa_3 = 1412/3766/4820/5000$.

[7] Note that the component decoders always return a valid codeword. Since, due to the carry, the effective word at level 2 is not a valid codeword, wrong correction occurs, leading to a high floor in the error rate.

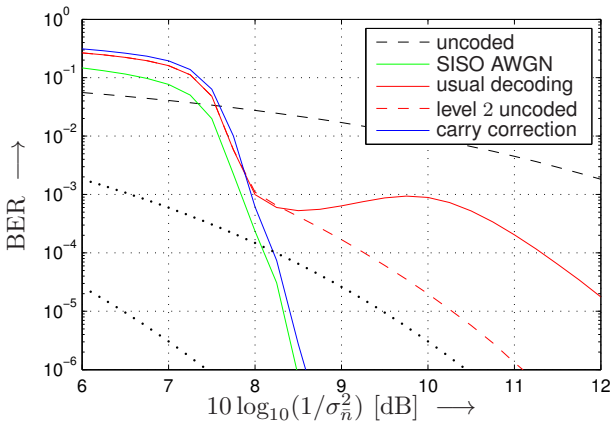

Fig. 4. Bit error rate over the inverse noise power (in dB). 16 QAM. Component codes: LDPC codes of length $N = 5000$. Rate 3 bit/symbol. $K = 3$ users. Fixed integer matrix $\boldsymbol{Z} = [1 \;\; \mathrm{j} \;\; -1{+}\mathrm{j}; \;\; 1 \;\; 0 \;\; -2\mathrm{j}; \;\; 2\mathrm{j} \;\; -1 \;\; -\mathrm{j}]$; $\det(\boldsymbol{Z}) = \mathrm{j}$. Dotted: asymptotic behavior.

As LDPC codes *irregular repeat-accumulate codes* [8] are used. The required subcode property $\mathfrak{C}_0 \subset \mathfrak{C}_1 \subset \mathfrak{C}_2 \subset \mathfrak{C}_3 = \mathbb{F}_2^N$ is guaranteed by constructing the parity-check matrices $\mathfrak{H}_l$ in such a way that $\mathfrak{H}_{l+1}$ is a subset of the rows of $\mathfrak{H}_l$. To that end, first the $(N - \kappa_2) \times N$ matrix $\mathfrak{H}_2$ is constructed with an ultra-sparse left part (column weight 2) and a right staircase part. This matrix is extended to $\mathfrak{H}_1$ by adding $\kappa_2 - \kappa_1$ rows in such a way that the newly added left part is ultra-sparse and that the staircase construction is seamless continued in the right part. Given $\mathfrak{H}_1$, the final parity-check matrix $\mathfrak{H}_0$ is constructed in the same way by adding $\kappa_1 - \kappa_0$ rows. Given the parity-check matrices $\mathfrak{H}_l$, generator matrices $\mathfrak{G}_l$ for systematic encoding are calculated. Noteworthy, this construction guarantees the subcode property (Constraint b)) of the linear (Constraint a)) component codes. However, Constraint c) is neither taken into account nor fulfilled.

Fig. 4 shows the error rate of the information bits over the noise level (in dB).[8] Here, $\boldsymbol{Z} = [1 \;\; \mathrm{j} \;\; -1{+}\mathrm{j}; \;\; 1 \;\; 0 \;\; -2\mathrm{j}; \;\; 2\mathrm{j} \;\; -1 \;\; -\mathrm{j}]$ with $\det(\boldsymbol{Z}) = \mathrm{j}$ is chosen randomly. Message-passing decoders using log-likelihood ratios based on nearest-neighbor approximation are used; at maximum 10 iterations are performed. Basically, the same behavior as in the example above is visible. The exact shape of the flattening in case of conventional (independent) multistage decoding depends on the operation of the decoder in case of non-converges (here: the current variable-node values are quantized and output). If level 2 is treaded as uncoded, the (gross) coding gain is again limited to 6 dB. Via the proposed generalized decoding scheme, a performance extremely close to that of the code over the SISO AWGN channel can be achieved.

V. Summary and Conclusions

In this paper, we have studied the application of multilevel codes in LRA equalization, i.e., the decoding of multilevel codes when (Gaussian) integer linear combinations of the codewords are present. A generalized version of multistage decoding incorporating a carry correction has been proposed which has only marginal additional complexity compared to independent decoding. Thereby, the constraints on the component codes are significantly relaxed and no lattice code has to be generated. Numerical results for arbitrary but fixed integer matrices have been given.

The next step is the performance evaluation over random (i.i.d. Gaussian) channel matrices. The factorization algorithm in [4] can straightforwardly be generalized such that only $\boldsymbol{Z}$ matrices with "odd" determinant are returned. However, almost no loss occurs if Minkowski reduction with its restriction to unimodular matrices is used. In summary, via the new decoding procedure, a simple combination of multilevel codes with low-complexity LRA equalization is enabled.

[8] Please note the different scaling of the x-axis compared to Fig. 3.

References

[1] S H. Chae, M. Jang, S.K. Ahn, J. Park, C. Jeong. Multilevel Coding Scheme for Integer-Forcing MIMO Receivers With Binary Codes. *IEEE Trans. Wireless Comm.*, vol. 16, no. 8, pp. 5428–5441, Aug. 2017.

[2] J.H. Conway, N.J.A. Sloane. *Sphere Packings, Lattices and Groups*. Springer Verlag, New York, Berlin, 3rd edition, 1999.

[3] R.F.H. Fischer, C. Windpassinger, C. Stierstorfer, C. Siegl, A. Schenk, Ü. Abay. Lattice-Reduction-Aided MMSE Equalization and the Successive Estimation of Correlated Data. *AEÜ—Int. Journal of Electronics and Communications*, vol. 65, no. 8, pp. 688–693, Aug. 2011.

[4] R.F.H. Fischer, M. Cyran, S. Stern. Factorization Approaches in Lattice-Reduction-Aided and Integer-Forcing Equalization. In *2016 Int. Zurich Seminar on Communications*, Zurich, Switzerland, March 2016.

[5] G.D. Forney. Coset Codes. I. Introduction and Geometrical Classification. *IEEE Trans. Information Theory*, vol. 34, no. 5, pp. 1123–1151, Sep. 1988.

[6] W.J. Gilbert. Arithmetic in Complex Bases. *Mathematics Magazine*, vol. 57, No. 2, pp. 77–81, March 1984.

[7] H. Imai and S. Hirakawa. A New Multilevel Coding Method Using Error Correcting Codes. *IEEE Trans. Information Theory*, vol. 23, no. 3, pp. 371–377, May 1977.

[8] H. Jin, A. Khandekar, R. McEliece. Irregular Repeat-Accumulate Codes. In *2nd Int. Symp. on Turbo Codes and Rel. Topics*, Brest, France, 2000.

[9] W. Kositwattanarerk, F. Oggier. Connections Between Construction D and Related Constructions of Lattices. *Designs, Codes and Cryptography*, vol. 73, no. 2, pp. 441–455, Nov. 2014.

[10] B. Nazer, M. Gastpar. Compute-and-Forward: Harnessing Interference Through Structured Codes. *IEEE Trans. Information Theory*, vol. 57, no. 10, pp. 6463–6486, Oct. 2011.

[11] W. Penney. A "Binary" System for Complex Numbers. *Journal of the Association for Computing Machinery (JACM)*, vol. 12, no. 2, pp. 247–248, April 1965.

[12] S. Stern, R.F.H. Fischer Optimal Factorization in Lattice-Reduction-Aided and Integer-Forcing Linear Equalization. In *11. Int. ITG Conf. on Systems, Communications, and Coding (SCC)*, Hamburg, Germany, February 2017

[13] G. Ungerböck. Channel Coding with Multilevel/Phase Signals. *IEEE Trans. Information Theory*, vol. 28, no. 1, pp. 55–67, Jan. 1982.

[14] U. Wachsmann, R.F.H. Fischer, J.B. Huber. Multilevel Codes: Theoretical Concepts and Practical Design Rules. *IEEE Trans. Information Theory*, vol. 45, no. 5, pp. 1361–1391, Jul 1999.

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

[16] H. Yao, G. Wornell. Lattice-Reduction-Aided Detectors for MIMO Communication Systems. In *IEEE Global Telecommunications Conference*, pp. 424–428, Taipei, Taiwan, Nov. 2002.

[17] J. Zhan, B. Nazer, U. Erez, M. Gastpar. Integer-Forcing Linear Receivers. *IEEE Trans. Information Theory*, vol. 60, no. 12, pp. 7661–7685, Dec. 2014.