Practical Encoders and Decoders for Euclidean Codes from Barnes-Wall Lattices

J. Harshan, *Member, IEEE*, E. Viterbo, *Fellow, IEEE*, and Jean-Claude Belfiore, *Fellow, IEEE*

***Abstract*—We address the application of Barnes-Wall (BW) lattice codes for communication over additive white Gaussian noise (AWGN) channels. We introduce Construction A' of complex BW lattices that makes new connection between linear codes over polynomial rings and lattices. We show that Construction A' of BW lattices is equivalent to the multilevel construction from Reed-Muller codes proposed by Forney. To decode the BW lattice code, we adapt the low-complexity sequential BW lattice decoder (SBWD) proposed by Micciancio and Nicolosi. First we study the error performance of SBWD for decoding the infinite lattice, and demonstrate that it is powerful in making correct decisions well beyond the packing radius. Subsequently, we use the SBWD to decode lattice codes through a novel noise trimming technique, where the received vector is appropriately scaled before applying the SBWD. We show that the noise trimming technique is most effective for decoding BW lattice codes in smaller dimensions, while the gain diminishes for decoding codes in larger dimensions.**

***Index Terms*—Barnes-Wall lattices, lattice codes, low-complexity lattice decoders.**

I. Introduction

EVER since random coding schemes were demonstrated to approach the capacity of additive white Gaussian noise (AWGN) channels [1], enormous research has taken place to find *structured* coding schemes which can accomplish the same job. The need for structured codes is to facilitate simpler analysis of the code performance and to achieve reduced complexity in encoding and decoding operations. A well known method to obtain such codes is to carve out a finite set of points from special structures in Euclidean space called *lattices* [2]-[6]. These codes are referred to as lattice codes, and are usually obtained as a set of coset representatives of a suitable quotient lattice. Importantly, lattice codes have the advantage of inheriting most of the code properties from the parent lattice, and as a result, the choice of the lattice is crucial to the performance of the code.

Manuscript received January 7, 2013; revised May 28 and August 5, 2013. The editor coordinating the review of this paper and approving it for publication was L. Dolecek.

J. Harshan and E. Viterbo are with the Department of Electrical and Computer Systems Engineering, Monash University, Melbourne, Australia-3800 (e-mail: {harshan.jagadeesh, emanuele.viterbo}@monash.edu).

J.-C. Belfiore is with the Department of Communications and Electronics, Telecom ParisTech, Paris, France (e-mail: belfiore@enst.fr).

Parts of this work are in the Proceedings of IEEE International Symposium on Information Theory (ISIT) 2012, held at Cambridge, MA, USA, and the International Symposium on Mathematical Theory of Networks and Systems (MTNS) 2012, held at Melbourne, Australia. This work was supported by the Monash Professional Fellowship 2012-2013 and by NPRP grant NPRP5-597-2-241 from the Qatar National Research Fund (a member of Qatar Foundation).

Digital Object Identifier 10.1109/TCOMM.2013.100213.130018

A. Motivation and contributions

In this paper we are interested in carving lattice codes from Barnes-Wall (BW) lattices [7], [8]. Specifically, we choose codes from complex BW lattices as (*i*) efficient low-complexity decoders for such lattices are readily available in [9], [10], and (*ii*) complex BW lattice codes with hypercube shaping are nothing but codes over quadrature amplitude modulation (QAM), and hence, are readily applicable in practice.

In [9] two low-complexity implementations of the bounded distance decoder for BW lattices have been proposed, namely (*i*) the sequential bounded distance decoder, and (*ii*) the parallel bounded distance decoder. Inspired by the parallel bounded distance decoder in [9], list decoders based on parallel implementation have been proposed in [10]. We point out that the parallel decoders of [9] and [10] have low-complexity only when implemented on a sufficiently large number of parallel processors. If the above decoders are implemented on a single processor, then the complexity advantages are lost, and specifically, the complexity of the list decoder grows larger than that of the sequential decoder in [9]. We are interested in lattice codes of large block lengths, and hence, we focus on the sequential bounded distance decoder which seems more suitable for practical implementation. The sequential decoder in [9] was proven to correct any error up to the packing radius. However, the possibility of a correct decision is not known when the received vector falls outside the bounded decoding ball of packing radius. In a nutshell, the exact error performance of the decoder is not known. The existence of this low-complexity decoder has motivated us to study its error performance, and use it to decode BW lattice codes. We refer to this decoder as the sequential BW lattice decoder (SBWD). This work stems from the preliminary results available in [11], [12]. The contribution of this paper on encoding and decoding of complex BW lattices are given below.

1) We introduce Construction A' of BW lattices which enables us to generate them from linear codes over *polynomial rings*. The proposed method is yet another construction of BW lattices and makes a new connection between codes over polynomial rings and lattices. We show that Construction A' is equivalent to the multilevel construction of BW lattices from Reed-Muller codes.
2) We study the error performance of the SBWD in AWGN channels. Since the SBWD exploits the multilevel construction from Reed-Muller (RM) codes, we study the error performance of the soft-input RM decoders as used in the SBWD. First, we use the Jacobi-Theta functions [13] to characterize the virtual binary channels that arise in the decoding process. Subsequently, we study the

noise statistics in the algorithm, and provide an upper bound on the error performance of the soft-input RM decoders. Through computer simulations, we showcase the error performance of the SBWD, and highlight that the decoder is powerful in making correct decisions well beyond the packing radius.

3) To decode the lattice code in AWGN channels, we adapt the SBWD along with a noise trimming technique, wherein the components of the received vector are appropriately scaled before passing them to the SBWD. With the noise trimming technique, the SBWD is forced to decode to a codeword which in turn improves the error performance. We refer to this decoder as the *BW lattice code decoder* (BWCD). We obtain the bit error rate (BER) of the BWCD for codes in complex dimensions $4, 16, 64, 256$, and 1024, and show that the BWCD outperforms the SBWD by 0.5 dB in smaller dimensions. We also show that the gains of the noise trimming technique diminishes with larger dimensions.

B. Prior work on Barnes-Wall lattices

BW lattices [7] is a special family of N-dimensional lattices that exists when N is a power of 2. These lattices were originally discovered as a solution to finding extreme quadratic forms in 1959. In 1983 the now well known connection between BW lattices and Reed-Muller codes was discovered in [14]. This connection can be found in [15], [16], [19] in different forms. Apart from the connection to classical linear codes, generator matrices of BW lattice can also be obtained though recursive Kronecker operation on the kernels [17], [9]

$$\begin{bmatrix} 1 & 1 \\ 0 & \sqrt{2} \end{bmatrix} \text{and} \begin{bmatrix} 1 & 1 \\ 0 & 1+i \end{bmatrix},$$

for real and complex lattices, respectively (where $i = \sqrt{-1}$).

In 1989 G.D. Forney has proposed a low-complexity bounded distance decoding algorithm for Leech lattices [22]. As a generalization, in the same paper, a similar algorithm has been shown to work in decoding all Construction D lattices. As BW lattices can be obtained through Construction D [15], bounded distance decoders for BW lattices were known in principle since [22]. In the 1990's, explicit bounded distance decoders for BW lattices were implemented for dimension up to 32, and numerical results on the error performance were reported [21], [23], [24]. In 2008, Micciancio and Nicolosi [9] have proposed two low-complexity implementations of the bounded distance decoder for BW lattices, namely (*i*) the sequential bounded distance decoder, and (*ii*) the parallel bounded distance decoder. If $N = 2^m$ denotes the dimension of a complex BW lattice, then the worst-case complexity of the decoders has been shown as $O\left(N\log^2(N)\right)$ and $O\left(\log^2(N)\right)$ for the fully sequential decoder and the fully parallel decoder, respectively. For the fully sequential decoder, the algorithm is implemented on a single processor, whereas for the fully parallel decoder, the algorithm is implemented on N^2 parallel processors. Inspired by the fully parallel implementation in [9], list decoder for BW lattices has been recently proposed in [10] where the decoder outputs a list of BW lattice points within any given radius from the target vector. The complexity of the list decoder is polynomial in the dimension of the lattice, and polynomial in the list size, which is a function of the Euclidean radius. Note that the SBWD exploits the multilevel construction of BW lattices from Reed-Muller (RM) codes. On the other hand, the list decoder does not exploit the multilevel construction of BW lattices, and hence, does not need the support of RM decoders.

The rest of this paper is organized as follows: In Section II, we provide a short background on BW lattice encoders from linear codes. In Section III, we introduce Construction A' of complex BW lattices. In Section IV, we study the error performance of the SBWD, while in Section V and Section VI, we use the SBWD to decode the BW lattice code. Finally, in Section VII, we conclude this paper and provide some directions for future work.

Notations: Throughout the paper, boldface letters and capital boldface letters are used to represent vectors and matrices, respectively. For a complex matrix $\mathbf{X}$, the matrices $\mathbf{X}^T$, $\Re(\mathbf{X})$ and $\Im(\mathbf{X})$ denote the transpose, real part, and imaginary part of $\mathbf{X}$, respectively. The set of integers, real numbers, and complex numbers are denoted by $\mathbb{Z}$, $\mathbb{R}$, and $\mathbb{C}$, respectively. We use i to represent $\sqrt{-1}$. For an n-length vector $\mathbf{x}$, we use x_j to represent the j-th component of $\mathbf{x}$. Cardinality of a set $\mathcal{S}$ is denoted by $|\mathcal{S}|$. Magnitude of a complex number x is denoted by $|x|$. The number of ways of picking n out of m objects is denoted by C_n^m. The symbol $\lceil\cdot\rfloor$ denotes the nearest integer of a real number, and we set $\lceil a + 0.5\rfloor = a$ for any $a \in \mathbb{Z}$. Finally, we use $\Pr(\cdot)$ to denote the probability operator.

II. Background on BW lattice construction from Reed-Muller codes

A complex lattice Λ over $\mathbb{Z}[i]$ is a discrete subgroup of $\mathbb{C}^n$ [15]. Alternatively, Λ is a $\mathbb{Z}[i]$-module generated by a basis set $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n \mid \mathbf{v}_j \in \mathbb{C}^n\}$ as $\Lambda = \left\{\sum_{j=1}^n q_j\mathbf{v}_j \mid \forall q_j \in \mathbb{Z}[i]\right\}$. It is well known that dense lattices can be obtained via binary linear codes [15]. Depending on the structure of the underlying linear codes, lattice construction can be categorized into different types e.g., Construction A [18], Construction B, and Construction D [15]. In this section, we recall the well known construction of complex BW lattices from Reed-Muller codes [15], [20].

Construction D:

If $\{\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_{k_r} \mid \mathbf{g}_j \in \mathbb{F}_2^{2^m}\}$ denotes a basis set of the r-th order binary RM code $\mathcal{RM}(r, m)$ for $0 \le r \le m-1$, then the complex BW lattice BW_{2^m} of dimension 2^m can be obtained using Construction D [9], [16] as

$$\left\{(1+i)^m\mathbb{Z}[i]^{2^m} + \sum_{r=0}^{m-1}\sum_{j=1}^{k_r}(1+i)^r a_{r,j}\mathbf{g}_j \mid \forall a_{r,j} \in \{0,1\}\right\}.$$

Complex code formula from Reed-Muller codes

A complex BW lattice can also be obtained from nested Reed-Muller codes using the complex code formula [16], [19] as given in (1), where $\psi : \mathbb{F}_2 \to \mathbb{Z}[i]$ is given by $\psi(0) = 0$ and $\psi(1) = 1$ on the alphabet of $\mathcal{C}$, where $\mathbb{F}_2 = \{0, 1\}$. For

$$BW_{2^m} = \left\{ (1+i)^m \mathbf{a} + \sum_{r=0}^{m-1} (1+i)^r \psi(\mathbf{c}_r) \mid \forall \mathbf{c}_r \in \mathcal{RM}(r,m), \forall \mathbf{a} \in \mathbb{Z}[i]^{2^m} \right\} \quad (1)$$

notational convenience, we also write (1) as

$$BW_{2^m} = (1+i)^m \mathbb{Z}[i]^{2^m} + \sum_{r=0}^{m-1} (1+i)^r \mathcal{RM}(r,m). \quad (2)$$

Although Construction D and the complex code formula look similar, there is a significant difference between them. Construction D generates a lattice with arbitrary nested linear codes, while the complex code formula does not generate lattices for arbitrary nested linear codes. A recent study has shown that the nested linear codes must satisfy an additional property of closeness under Schur's product to generate a lattice via complex code formula [27]. Nested RM codes is one such example, and hence, complex code formula generates BW lattice.

Motivation for Construction A':

In this work we facilitate one-shot encoding of all RM codewords by using a single linear code over polynomial rings, and then obtain the lattice code

$$\mathcal{EC}_{2^m} = \left\{ \sum_{r=0}^{m-1} (1+i)^r \psi(\mathbf{c}_r) \mid \forall \mathbf{c}_r \in \mathcal{RM}(r,m) \right\} \quad (3)$$

as embedding of the codewords of the linear code into the Euclidean space. Since we obtain the lattice code $\mathcal{EC}_{2^m}$ as an embedding of single code, our construction resembles Construction A [15], and hence, we refer to it as Construction A'.

III. Construction A' of BW lattice

To introduce Construction A', we first define polynomial rings and codes over polynomial rings.

Definition 1: We define the polynomial quotient ring $\mathcal{U}_m = \mathbb{F}_2[u]/u^m$ in variable u for any $m \geq 1$ as

$$\mathcal{U}_m = \left\{ \sum_{k=0}^{m-1} b_k u^k \mid b_k \in \mathbb{F}_2 \right\},$$

with regular polynomial addition and multiplication over $\mathbb{F}_2$ coefficients along with the quotient operation $u^m = 0$, which is equivalent to cancelling all the terms of degree greater than or equal to m.

Definition 2: A linear code $\mathcal{C}$ over $\mathcal{U}_m$ is a subset of $\mathcal{U}_m^n$ which can be obtained through a generator matrix $\mathbf{G} \in \mathcal{U}_m^{k \times n}$ as

$$\mathcal{C} = \left\{ \mathbf{zG} \mid \forall \mathbf{z} \in \mathcal{U}_m^k \right\},$$

for some $k \leq n$ and the matrix multiplication is over the ring $\mathcal{U}_m$.

We now introduce Construction A' of BW lattices in the following definition.

Definition 3: A complex BW lattice BW_{2^m} is obtained by Construction A' from a linear code $\mathcal{C}$ over $\mathcal{U}_m$ for $m \geq 1$ if BW_{2^m} can be written as

$$BW_{2^m} = (1+i)^m \mathbb{Z}[i]^n + \mathcal{EC}, \quad (4)$$

where $\mathcal{EC} = \{\Phi(\mathbf{c}) \mid \forall \mathbf{c} \in \mathcal{C}\} \subseteq \mathbb{Z}[i]^n$ is a lattice code obtained from the linear code $\mathcal{C}$ through the mapping $\Phi : \mathcal{U}_m \to \mathbb{Z}[i]$ given by

$$\Phi\left(\sum_{j=0}^{m-1} b_j u^j \right) = \sum_{j=0}^{m-1} \psi(b_j) \left(\Phi(u)\right)^j,$$

where $\psi : \mathbb{F}_2 \to \mathbb{Z}[i]$ is given by $\psi(0) = 0$ and $\psi(1) = 1$, and $\Phi(u) = 1 + i$.

In the rest of this section, we use Construction A' to obtain complex BW lattices from a suitable linear code $\mathcal{C}_{2^m}$ over the quotient ring $\mathcal{U}_m$.

A. Linear codes for Construction A':

In order to obtain BW_{2^m} through Construction A', we first need to find a suitable linear code $\mathcal{C}_{2^m}$ over the ring $\mathcal{U}_m$. We propose such a linear code which can be obtained by the generator matrix

$$\mathbf{G}_{2^m} = \begin{bmatrix} 1 & 1 \\ 0 & u \end{bmatrix}^{\otimes m},$$

where the tensor operation is over the ring $\mathcal{U}_m$.

Example 1: To obtain BW_4, the linear code $\mathcal{C}_4$ can be generated using the generator matrix

$$\mathbf{G}_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & u & 0 & u \\ 0 & 0 & u & u \\ 0 & 0 & 0 & 0 \end{bmatrix} \in \mathcal{U}_2^{4 \times 4}.$$

Encoding of linear code $\mathcal{C}_{2^m}$

By using $\mathbf{G}_{2^m}$ as a matrix over $\mathcal{U}_m$, the code $\mathcal{C}_{2^m}$ is obtained as follows: Let $\mathbf{z} \in \mathcal{U}_m^{2^m}$ i.e., the j-th component of $\mathbf{z}$ is given by

$$z_j = \sum_{k=0}^{m-1} b_{k,j} u^k, \quad (5)$$

where $b_{k,j} \in \mathbb{F}_2$ for all k, j. Using $\mathbf{z}$ and $\mathbf{G}_{2^m}$, the code $\mathcal{C}_{2^m} \subseteq \mathcal{U}_m^{2^m}$ can be obtained as

$$\mathcal{C}_{2^m} = \left\{ \mathbf{x} = \mathbf{zG}_{2^m} \mid \forall \mathbf{z} \in \mathcal{U}_m^{2^m} \right\}, \quad (6)$$

where the matrix multiplication is over $\mathcal{U}_m$.

We now provide an example for the above encoding technique showing the positions of the information bits that get encoded to the codewords of $\mathcal{C}_{2^m}$.

Example 2: For $m = 2$, the input vector $\mathbf{z}$ and the generator matrix $\mathbf{G}_4$ are of the form

$$\mathbf{z}^T = \begin{bmatrix} b_{0,1} + b_{1,1} u \\ b_{0,2} \\ b_{0,3} \\ 0 \end{bmatrix} \text{ and } \mathbf{G}_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & u & 0 & u \\ 0 & 0 & u & u \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

We define the rate of the linear code $\mathcal{C}_{2^m}$ as the ratio of the number of information bits per codeword and the length of the code.

Proposition 1: The rate of the code $\mathcal{C}_{2^m}$ is $\frac{m}{2}$.

Proof: Each component of $\mathbf{z}$ carries m information bits in the variables $\{b_{k,j}\}_{k=0}^{m-1}$ as shown in (5). This amounts to a total of $m2^m$ bits carried by $\mathbf{z}$. However, since the matrix multiplication is over $\mathcal{U}_m$, not all the information bits of $\{b_{k,j}\}_{k=0}^{m-1}$ are encoded onto the codewords of $\mathcal{C}_{2^m}$ (since $u^k = 0$ for $k \geq m$). Using the structure of $\mathbf{G}_{2^m}$ it is possible to identify the indices (k, j) whose information bits are encoded. Let $\mathcal{I}_q$ for $0 \leq q \leq m-1$ denote the set of row indices of $\mathbf{G}_{2^m}$ that contains either 0 or u^q. Due to the quotient operation $u^m = 0$, the components of $\mathbf{z}$ having indices in $\mathcal{I}_q$ are essentially of the form,

$$z_j = \sum_{k=0}^{m-1-q} b_{k,j} u^k \ \forall j \in \mathcal{I}_q.$$

For instance, $z_1 = \sum_{k=0}^{m-1} b_{k,1} u^k$ and $z_{2^m} = 0$. Using the structure of $\mathbf{G}_{2^m}$ we observe that $|\mathcal{I}_q|$ is C_q^m, and hence, the total number of information bits per codeword of $\mathcal{C}_{2^m}$ is $\sum_{k=0}^{m-1}(m-k)C_k^m = \frac{m}{2}2^m$. ∎

We now show the equivalence between our encoding technique and the complex code formula [16]. In other words, the following theorem shows that the codewords generated in (6) can be uniquely represented as vectors of a multi-level code of nested RM codes.

Theorem 1: The codewords generated in (6) can be uniquely represented as codewords obtained through the complex code formula in (1).

Proof: The entries of $\mathbf{G}_{2^m}$ take values from the set $\{0, 1, u, u^2, \ldots, u^{m-1}\}$. After suitable row permutations, $\mathbf{G}_{2^m}$ can be written as

$$\mathbf{G}_{2^m} = \begin{bmatrix} \mathbf{R}_0 \\ u\mathbf{R}_1 \\ \vdots \\ u^{m-1}\mathbf{R}_{m-1} \\ u^m\mathbf{R}_m \end{bmatrix}, \tag{7}$$

where $\mathbf{R}_k \in \mathbb{F}_2^{C_k^m \times 2^m}$. Note that $[\mathbf{R}_0^T \ \mathbf{R}_1^T \ \cdots \ \mathbf{R}_r^T]^T$ is a generator matrix of the r-th order RM code for $r \leq m$. Recalling the encoding technique, the code $\mathcal{C}_{2^m}$ is obtained as

$$\mathcal{C}_{2^m} = \left\{ \mathbf{x} = \mathbf{z}\mathbf{G}_{2^m} \mid \forall\ \mathbf{z} \in \mathcal{U}_m^{2^m} \right\},$$

where the matrix multiplication is over $\mathcal{U}_m$. Further, the vector $\mathbf{z}$ can be written as $\mathbf{z} = \mathbf{u}\mathbf{B}$, where

$$\mathbf{u} = \begin{bmatrix} 1 & u & u^2 & \cdots & u^{m-2} & u^{m-1} \end{bmatrix} \in \mathcal{U}_m^{1\times m}$$

and

$$\mathbf{B} = \begin{bmatrix} b_{0,1} & b_{0,2} & \cdots & b_{0,2^m-1} & b_{0,2^m} \\ b_{1,1} & b_{1,2} & \cdots & b_{1,2^m-1} & b_{1,2^m} \\ b_{2,1} & b_{2,2} & \cdots & b_{2,2^m-1} & b_{2,2^m} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ b_{m-2,1} & b_{m-2,2} & \cdots & b_{m-2,2^m-1} & b_{m-2,2^m} \\ b_{m-1,1} & b_{m-1,2} & \cdots & b_{m-1,2^m-1} & b_{m-1,2^m} \end{bmatrix}.$$

Note that $b_{k,j}$ are the information bits to be encoded onto the codewords of $\mathcal{C}_{2^m}$. We partition the information matrix $\mathbf{B}$ as $[\mathbf{B}_0 \ \mathbf{B}_1 \ \cdots \ \mathbf{B}_m]$ where $\mathbf{B}_k \in \mathbb{F}_2^{m \times C_k^m}$ for $k = 1, 2, \ldots, m$. Incorporating the above partition, the BW lattice vector $\mathbf{x}$ can be written as

$$\mathbf{x} = \mathbf{u}[\mathbf{B}_0 \ \mathbf{B}_1 \ \cdots \ \mathbf{B}_m] \begin{bmatrix} \mathbf{R}_0 \\ u\mathbf{R}_1 \\ \vdots \\ u^{m-1}\mathbf{R}_{m-1} \\ u^m\mathbf{R}_m \end{bmatrix}.$$

The R.H.S of the above equation can be alternately written as

$$\mathbf{x} = \mathbf{u}[\bar{\mathbf{B}}_0 \ \bar{\mathbf{B}}_1 \ \cdots \ \bar{\mathbf{B}}_m] \underbrace{\begin{bmatrix} \mathbf{R}_0 \\ \mathbf{R}_1 \\ \vdots \\ \mathbf{R}_{m-1} \\ \mathbf{R}_m \end{bmatrix}}_{\mathbf{G}_{RM}},$$

where $\bar{\mathbf{B}}_k = \begin{bmatrix} \mathbf{0}_{k\times C_k^m} \\ \mathbf{B}_k([1:m-k],:) \end{bmatrix}$ and $\mathbf{B}_k([1:m-k],:)$ denotes the first $m-k$ rows of $\mathbf{B}_k$. Note that $\mathbf{G}_{RM}$ is the matrix containing the generator matrices of nested RM codes. We use the notation $\bar{\mathbf{B}} = [\bar{\mathbf{B}}_0 \ \bar{\mathbf{B}}_1 \ \cdots \ \bar{\mathbf{B}}_m]$, and point out that the informations bits in each row of $\bar{\mathbf{B}}$ are encoded onto the codewords of an appropriate RM code by the matrix multiplication $\bar{\mathbf{B}}\mathbf{G}_{RM}$. Due to the presence of zeros in $\bar{\mathbf{B}}$, the matrix $\bar{\mathbf{B}}$ has only $\sum_{n=0}^{k-1} C_n^m$ information bits in the k-th row of $\bar{\mathbf{B}}$ for $k = 1, 2, \ldots, m$. Since these $\sum_{n=0}^{k-1} C_n^m$ bits are placed in the first as many columns of $\bar{\mathbf{B}}$, the information bits in the k-th row of $\bar{\mathbf{B}}$ are encoded onto a codeword of $\mathcal{RM}(k-1, m)$. Finally, by the multiplication of $\mathbf{u}$ from left, the generated RM codewords are appropriately weighed by different powers of u and then added. This proves the equivalence of our construction to multilevel construction from RM codes. ∎

Remark 1: The equivalence shown in Theorem 1 implies that Construction A' provides the same bit labelling properties as that of the multilevel construction.

B. Embedding the linear code into the Euclidean space

We now discuss the embedding operation of $\mathcal{C}_{2^m}$ into the Euclidean space. By using the map $\Phi(\cdot)$ on the components of $\mathcal{C}_{2^m}$, we get the lattice code $\mathcal{EC}_{2^m}$. It can be verified that $\mathcal{EC}_{2^m}$ is an arbitrary subset of BW_{2^m} and does not have hypercube shaping. To fix this problem, we propose a one-to-one mapping ϕ on $\mathcal{EC}_{2^m}$ to obtain a new lattice code (denoted by $\mathcal{L}_{2^m}$) with the hypercube shaping property.

C. BW lattice codes with hypercube shaping property

We propose a one-to-one mapping ϕ on $\mathcal{EC}_{2^m}$ to obtain a new lattice code $\mathcal{L}_{2^m}$. Under such a mapping, we get hypercube shaping property when m is even and the rectangular shaping property when m is odd. For $\mathbf{x} =$

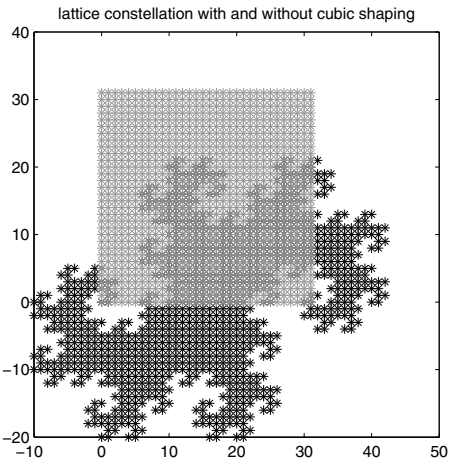

Fig. 1. Complex points generated by $\sum_{r=0}^{m-1}(1+i)^r b_r$ and $\phi\left(\sum_{r=0}^{m-1}(1+i)^r b_r\right)$ for $m=10$.

$[x_1, x_2, x_3, \ldots, x_{2^m}] \in \mathcal{EC}_{2^m}$, the mapping ϕ operates on each component of $\mathbf{x}$ as

$$\phi(x_j) = \begin{cases} x_j \bmod 2^{\frac{m}{2}}, & \text{when } m \text{ is even} \\ \varphi\left(x_j \bmod 2^{\frac{m+1}{2}}\right), & \text{when } m \text{ is odd,} \end{cases} \tag{8}$$

where $\varphi(\cdot)$ is defined on $\mathbb{Z}_{2^{\frac{m+1}{2}}}[i]$ as

$$\varphi(z) = \begin{cases} z, \text{ when } \Im(z) < 2^{\frac{m-1}{2}} \\ z + \left(2^{\frac{m-1}{2}} - i2^{\frac{m-1}{2}}\right), \text{ when } \Re(z) < 2^{\frac{m-1}{2}} \\ \qquad \text{and } \Im(z) \geq 2^{\frac{m-1}{2}} \\ z - \left(2^{\frac{m-1}{2}} + i2^{\frac{m-1}{2}}\right), \text{ when } \Re(z) \geq 2^{\frac{m-1}{2}} \\ \qquad \text{and } \Im(z) \geq 2^{\frac{m-1}{2}}. \end{cases} \tag{9}$$

The mapping ϕ guarantees the following property on $\mathcal{L}_{2^m}$:

$$\mathcal{L}_{2^m} \subseteq \begin{cases} \left\{\mathbb{Z}_{2^{\frac{m}{2}}}[i]\right\}^{2^m}, & \text{if } m \text{ is even} \\ \left\{\mathbb{Z}_{2^{\frac{m+1}{2}}}\right\}^{2^m} + i\left\{\mathbb{Z}_{2^{\frac{m-1}{2}}}\right\}^{2^m}, & \text{if } m \text{ is odd.} \end{cases} \tag{10}$$

From (10), each component of a vector in $\mathcal{L}_{2^m}$ is in a cubic box and a rectangular box, when m is even and odd, respectively. In Fig. 1, we present the complex points $\sum_{r=0}^{m-1}(1+i)^r b_r$ with and without the mapping ϕ for $m=10$.

Proposition 2: The mapping ϕ given in (8) is one-to-one.

Proof: Here we only provide the proof when m is even. For any $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{EC}_{2^m}$ we prove that $\phi(\mathbf{x}_1) \neq \phi(\mathbf{x}_2)$ if and only if $\mathbf{x}_1 \neq \mathbf{x}_2$. Applying the modulo operation in (8), $\mathbf{x}_j$ satisfies $\mathbf{x}_j = 2^{\frac{m}{2}}\mathbf{r}_j + \phi(\mathbf{x}_j)$ for each $j = 1, 2$, where $\phi(\mathbf{x}_j) \in \mathcal{L}_{2^m}$ and $\mathbf{r}_j \in \mathbb{Z}[i]^{2^m}$. This implies

$$\phi(\mathbf{x}_j) = \mathbf{x}_j - 2^{\frac{m}{2}}\mathbf{r}_j = \mathbf{x}_j + (1+i)^m \mathbf{r}'_j, \tag{11}$$

for some $\mathbf{r}'_j \in \mathbb{Z}[i]^{2^m}$. The second equality in (11) follows as $(1+i)^m = a2^{\frac{m}{2}}$, where $a \in \{1, -1, i, -i\}$. Further, since $\mathbf{x}_j$ is of the form $\sum_{r=0}^{m-1}(1+i)^r \psi(\mathbf{b}_r)$ for $\mathbf{b}_r \in \mathcal{RM}(r, m)$, the RHS of (11) is nothing but the multilevel representation of BW lattice from RM codes [16]. Since such a representation is unique, we have $\phi(\mathbf{x}_1) \neq \phi(\mathbf{x}_2)$ if and only if $\mathbf{x}_1 \neq \mathbf{x}_2$. This completes the proof when m is even. The one-to-one nature of ϕ can be proved on the similar lines when m is odd. ■

```
procedure SEQBW(r, y)
    if y ∈ C^N and N ≤ 2^r
        return ⌈y⌋
    else
        b = ⌈ℜ(y)⌋ + ⌈ℑ(y)⌋ mod 2
        ρ = 1−2(max(|⌈ℜ(y)⌋ − ℜ(y)|, |⌈ℑ(y)⌋| − ℑ(y)))
        ĉ = RMDEC(r, b, ρ)
        v = SEQBW(r + 1, (y − ĉ)/(1 + i))
        return ĉ + (1 + i)v
    end if
end procedure
```

Fig. 2. Algorithm for the SBWD of [9].

The above proposition implies that mapping ϕ provides a new lattice code with better shaping property. The following theorem shows that $\mathcal{L}_{2^m}$ can be used as a tile to obtain the BW lattice BW_{2^m}.

Theorem 2: The lattice code $\mathcal{L}_{2^m}$ and the lattice BW_{2^m} are related as $BW_{2^m} = (1+i)^m\mathbb{Z}[i]^{2^m} + \mathcal{L}_{2^m}$.

Proof: See the proof of Theorem 2 in [11]. ■

IV. On the error performance of the SBWD

We study the error performance of the SBWD for decoding the infinite BW lattice. In [9] it has been shown that for $\mathbf{x} \in BW_{2^m}$, if there exists $\mathbf{y} \in \mathbb{C}^{2^m}$ such that $d^2(\mathbf{x}, \mathbf{y}) \leq \frac{N}{4}$, where $N = 2^m$, then the SBWD correctly finds (or decodes) the lattice point $\hat{\mathbf{x}} = \mathbf{x}$. In the context of using SBWD in AWGN channels, $\mathbf{y}$ corresponds to $\mathbf{y} = \mathbf{x} + \mathbf{n}$, where $\mathbf{x} \in BW_{2^m}$ and $n_j \sim \mathcal{CN}(0, \sigma^2)\ \forall j$. This implies that the codeword error rate (CER) of the SBWD given by $\Pr(\hat{\mathbf{x}} \neq \mathbf{x})$ is upper bounded as

$$\Pr(\hat{\mathbf{x}} \neq \mathbf{x}) \leq \Pr\left(|\mathbf{n}|^2 > \frac{N}{4}\right).$$

Note that $\sqrt{\frac{N}{4}}$ is the packing radius of BW_{2^m}, and hence, the above bound is the well known *sphere upper bound* (SUB) [26]. In [9] the focus was only on the complexity of the decoder but not on the analysis of the tightness of the SUB. In other words, the possibility of correct decision is not known when $|\mathbf{n}|^2 > \frac{N}{4}$. We study the error performance and show that the decoder is powerful in making correct decisions well beyond the packing radius.

We first recall the SBWD algorithm of [9] in Fig. 2. This decoder is a successive interference cancellation (SIC) type decoder which exploits the BW lattice structure as multi-level construction of nested RM codes. At each level, the algorithm uses a variant of the soft-input RM decoder [25] (denoted by the function RMDEC which is given as Algorithm 3 in [9]) to decode the RM code in a modulo-$(1+i)$ channel. Thus, the error performance of the SBWD is determined by the error performance of the underlying soft-input RM decoders in the modulo-$(1+i)$ channels. In particular, we have

$$\Pr(\hat{\mathbf{x}} \neq \mathbf{x}) \quad = \quad \Pr\left(\bigcup_r \mathcal{E}(\hat{\mathbf{c}}_r \neq \mathbf{c}_r)\right), \tag{12}$$

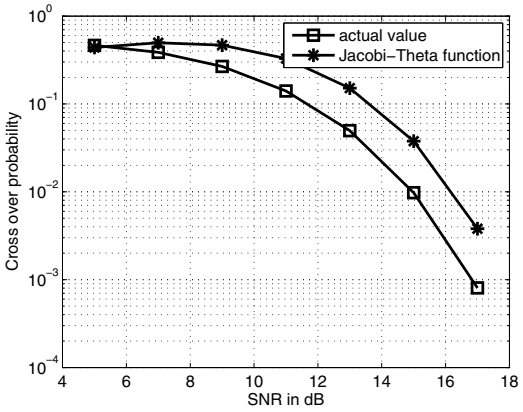

Fig. 3. Comparison of the cross-over probability with the upper bound using the Jacobi-Theta function.

where $\mathcal{E}(\hat{\mathbf{c}}_r \neq \mathbf{c}_r)$ denotes an error event while decoding $\mathcal{RM}(r,m)$. Hence, it is important to compute $\Pr(\hat{\mathbf{c}}_r \neq \mathbf{c}_r)$ for each $\mathcal{RM}(r,m)$. Along that direction, it is necessary to model the effective modulo-$(1+i)$ channel induced for each RM code $\mathcal{RM}(r,m)$.

A. Modulo-$(1+i)$ channel for $\mathcal{RM}(0,m)$

We propose a model for the modulo-$(1+i)$ channel which is accurate for $\mathcal{RM}(0,m)$. For $r \neq 0$, we cannot accurately model the channel due to the error propagation in the SIC decoder. Without loss of generality, we study the error performance when the zero lattice point is transmitted. To decode $\mathcal{RM}(0,m)$, a hard-decision binary value b_j is obtained from y_j as

$$b_j = \lceil \Re(y_j) \rfloor + \lceil \Im(y_j) \rfloor \bmod 2. \tag{13}$$

Note that the above expression is a realization of the modulo $(1+i)$ operation. Due to the combination of the round and the modulo operation (henceforth referred to as the round-modulo operation) in (13), the codewords of $\mathcal{RM}(0,m)$ are passed through a virtual binary channel with the cross-over probability given by

$$P_c = \Pr(b_j = 1 \mid c_j = 0).$$

Since the zero lattice point is transmitted, $\mathbf{c}$ is the all zero codeword for each $\mathcal{RM}(0,m)$, and hence, the relevant cross-over probability is $\Pr(b_j = 1 \mid c_j = 0)$. The following theorem shows that P_c can be upper bounded by a Jacobi-Theta function [13].

Theorem 3: The cross-over probability P_c induced by the round-modulo operation in (13) is upper bounded as

$$P_c \leq \left(e^{-\frac{1}{4\sigma^2}}\right) \vartheta\left(\frac{i4}{\pi\sigma^2}, \frac{i}{\pi\sigma^2}\right), \tag{14}$$

where $\vartheta(z,\tau)$ is the Jacobi-Theta function given by

$$\vartheta(z,\tau) = \sum_{a=-\infty}^{\infty} e^{\pi i a^2 \tau + 2\pi i a z}.$$

Proof: We first compute P_c and then propose an upper bound. To assist the computation of P_c, we compute the probability that $\Re(y_j)$ (or $\Im(y_j)$) falls within an interval $(z-0.5, z+0.5]$ centred around an integer z, when $c_j = 0$. Since the additive noise is circularly symmetric, it is sufficient to calculate the above probability for either $\Re(y_j)$ or $\Im(y_j)$. We use y to denote either $\Re(y_j)$ or $\Im(y_j)$. For the odd integer case, we have

$$\begin{aligned} P_{\mathrm{o}} &\triangleq \sum_{a=-\infty}^{\infty} \Pr\left(2a+0.5 < y \leq 2a+1.5\right) \\ &= \sum_{a=-\infty}^{\infty} \left[\int_{2a+0.5}^{2a+1.5} \mathrm{P}_{\mathbf{y}}(y)dy\right] \\ &= \sum_{a=-\infty}^{\infty} \left[Q\left(\frac{2a+0.5}{\sigma/\sqrt{2}}\right) - Q\left(\frac{2a+1.5}{\sigma/\sqrt{2}}\right)\right], \end{aligned} \tag{15}$$

where $\mathrm{P}_{\mathbf{y}}(y)$ is the probability density function of y, $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} e^{-\frac{u^2}{2}} du$, and $\sigma^2/2$ is the variance of y. For the even integer case, we have

$$\begin{aligned} P_{\mathrm{e}} &\triangleq \sum_{a=-\infty}^{\infty} \Pr\left(2a-0.5 < y \leq 2a+0.5\right) \\ &= \sum_{a=-\infty}^{\infty} \left[\int_{2a-0.5}^{2a+0.5} \mathrm{P}_{\mathbf{y}}(y)dy\right] \\ &= \sum_{a=-\infty}^{\infty} \left[Q\left(\frac{2a-0.5}{\sigma/\sqrt{2}}\right) - Q\left(\frac{2a+0.5}{\sigma/\sqrt{2}}\right)\right]. \end{aligned} \tag{16}$$

Note that b_j is 1 whenever $\lceil \Re(y_j) \rfloor + \lceil \Im(y_j) \rfloor$ is an odd number. This can happen when (*i*) $\lceil \Re(y_j) \rfloor$ is odd and $\lceil \Im(y_j) \rfloor$ is even, or (*ii*) $\lceil \Re(y_j) \rfloor$ is even and $\lceil \Im(y_j) \rfloor$ is odd. From (15) and (16), we can write

$$P_c = P_{\mathrm{o}}(1-P_{\mathrm{o}}) + (1-P_{\mathrm{o}})P_{\mathrm{o}} \tag{17}$$

$$= 2P_{\mathrm{o}} - 2(P_{\mathrm{o}})^2. \tag{18}$$

By dropping the term $2(P_{\mathrm{o}})^2$, we upper bound P_c as

$$P_c \leq 2P_{\mathrm{o}}$$

$$\leq 2\sum_{a=-\infty}^{\infty} \left[Q\left(\frac{2a+0.5}{\sigma/\sqrt{2}}\right)\right] \tag{19}$$

$$\leq \sum_{a=-\infty}^{\infty} e^{-\frac{(2a+0.5)^2}{\sigma^2}} \tag{20}$$

$$= e^{-\frac{(0.5)^2}{\sigma^2}} \sum_{a=-\infty}^{\infty} e^{\frac{-4a^2-2a}{\sigma^2}}$$

$$= \left(e^{-\frac{1}{4\sigma^2}}\right) \vartheta\left(\frac{i4}{\pi\sigma^2}, \frac{i}{\pi\sigma^2}\right),$$

where the bound in (19) comes from dropping the terms of the form $Q\left(\frac{2a+1.5}{\sigma/\sqrt{2}}\right)$ in (15), and the bound in (20) is due to the Chernoff bound $Q(x) \leq \frac{1}{2}e^{\frac{-x^2}{2}}$. ■

Note that the Jacobi-Theta function can be evaluated at any pair (τ, z). In Fig. 3, the empirical values of P_c are presented along with the bound in (14) for various values of SNR $= \frac{1}{\sigma^2}$. We point out that the bound is not tight due to the Chernoff-bound on each $Q(\cdot)$ function.

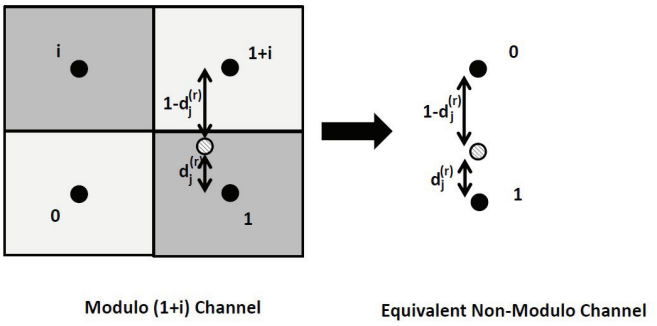

Fig. 4. Equivalence between the modulo-$(1+i)$ channel and the non-modulo channel when $n_{e,j}^{(r)} = 1 - d_j^{(r)}$.

B. Upper bound on the error performance for decoding $\mathcal{RM}(r,m)$

We provide an upper bound on $\Pr(\hat{\mathbf{c}}_r \neq \mathbf{c}_r)$. To distinguish the decoding operation for each $\mathcal{RM}(r,m)$, we denote the hard-decision vector $\mathbf{b}$ and the soft-input vector $\mathbf{d}$ by $\mathbf{b}^{(r)}$ and $\mathbf{d}^{(r)}$, respectively, where

$$\mathbf{b}^{(r)} = \lceil \Re(\mathbf{y}^{(r)}) \rfloor + \lceil \Im(\mathbf{y}^{(r)}) \rfloor \bmod 2, \text{ and}$$

$$\mathbf{d}^{(r)} = \max\left(|\lceil \Re(\mathbf{y}^{(r)}) \rfloor - \Re(\mathbf{y}^{(r)})|, |\lceil \Im(\mathbf{y}^{(r)}) \rfloor| - \Im(\mathbf{y}^{(r)}) \right).$$

Note that $d_j^{(r)}$ is the effective distance between the received complex number $y_j^{(r)}$ and the nearest coset representative of $b_j^{(r)}$ in the modulo-$(1+i)$ channel. As depicted in Fig. 4, $d_j^{(r)}$ is the effective distance between $y_j^{(r)}$ (as shown in white circle with diagonal stripes) and the nearest coset representative of $b_j^{(r)} = 1$ (the complex number 1). We now propose an equivalent non-modulo channel for the modulo-$(1+i)$ channel at each level. If $\mathbf{c}^{(r)}$ denotes a codeword of $\mathcal{RM}(r,m)$, one can imagine $\mathbf{b}^{(r)}$ and $\mathbf{d}^{(r)}$ to be obtained from an equivalent non-modulo channel given by

$$\mathbf{y}_e^{(r)} = \mathbf{c}^{(r)} + \mathbf{n}_e^{(r)},$$

where

$$n_{e,j}^{(r)} = \begin{cases} d_j^{(r)}, & \text{when } b_j^{(r)} = c_j^{(r)} \\ 1 - d_j^{(r)}, & \text{when } b_j^{(r)} \neq c_j^{(r)}, \end{cases} \tag{21}$$

for $1 \leq j \leq N$, where $n_{e,j}^{(r)}$ denotes the j-th component of $\mathbf{n}_e^{(r)}$. This equivalence between the modulo-$(1+i)$ channel and the non-modulo channel is shown in Fig. 4. Note that $n_{e,j}^{(r)}$ has bounded support in the interval $[0,1]$. For an analogy with respect to the model in [25], the code alphabet of $\mathcal{RM}(r,m)$ here corresponds to the code alphabet $\{-1,1\}$ in [25], and the effective noise $\mathbf{n}_e^r$ here corresponds to the AWGN in [25]. At the r-th level of the BW lattice, the code $\mathcal{RM}(r,m)$ has the minimum squared distance of 2^{m-r}. Since the soft-input RM decoder is a bounded distance decoder with radius equal to the minimum squared distance, the probability of incorrect decision of the soft-input RM decoder is upper bounded (using the proposition in Section IV.A of [25]) as

$$\Pr(\hat{\mathbf{c}}_r \neq \mathbf{c}_r) \leq \Pr\left(|\mathbf{n}_e^{(r)}|^2 > \frac{2^{m-r}}{4} \right). \tag{22}$$

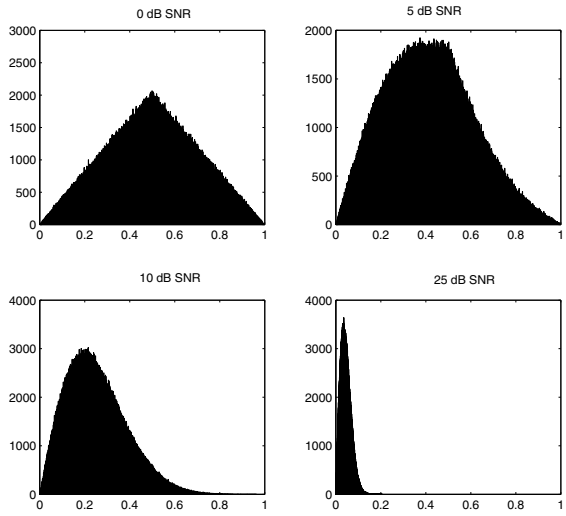

Fig. 5. Histogram of $n_e^{(0)}$ when SNR = $\frac{1}{\sigma^2}$ takes the values 0 dB, 5 dB, 10 dB, and 25 dB.

For $r = 0$, the bound becomes

$$\Pr(\hat{\mathbf{c}}_0 \neq \mathbf{c}_0) \leq \Pr\left(|\mathbf{n}_e^{(0)}|^2 > \frac{N}{4} \right).$$

It is important to note that the above bound is different from $\Pr(|\mathbf{n}|^2 > \frac{N}{4})$ since $\mathbf{n}$ is Gaussian distributed while $\mathbf{n}_e^{(0)}$ is not. We do not have a closed form expression on the distribution of $|\mathbf{n}_e^{(r)}|^2$. In Fig. 5, we display the histogram of the realizations of $n_e^{(0)}$ for various values of σ^2, when the zero RM codeword is transmitted. Note that for σ^2 = 0 dB, the histogram of $n_e^{(0)}$ has a triangular shape centred around 0.5, which implies a very high (close to 0.5) cross-over probability when obtaining the hard decision vector $\mathbf{b}$. On the other hand, at lower values of σ^2, the distribution is skewed towards zero indicating smaller cross-over probability.

V. SBWD TO DECODE BW LATTICE CODE $\mathcal{L}_{2^m}$ FOR AWGN CHANNEL

We discuss the use of SBWD to decode the lattice code $\mathcal{L}_{2^m}$. First, we describe a method to transmit the codewords of $\mathcal{L}_{2^m}$. For any $\mathbf{x} \in \mathcal{L}_{2^m}$, the transmitted vector is of the form

$$\mathbf{x}_t = (2\mathbf{x} - c), \tag{23}$$

where

$$c = \begin{cases} \left(2^{\frac{m}{2}} - 1\right) + i\left(2^{\frac{m}{2}} - 1\right), & \text{when } m \text{ is even} \\ \left(2^{\frac{m+1}{2}} - 1\right) + i\left(2^{\frac{m-1}{2}} - 1\right), & \text{when } m \text{ is odd.} \end{cases} \tag{24}$$

The components of the transmitted vector are offset by a constant c towards the origin to reduce the average transmit energy. Using the scale and the shift operation in (23), each component of $\mathbf{x}_t$ takes a value from the regular 2^m-QAM constellation. In particular, the QAM constellation is square and non-square when m is even and odd, respectively. When $\mathbf{x}_t$ is transmitted, the received vector $\bar{\mathbf{y}}$ is given by

$$\bar{\mathbf{y}} = \mathbf{x}_t + \bar{\mathbf{n}}, \tag{25}$$

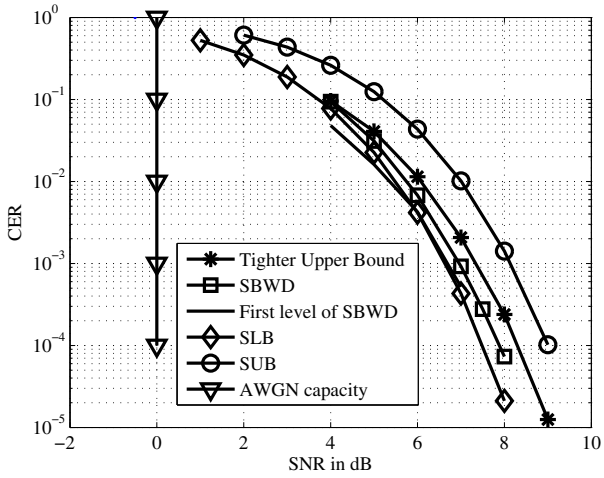

Fig. 6. CER of SBWD for decoding $\mathcal{L}_4$

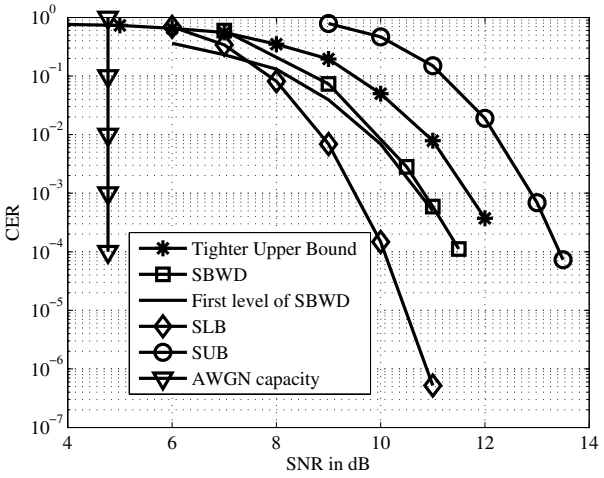

Fig. 7. CER of SBWD for decoding $\mathcal{L}_{16}$.

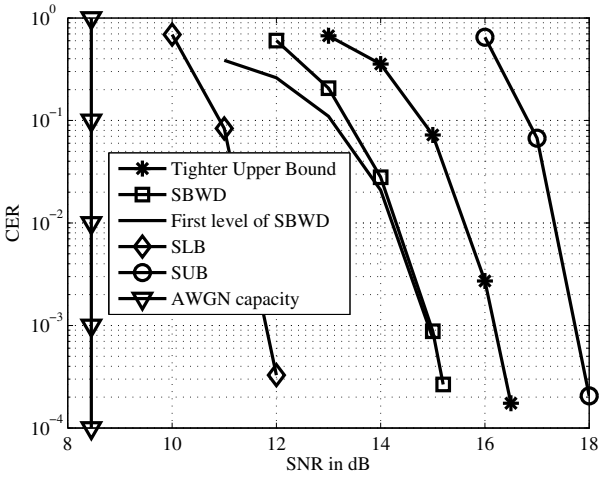

Fig. 8. CER of SBWD for decoding $\mathcal{L}_{64}$.

where $\bar{\mathbf{n}}$ is the AWGN with $\bar{n}_j \sim \mathcal{CN}(0, \sigma^2)\ \forall j$. In this section, SNR of the channel is defined as E_s/σ^2, where E_s denotes the average energy of 2^m-QAM constellation. With the inverse operation to (23) as $\mathbf{y} = \frac{1}{2}\bar{\mathbf{y}} + c$, the equivalent AWGN channel becomes

$$\mathbf{y} = \mathbf{x} + \mathbf{n}, \tag{26}$$

where $\mathbf{x} \in \mathcal{L}_{2^m}$ and $n_j \sim \mathcal{CN}(0, \frac{\sigma^2}{4})$. We use the SBWD [9] on (26) to recover the information. When a codeword of $\mathcal{L}_{2^m}$ is transmitted, the SBWD decodes to a lattice point in the infinite lattice BW_{2^m}. In such a decoding method, irrespective of whether the decoded lattice point falls in the code or not, the information bits can be recovered from the decoded RM codewords at every level of SBWD (as shown in the algorithm in Fig. 2).

A. Simulation results on the codeword error rate (CER) of SBWD

In this subsection, we present the CER of the SBWD along with some upper bounds and lower bounds. For the simulation results, we use SNR $= E_s/\sigma^2$, where E_s denotes the average energy of the regular 2^m-QAM constellation. In each of Fig. 6-10, we present (*i*) the CER of the SBWD, (*ii*) the SUB [26], (*iii*) the sphere lower bound (SLB) [26]), (*iv*) the CER in decoding $\mathcal{RM}(0, m)$ at the first level of the SBWD, and (*v*) the upper bound on the CER in decoding $\mathcal{RM}(0, m)$ given by $\Pr(|\mathbf{n}_e^{(0)}|^2 > \frac{N}{4})$ (obtained through simulation results by empirically generating $\mathbf{n}_e^{(0)}$).

From Fig. 6-10, we make the following observations: the SUB is not a tight upper bound on the CER of SBWD. Also, $\Pr(|\mathbf{n}_e^{(0)}|^2 > \frac{N}{4})$ is an upper bound on the CER of SBWD, and in particular, it is a tighter upper bound than the SUB. In general, an upper bound on $\Pr(\hat{\mathbf{c}}_0 \neq \mathbf{c}_0)$ need not be an upper bound on $\Pr(\hat{\mathbf{x}} \neq \mathbf{x})$. However, in this case, our observation that $\Pr(|\mathbf{n}_e^{(0)}|^2 > \frac{N}{4})$ is an upper bound on the CER of SBWD follows from Fig. 6-10. The CER of the soft-input RM decoder for $\mathcal{RM}(0, m)$ is a tight lower bound on the CER of the SBWD. This implies that if there is no error at the first level of the decoder, then with high probability, there will be no errors at subsequent levels of the soft-input RM decoder. We have also marked the SNR required by a capacity approaching scheme to achieve the spectral efficiency of $m/2$ bits per channel use (using the expression $C = \log_2(1 + \text{SNR})$). The plots show that the SBWD performs away from the channel capacity for large block lengths. In summary, the simulation results highlight that the SBWD is powerful in making correct decisions even beyond the packing radius, and the deviation from the SUB increases for larger dimensions. As a result SBWD can be employed to efficiently decode lattice codes of large block lengths with low-complexity.

B. Comparing the complexity of the SBWD with the list decoder in [10]

We compare the complexity of the SBWD with the BW list decoder [10]. For a fair comparison, we assume that the list decoder is implemented on a single processor. On a single processor, the complexity of the SBWD is $O(N\log^2(N))$, whereas the complexity of the list decoder is $O(N^2)(l(m, \eta))^2$, where $l(m, \eta)$ is the worst case list

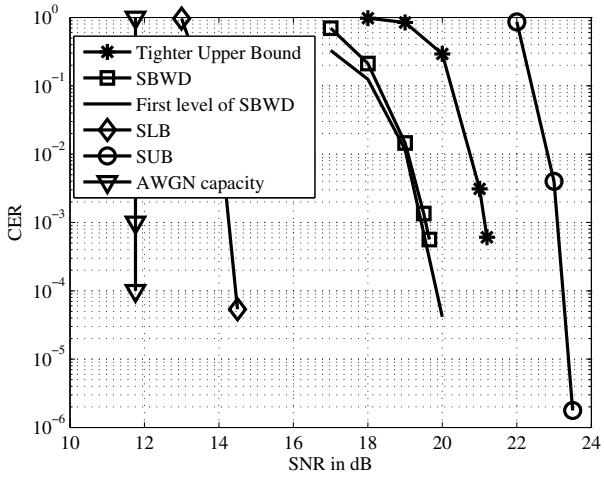

Fig. 9. CER of SBWD for decoding $\mathcal{L}_{256}$.

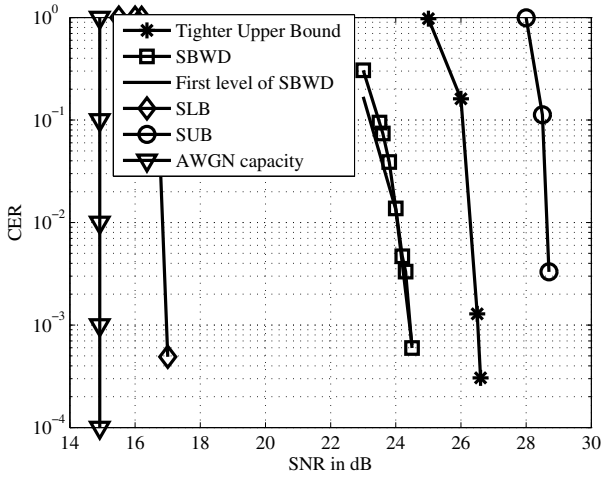

Fig. 10. CER of SBWD for decoding $\mathcal{L}_{1024}$.

size at a relative squared distance of η (the relative squared distance is the squared Euclidean distance normalized by the dimension of the lattice). We compare the complexity of the two decoders for a codeword error rate of 10^{-3}. In particular, we first approximate the error performance of the SBWD as a bounded distance decoder for some radius $\bar{\eta}$, and then compute the complexity of the list decoder with the corresponding value of $\bar{\eta}$. In Table I, we display the lower bound (as given in Theorem 1.3 in [10]) on the complexity of the list decoder to achieve the error performance of SBWD. The table shows that the list decoder has higher complexity than the SBWD to achieve the same performance. In summary, for single processor implementation, SBWD can be preferred to the list decoder to decode BW lattice codes of large block lengths. However, for codeword error rates lower than that of SBWD, the list decoder has to be used, preferably on parallel processors. Table I also shows the potential of SBWD to decode well beyond the relative squared distance of $\eta = 0.25$. For complex dimensions of 256 and 1024, the effective radius of SBWD is as high as $\frac{N}{2}$ and $\frac{2N}{3}$, respectively.

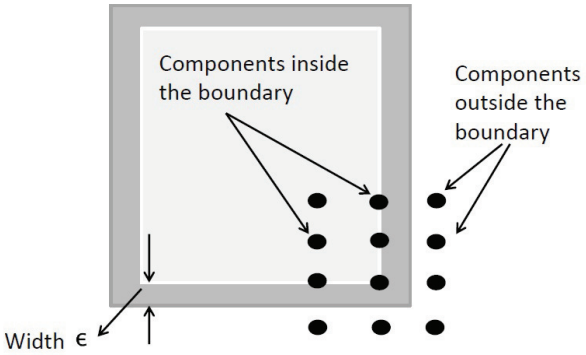

Fig. 11. Geometric explanation of the noise trimming technique for each component of the received vector.

procedure TRIM(y, ϵ) ▷ Input y is either $\Re(y_j)$ or $\Im(y_j)$
  $\Delta = (2^{\frac{m}{2}} - 1)/2$
  $r = y - \Delta$
  $t = \Delta + \epsilon$
  if $|r| > t$ ▷ Check out of boundary components
    $s = t/|r|$ ▷ Choose an appropriate scale value
    $b = sr$ ▷ Scale the received component
  else
    $b = r$ ▷ Do not scale
  end if
  return $b + \Delta$
end procedure

Fig. 12. Algorithm for the trimming technique when m is even.

VI. Noise trimming technique for the SBWD

When a codeword of $\mathcal{L}_{2^m}$ is transmitted, the SBWD decodes to a lattice point in the infinite lattice BW_{2^m}. To improve the error performance, we use a noise trimming technique that forces the SBWD to decode to a codeword of $\mathcal{L}_{2^m}$. We refer to such a decoder as the BW lattice code decoder (BWCD). From (10), each component of a codeword is within a rectangular box $\mathcal{B} \subseteq \mathbb{C}$. In particular, the box $\mathcal{B}$ shares its edges with that of $\mathbb{Z}_{2^{\frac{m}{2}}}[i]$ and $\mathbb{Z}_{2^{\frac{m+1}{2}}} + i\mathbb{Z}_{2^{\frac{m-1}{2}}}$ when m is even and odd, respectively. To decode to a codeword, we first *trim* the in-phase and quadrature components of the received vector to lie within a box $\mathcal{B}' \supseteq \mathcal{B}$ marginally larger than $\mathcal{B}$ by length ϵ on each dimension, and then feed the trimmed received vector to the SBWD. A geometric interpretation of the noise trimming technique is shown in Fig. 11. Note that the choice of ϵ is crucial to decode a codeword. In Fig. 12, we present the algorithm for the trimming method which works independently on the in-phase and quadrature component of the scalars in $\mathbf{y} = [y_1, y_2, \ldots, y_{2^m}]$ in (26) when m is even. Extension to the case when m is odd is straightforward.

Using the BWCD, we have obtained the BER results for dimensions 2^m when $m = 2, 4, 6, 8$, and 10, and have compared them with the BER of the SBWD (without noise trimming technique). The plots as shown in Fig. 13 indicate that the BWCD outperforms the SBWD by at most 0.5 dB for $m = 2$ and by 0.1 dB for $m = 10$. Note that the

TABLE I
COMPLEXITY OF THE LIST DECODER [10] TO ACHIEVE THE PERFORMANCE OF SBWD

Dimension N	$\bar{\eta}$	A lower bound on $N^2(l(m,\bar{\eta}))^2$	$N\log^2(N)$ (complexity of SBWD)
4	0.33	16	16
16	0.4	256	256
64	0.48	4096	2304
256	0.56	262144	16384
1024	0.67	1.07×10^9	102400

trimming technique is most effective on the codewords near the boundary of the code. For the simulation results, BER is obtained by averaging over all the codewords of $\mathcal{L}_{2^m}$ with the assumption that information bits are uniformly distributed. For large values of m, the size of the underlying QAM increases which in turn reduces the percentage of codeword components along the boundary. Hence, the advantage of the noise trimming technique diminishes for codes with larger QAM constellations. For the presented results, we have used $\epsilon = \frac{1}{2\sqrt{2}}$, which corresponds to the packing radius of $\sqrt{\frac{N}{4}}$. The above value of ϵ was optimized based on the simulation results by comparing the BER for various values of ϵ. Intuitively, trimming the received vector to fall within the packing radius of a lattice point in the boundary of the lattice code forces the SBWD to decode to a codeword instead of a lattice point outside the code. In general, the proposed noise trimming technique is applicable for any lattice code with hypercube shaping property.

Remark 2: The proposed noise trimming technique may remind the reader of minimum mean square error (MMSE) scaling proposed in [5]. However, the two techniques have the following significant differences: (*i*) MMSE scaling uses a constant scale factor applied on every component of the received vector irrespective of whether it falls outside the boundary of the lattice code or otherwise, whereas the noise trimming is applied with different scales to different components depending on the relative distance from the boundary of the lattice code. (*ii*) MMSE scaling can be applied to lattice codes with arbitrary shaping. However, the noise trimming technique is easily applicable for hypercube shaping. For arbitrary shaping, generalization of the noise trimming technique is too complex to implement.

VII. CONCLUSION AND DIRECTIONS FOR FUTURE WORK

We introduced a new method of encoding complex BW lattices using linear codes over polynomial rings and then have studied the performance of complex BW lattice codes for communication over AWGN channels. To encode the code, we use Construction A', and to decode the code we adapt the SBWD. We have studied the error performance of the SBWD, and have shown that the Jacobi-Theta functions can characterize the virtual binary channels that arise in the decoding process. We have also shown that the SBWD is powerful in making correct decisions beyond the packing radius. Subsequently, we have used the SBWD to decode the complex lattice code through the noise trimming technique. This work can be extended in one of the following ways: The SBWD proposed in [9] uses a soft-input, hard-output RM decoder at each level of BW lattice. It will be interesting to study the error performance of the lattice decoder with soft-input, soft-output iterative RM decoders. We have presented the error performance of the SBWD through simulation results, and hence, we now know the SBWD error performance with reference to the sphere lower bound and the sphere upper bound. A closed form expression on the error performance of the SBWD could be obtained for a better understanding of the decoder performance.

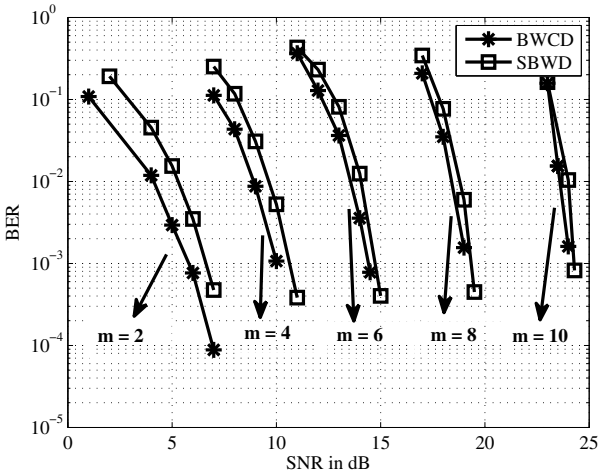

Fig. 13. BER comparison between BWCD and SBWD.

REFERENCES

[1] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. Wiley, 2006.
[2] N. Sommer, M. Feder, and O. Shalvi, "Low-density lattice codes," *IEEE Trans. Inf. Theory*, vol. 54, no. 4, pp. 1561-1585, Apr. 2008.
[3] M. Sadeghi, A. Banihashemi, and D. Panario, "Low-density parity-check lattices: construction and decoding analysis," *IEEE Trans. Inf. Theory*, vol. 52, no. 10, pp. 4481-4495, Oct. 2006.
[4] O. Shalvi, N. Sommer, and M. Feder, "Signal codes: convolutional lattice codes," *IEEE Trans. Inf. Theory*, vol. 57, no. 8, pp. 5203-5226, Aug. 2011.
[5] U. Erez and R. Zamir, "Achieving $\frac{1}{2}$log(1+SNR) on the AWGN channel with lattice encoding and decoding," *IEEE Trans. Inf. Theory*, vol. 50, no. 10, pp. 2293–2314, Oct. 2004.
[6] G. D. Forney and G. Ungerboeck, "Modulation and coding for linear Gaussian channels," *IEEE Trans. Inf. Theory*, vol. 44, no. 6, pp. 2384-2413, Oct. 1998.
[7] E. S. Barnes and G. E. Wall, "Some extreme forms defined in terms of Abelian groups," *J. Austral. Math. Soc.* 1, pp. 47–63, 1959.
[8] G. E. Wall, J. Pitman, and R. B. Potts, "Eric Stephen Barnes 1924-2000," *Historical Records of Australian Science*, vol. 15, no. 1, pp. 21–45, June 2004.
[9] D. Micciancio and A. Nicolosi, "Efficient bounded distance decoders for Barnes-Wall lattices," in *Proc. 2008 IEEE ISIT*.
[10] E. Grigorescu and C. Peikert, "List decoding Barnes-Wall lattices," in *Proc. 2012 IEEE Conference on Computational Complexity*, pp. 316–325. Also available at arXiv:1112.1994v2, Dec. 2011.
[11] J. Harshan, E. Viterbo, and J.-C. Belfiore, "Construction of Barnes-Wall lattices from linear codes over rings," in *Proc. 2012 IEEE ISIT*.

[12] J. Harshan, E. Viterbo, and J.-C. Belfiore, "Practical decoders for Barnes-Wall lattice constellations," in *2012 International Symposium on Mathematical Theory of Networks and Systems*.
[13] W. Reinhardt and P. Walker, *Theta Functions*. Cambridge University Press, 2010.
[14] E. S. Barnes and N. J. A Sloane, "New lattice packings of spheres," *Canadian J. Mathematics*, vol. 35, pp. 117–130, 1983.
[15] J. H. Conway and N. J. A Sloane, *Sphere Packings, Lattices and Groups*, 2nd ed. Springer-Verlag, 1993.
[16] G. D. Forney, "Coset codes—part II: binary lattices and related codes," *IEEE Trans. Inf. Theory*, vol. 34, no. 5, pp. 1152–1187, Sept. 1988.
[17] G. Nebe, E. M. Rains, and N. J. A. Sloane, "A simple construction of the Barnes-Wall lattices," in *Codes, Graphs, and Systems: A Celebration of the Life and Career of G. David Forney, Jr. on the Occasion of his Sixtieth Birthday*, 2002, pp. 333–342.
[18] A. H. Banihashemi and F. R. Kschischang, "Tanner graphs for group block codes and lattices: construction and complexity," *IEEE Trans. Inf. Theory* vol. 47, no. 2, pp. 822–834, 2001.
[19] A. J. Salomon and O. Amrani, "Augmented product codes and lattices: Reed-Muller codes and Barnes-Wall lattices," *IEEE Trans. Inf. Theory*, vol. 51, no. 11, pp. 3918–3930, Nov. 2005.
[20] R. E. Blahut, *Theory and Practice of Error Control Codes*. Addison-Wesley Publishing Company, Inc., 1983.
[21] G. D. Forney and A. Vardy, "Generalized minimum-distance decoding of Euclidean-space codes and lattices," *IEEE Trans. Inf. Theory*, vol. 42, no. 6, pp. 1992–2026, Nov. 1996.
[22] G. D. Forney "A bounded-distance decoding algorithm for the Leech lattice, with generalizations," *IEEE Trans. Inf. Theory*, vol. 35, no. 4, pp. 906–909, July 1989.
[23] M. Ran and J. Snyders, "Efficient decoding of the Gosset, Coxeter-Todd and the Barnes-Wall lattices," in *Proc. 1998 IEEE ISIT*.
[24] M. D. Yucel "New decoding strategy for the 32-dimensional Barnes-Wall lattice," *IEEE Electron. Lett.*, vol. 29, no. 13, pp. 1231–1232, June 1993.
[25] G. Schnabl and M. Bossert, "Soft-decision decoding of RM codes as generalized multiple concatenated codes," *IEEE Trans. Inf. Theory*, vol. 41, no. 1, pp. 304–308, Jan. 1995.
[26] E. Viterbo and E. Biglieri, "Computing the Voronoi cell of a lattice: the diamond-cutting algorithm," *IEEE Trans. Inf. Theory*, vol. 42, no. 1, pp. 161–171, Jan. 1996.
[27] W. Kositwattanarerk and F. Oggier, "On construction D and related constructions of lattices from linear codes," in *Proc. 2013 Int. Workshop on Coding and Cryptography*, pp. 428–437.

J. Harshan (M'12) was born in Bangalore, India. He received the B.E degree in Electronics and Communications from Visvesvaraya Technological University, India in 2004, and the Ph.D. degree from the Department of Electrical Communication Engineering, Indian Institute of Science, India in 2010. From September 2011, he is a post-doctoral researcher at the Department of Electrical and Computer Systems Engineering, Monash University, Melbourne. His research interests are in the broad areas of Signal design for wireless networks, Design, development, and deployment aspects of next-generation wireless communication systems, and Applications of information theory and coding theory to communications.

During 2004-2005 and 2010-2011, he was with Robert Bosch (India) Limited, Bangalore, India and Broadcom Communications Technologies, Bangalore, India, respectively. He has served in the TPC of the IEEE Australian Communication Theory Workshop 2013 held at Adelaide, Australia. He is the recipient of the Best Ph.D. Thesis award - The Seshagiri-Kaikini Medal for 2010-2011 from the council of Indian Institute of Science.

Emanuele Viterbo (M'95-SM'04-F'11) received the Laurea and Ph.D. degrees from the Politecnico di Torino, Torino, Italy, in 1989 and 1995, respectively, both in electrical engineering. From 1990 to 1992, he was with the European Patent Office, The Hague, The Netherlands, as a Patent Examiner working in the field of dynamic recording and error-control coding. Between 1995 and 1997, he held a postdoctoral position with the Dipartimento di Elettronica, Politecnico di Torino, Torino, Italy. During 19971998, he was a Post-Doctoral Research Fellow with the Information Sciences Research Center, AT&T Research, Florham Park, NJ. He became first an Assistant Professor (1998) and then an Associate Professor (2005) with the Dipartimento di Elettronica at Politecnico di Torino. In 2006, he became a Full Professor with DEIS, University of Calabria, Italy. Since 2010, he has been a Full Professor with the Department of Electrical and Computer Systems Engineering and Associate Dean for Research Training in the Faculty of Engineering, Monash University, Clayton, Australia. In 1993, he was a Visiting Researcher with the Communications Department, DLR, Oberpfaffenhofen, Germany. In 1994 and 1995, he was visiting the cole Nationale Superieure des Telcommunications, Paris, France. In 2003, he was a Visiting Researcher with the Math Department, EPFL, Lausanne, Switzerland. In 2004, he was a Visiting Researcher with the Telecommunications Department, UNICAMP, Campinas, Brazil. In 2005, 2006, and 2009 he was a Visiting Researcher with the ITR, UniSA, Adelaide, Australia. In 2007, he was a Visiting Fellow with the Nokia Research Center, Helsinki, Finland. His main research interests are in lattice codes for the Gaussian and fading channels, algebraic coding theory, algebraic space-time coding, digital terrestrial television broadcasting, and digital magnetic recording. Prof. Emanuele Viterbo is an ISI Highly Cited Researcher and Member of the Board of Governors of the IEEE Information Theory Society (2011–2013). He was an associate editor of the IEEE TRANSACTIONS ON INFORMATION THEORY and Guest Editor for the IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING Special Issue on Managing Complexity in Multiuser MIMO Systems. He was awarded a NATO Advanced Fellowship in 1997 from the Italian National Research Council.

Jean-Claude Belfiore received the Doctorat (PhD) from ENST in 1989. He was then enrolled at the Ecole Nationale Superieure des Telecommunications, ENST (Telecom ParisTech) where he is full Professor in the Communications and Electronics department. Prof. Belfiore has made pioneering contributions on modulation and coding for wireless systems by using tools of number theory. He is now working on wireless network coding, and coding for physical security and for interference channels. He is (co-)author of more than 200 papers and has served as advisor for more than 30 Ph.D. students. Prof. Belfiore is the recipient of the 2007 Blondel Medal and was appointed Associate Editor for Coding Theory of the IEEE TRANSACTIONS ON INFORMATION THEORY (Aug. 2011).