Non-Linear Precoding with Periodically Flipped Constellation for MIMO Downlink Channels

Laurent Mazet, Sheng Yang, Lina Mroueh and Marc de Courville
Motorola-Labs Paris – Parc Les Algorithmes, Saint Aubin – 91193 Gif sur Yvette Cedex – France
mazet@softndesign.org, sheng.yang@supelec.fr, lina.mroueh@enst.fr,
marc.de.courville@motorola.com

Abstract

In this paper, we propose a new constellation scheme for non-linear precoding in MIMO downlink channels. Instead of using a periodically replicated constellation, the proposed periodically flipped constellation (PFC) successively mirrors the existing constellation to form an infinite constellation. It is shown that the mirroring operation increases the effective minimum distance of the original constellation and provides a superior error resistance over the conventional constellation scheme for low order modulations, and low error rate scenarios. By allowing for a contraction on the PFC, further performance improvement due to a better tradeoff between the average transmit power and minimum distance is achieved. Then, practical implementation of the PFC specified sphere encoder is proposed. With a Schnorr-Euchner based algorithm, the sphere encoder with PFC does not suffer from any penalty in terms of complexity. Finally, we apply the PFC in an OFDMA system and evaluate the performance with numerical simulations.

I. Introduction

The next generation cellular system (such as IEEE 802.16m [1], LTE advanced [2], etc) features Multiple-Input Multiple-Output (MIMO) transmission (see [3], [4]) and multi-user communications.

In the uplink channel, as known as the Multiple Access Channel (MAC), of such systems, multiple mobile terminals transmit simultaneously to the base station. The latter treats the received signal in such a way that messages from different mobile terminals are distinguishable. The capacity region of a multiple access[5] has been known decades ago. A capacity achieving scheme is the Successive Interference Cancellation (SIC). This scheme has been well studied and extends naturally in the MIMO case.

Unlike the uplink channel, little is known for the downlink channel, as known as the Broadcast Channel (BC), until recent years. Solid progress on the capacity region of MIMO broadcast channel has been made in [6], [7], [8], [9], and the exact characterization of the capacity region was found in [10]. It has been shown that the Dirty Paper Coding (DPC) achieves the capacity region. As a dual counterpart of the SIC for the MAC, the DPC for the BC successively removes the inter-user interference at transmitter provided that exact Channel State Information (CSI) is available at the transmitter side.

The main hindrance to the practical implementation of the DPC is its high complexity (see, for example, [11]) and its sensibility to the CSI, as shown by [12]. Low complexity solutions come naturally to the channel inversion based schemes, such as the Zero-Forcing (ZF) and the Minimum Mean Square Error (MMSE) precoders. The main idea is to inverse the channel matrix at transmitter in such a way that the inter-user interference is gone at the receiver side. However, direct application of such precoders either requires high transmit power or results in lower Signal to Noise Ratio (SNR) for a fixed transmit power, especially when the channel matrix is ill-conditioned. One of the workaround is to apply the lattice basis reduction at the transmitter side. The lattice reduction yields a better conditioned basis and a cubic constellation carved from the reduced basis has a much smaller average energy (see [13]). Another workaround is the vector perturbation scheme proposed by Hochwald *et al* [14]. Instead of sending symbol from the cubic constellation carved from the lattice of the inverse of the channel matrix, one can send any symbol from the coset of this symbol. The optimal choice can be decided by so-called sphere encoder. In both schemes, a modulo operation is involved at both the transmitter and receiver sides.

The performance of the vector perturbation scheme depends highly on the modulo function sensitivity to noise perturbation at low SNR. In our contribution, we propose to implement a new non-linear precoder based on the use of a more sophisticated constellation scheme called Periodically Flipped Constellation (PFC) at the encoder associated to a modified modulo function at the receiver to perform decoding. We show that this technique improves the performance at low SNR by reducing detection errors due to noise perturbation. Unlike in the original scheme in [14], the sphere encoder cannot be applied directly for the PFC case. We proposed therefore a general sphere encoder structure based on Schnorr-Euchner algorithm (see [15]) that works for a broad class of constellation including the PFC. Moreover, it turns out that the complexity of the general sphere encoder is practically the same as the original sphere encoder.

The rest of the paper is organized as follows. The system model and some basic assumptions are presented in section II. Section III introduces the proposed periodically flipped constellation scheme. The improvement that can be achieved with constellation contraction is detailed in section IV. More over some numerical simulation results are shown in this section to asset the performance gain compared to conventional scheme. Section V deals with the implementation of the proposed scheme. Finally, section VI concludes the paper.

II. Preliminaries

A. Notations

The notations used in this paper are as follows. Boldface lower case letters $\boldsymbol{v}$ denote vectors, boldface capital letters $\boldsymbol{M}$ denote matrices. $\mathcal{CN}$ represents the complex Gaussian random variable. $\mathbb{E}\left(X\right)$ is the mathematical expectation of random variable X. $\|\boldsymbol{v}\|$ stands for the Euclidean norm of vector $\boldsymbol{v}$. $\|\boldsymbol{H}\|_{\mathrm{F}}$ is the Frobenius norm of matrix $\boldsymbol{H}$ and $\boldsymbol{H}^{-1}$ is the inverse of a square matrix $\boldsymbol{H}$. $\mathbb{Z}^K[i]$ is the set of K length complex integer vectors. Finally, $\lfloor x \rfloor$ and $[x]$ denote respectively the floor and round operators.

B. System Model

We consider a broadcast channel with one source equipped with N_t antennas and K destinations each one having single antenna. For the ease of presentation, we assume that $K = N_t$, although the results can be extended straightforwardly to the case with $K < N_t$. The signal model is

$$\boldsymbol{y} = \boldsymbol{H}\frac{\boldsymbol{x}}{\sqrt{\gamma}} + \boldsymbol{z} \tag{1}$$

where $\boldsymbol{z} \in \mathcal{CN}(0, \mathbf{I})$ is the Additive White Gaussian Noise (AWGN), $\boldsymbol{H}$ is the channel matrix, and γ is the power normalization factor that does not depend on the transmitted message, but can depend on the channel $\boldsymbol{H}$. Here, we set

$$\gamma \triangleq \frac{\mathbb{E}(\|\boldsymbol{x}\|^2)}{P} \tag{2}$$

with P being the total transmission power of the source. Since the power of $\boldsymbol{y}$ can depend on both the source message and the channel $\boldsymbol{H}$, we impose that the expectation in (2) is only over the source message for a given channel realization $\boldsymbol{H}$.

C. Channel Inversion

A simple linear precoding scheme is the zero-forcing precoding (also denoted channel inversion) where

$$\boldsymbol{x} \triangleq \boldsymbol{H}^{-1}\boldsymbol{s} \tag{3}$$

with $\boldsymbol{s}$ being the vector of signals intended for different users. It is assumed that $\boldsymbol{s}$ belongs to a constellation carved from the translated lattice Λ defined by

$$\Lambda \triangleq \tau_c\mathbb{Z}^K[i] + \tau_c\frac{1+i}{2} \tag{4}$$

and is normalized in power, i.e. $\mathbb{E}(|s_i|^2) = 1$ for all i. τ_c is the minimum distance between two different points in the constellation. With the ZF precoding, the equivalent channel is

$$y_k = \frac{\sqrt{P}}{\|\boldsymbol{H}^{-1}\|_{\mathrm{F}}} s_k + z_k, \quad \forall\, k \tag{5}$$

Since we can approximate $\frac{1}{\|\boldsymbol{H}^{-1}\|_{\mathrm{F}}}$ by $\sigma_{\min}(\boldsymbol{H})$, the minimum singular value of $\boldsymbol{H}$, it can be deduced that this scheme suffers from significant loss in terms of power and diversity (of order 1 in Rayleigh fading channels).

D. Vector Perturbation

[Fig. 1 about here.]

A fix to this problem is to use non-linear precoding scheme. A vector version of the Tomlinson-Harashima precoding [16], [17] scheme, also known as the vector perturbation scheme, is proposed in [14] and is described briefly as follows (cf Fig. 1). This transmitted signal $\boldsymbol{x}$ is

$$\boldsymbol{x} = \boldsymbol{H}^{-1}(\boldsymbol{s} + \boldsymbol{p}(\boldsymbol{s})) \tag{6}$$

with $\boldsymbol{p}(\boldsymbol{s}) \in \mathcal{P}(\boldsymbol{s})$ being the perturbation vector. Thus, an obvious optimal choice of $\boldsymbol{p}$ is

$$\begin{aligned} \boldsymbol{p}^*(\boldsymbol{s}) &= \arg \min_{\boldsymbol{p} \in \mathcal{P}(\boldsymbol{s})} \|\boldsymbol{x}\|^2 \\ &= \arg \min_{\boldsymbol{p} \in \mathcal{P}(\boldsymbol{s})} \|-\boldsymbol{H}^{-1}\boldsymbol{s} - \boldsymbol{H}^{-1}\boldsymbol{p}\|^2. \end{aligned} \tag{7}$$

Note that the naive ZF scheme is a particular case of the above scheme, which can be seen by setting trivially $\mathcal{P}(\boldsymbol{s}) = \{0\}$. Therefore, the non-linear scheme is at least as good as the linear scheme. In [14], $\mathcal{P}(\boldsymbol{s})$ is set as a sub-lattice $\tau\mathbb{Z}^K[i]$ of the lattice $\tau_c\mathbb{Z}^K[i]$ independent of $\boldsymbol{s}$. τ is chosen in order to get a periodic extension of the original signal constellation. Thus, $\tau/\tau_c \in \mathbb{Z}$ and $\boldsymbol{s} + \boldsymbol{p}(\boldsymbol{s})$ belongs to a coset of $\tau\mathbb{Z}^K[i]$ determined by $\boldsymbol{s}$[1]. The received signal for each user being

$$y_k = \frac{s'_k}{\sqrt{\gamma}} + z_k, \quad s'_k \in \tau_c\mathbb{Z}[i] + \tau_c\frac{1+i}{2},$$

the receiver tries to decide the most probable coset. For a hard detector, the closest lattice point is first found and then is used to determine the representation $\hat{\boldsymbol{s}}$ of the coset by a mod-$\tau\mathbb{Z}^K[i]$ operation using the modulo function $f_\tau(\cdot)$ where $f_\tau(y) = y - \left\lfloor \frac{y+\tau/2}{\tau} \right\rfloor \tau$.

III. Periodically Flipped Constellations

As shown above, the conventional vector perturbation scheme restricts the possible perturbation vectors within the sub-lattice $\tau\mathbb{Z}^K[i]$. In this section, we show that the performance can be improved with another set of perturbation vectors. The motivation is shown by the following example.

[1] For a QAM signaling, it is readily shown that the cardinality of the constellation is $(\tau/\tau_c)^2$.

A. *Motivating example*

[Fig. 2 about here.]

For simplicity of demonstration, we consider, in this example, the special case of QPSK modulation. Suppose that the base station needs to transmit to some user k a symbol $d_k = -1 + i$. With the conventional vector perturbation scheme, a replicated constellation is used as an infinite extension of the original constellation (cf. Fig. 2(a)). Let us assume that another point (say, $-5 + i$) that is in the same coset turns out to minimize the transmit power and is chosen. If the noise happens to draw the received symbol outside the constellation as shown in Fig. 2(a), the receiver will make a wrong decision by searching the closest point in the constellation to the received symbol.

The situation can be improved with a better choice of perturbation set, i.e. a better infinite extension. The idea is shown in Fig. 2(b). Assume that $d_k = 1 + i$ is the information symbol. Instead of associating the information symbol with its periodically replicated counterparts, as in the previous case, the original constellation is successively flipped away. We call this constellation scheme the periodically flipped constellation. In this example, the transmitter finds that $-5 + i$ minimizes the transmit power over all the associated points of $d_k = 1 + i$ in the PFC. Now, with the same noise as in the previous case, the receiver can make a right decision by searching the closest point at the extended constellation, i.e. the PFC. Thus, the overall error rate performance is improved.

B. *Scheme definition*

Mathematically, a PFC of a K-dimensional QAM constellation $\mathcal{C}$ can be represented by

$$\left\{ \boldsymbol{s} + \boldsymbol{p}(\boldsymbol{s}) \;\middle|\; \boldsymbol{s} \in \mathcal{C},\; \boldsymbol{p}(\boldsymbol{s}) \in \mathcal{P}_{\text{PFC}}(\boldsymbol{s}) \right\}$$

with the set of perturbation vectors defined by

$$\mathcal{P}_{\text{PFC}}(\boldsymbol{s}) \triangleq \left\{ \boldsymbol{p} \;\middle|\; \begin{array}{r} \Re\{p_i\} \in (2\tau\mathbb{Z}) \cup (f(\Re\{s_i\}) + 2\tau\mathbb{Z}) \\ \forall i = 1, \ldots, K, \\ \Im\{p_i\} \in (2\tau\mathbb{Z}) \cup (f(\Im\{s_i\}) + 2\tau\mathbb{Z}) \\ \forall i = 1, \ldots, K \end{array} \right\} \tag{8}$$

where $f(s) = \tau - s$, $s \in \mathbb{R}$ is the flip function; $\Re\{x\}$ and $\Im\{x\}$ represent the real and imaginary parts of x, respectively. Since the above set is defined in a dimensional-wise manner, we can rewrite it as

$$\mathcal{P}_{\text{PFC}}(\boldsymbol{s}) \triangleq \left\{ \boldsymbol{p} \;\middle|\; \begin{array}{r} \Re\{p_i\} \in \mathcal{P}_{\text{PFC}}(\Re\{s_i\}) \\ \forall i = 1, \ldots, K, \\ \Im\{p_i\} \in \mathcal{P}_{\text{PFC}}(\Im\{s_i\}) \\ \forall i = 1, \ldots, K \end{array} \right\} \tag{9}$$

With the flipped replication, it is obvious that points at the border of the constellation enjoy a better protection. This is due to the fact that the number of neighbors that are at the minimum distance to any of these points is reduced by at least one[2]. For BPSK, the number of neighbors of minimum distance to any symbol is 1 for PFC compared to 2 in the conventional case. Similarly, this number is equal to 2 compared to 4 for QPSK and, to 2 or 3 compared to 4 for 16QAM.

At the receiver side, similar operation is performed as with the conventional vector perturbation schemes. More specifically, the closest lattice point is first found, and then is used to determine the representation in the original coset using a modified modulo function which corresponds to $-f_{\tau}(\cdot)$ when the closest point happens to be within one of the flipped constellations and $f_{\tau}(\cdot)$ if not. Hence, detection complexity remains the same as the conventional constellation with the PFC.

C. Numerical Results

For illustration, we consider the case of a broadcast channel with 4 transmit antennas and 4 selected single antennas users among a large number of users.

In Fig. 3, we compare for this antenna configuration the packet error rate for the conventional vector perturbation technique versus the PFC perturbation schemewhen a convolutional code $[$o133 o171$]$ is used for a packet size of 1kB. A MIMO OFDMA system, having N subcarriers, where $N = 512$, 1024, 2048 in practical standards is used. On each subcarrier, the channel is considered flat.

Knowing that channel coefficients on two subcarriers f_i and f_j such that $|f_i - f_j| \geq B_c$ (B_c being the coherence band) are uncorrelated, a frequency interleaver can be used to have independent channel on interleaved subcarriers. For each user, a cluster of 10 independent subcarriers is assigned. This kind of feature is usually promotted as "distributed subcarriers" in modern OFDMA systems such as IEEE 802.16e and 3GPP-LTE.

We can see that the PFC provides a gain of 1.5dB for QPSK at BER $= 10^{-2}$. Although this technique protects the border points in the constellation, there is no gain for higher order of modulations (16QAM and 64QAM). This loss in gain can be explained from the fact that the power normalization factors with replicated constellation is, in average, smaller than the one with periodically flipped constellation.

[Fig. 3 about here.]

We also observed (on uncoded performance curves not presented in this paper) that gains of PFC over the classical SE are observed only in the low SNR regime. This can be explained by the fact that the PFC enhances performance of sphere encoding against noise perturbation. At low SNR,

[2] In the QPSK example, the number of neighbors is reduced by two for all constellation points.

the probability that the error is due to noise perturbation is significant. However, at high SNR, the variance of the noise is very small, this error probability is reduced.

Let's remark, that there is no restriction on using the PFC with channel inversion. PFC can be also applied to the case of regularized sphere encoding [14].

IV. Improvement with Constellation Contraction

As stated in previous sections, the gain of the PFC over the standard periodically replicated constellation comes from the fact that constellation points at the border of the original constellation enjoy extra protection. As it will be shown in the following, such protection is excessive, since it can be reduced without impacting the performance. Meanwhile, reducing the protection may save transmit power.

A. Constellation Contraction

In this section, we propose to improve the performance of the vector perturbation scheme by introducing a contraction on the PFC. The PFC with contraction can be regarded as a generalized PFC, defined by an additional contraction factor α (Fig. 4). This idea is motivated by the following two observations. For simplicity of demonstration, we consider a BPSK modulation as an example.

[Fig. 4 about here.]

First, we analyze the impact of α at the receiver side. The PFC of a BPSK modulation is shown in Fig. 5. Assuming that "□" is transmitted, we have the following approximation of the union bound

$$\begin{aligned}
&\text{Prob}\{\text{error} \,|\, \square \,\text{is transmitted}\}\\
&\qquad\qquad\qquad\qquad \lesssim \text{PEP}\{\square \to \bigcirc_1\} + \text{PEP}\{\square \to \bigcirc_2\}\\
&\qquad\qquad\qquad\qquad = Q\left(\frac{d_{\min}}{\sqrt{2\sigma^2}}\right) + Q\left(\frac{(1+\alpha)d_{\min}}{\sqrt{2\sigma^2}}\right) \qquad (10)
\end{aligned}$$

where $\text{PEP}\{A \to B\}$ is the pair-wise error probability that A is confused with B at the receiver; $Q(x)$ is the Gaussian tail function; σ^2 is the variance of the AWGN. Since $Q(t)$ can be well approximated by $1/2\exp(-t^2/2)$, we know that

$$Q\left(\frac{d_{\min}}{\sqrt{2\sigma^2}}\right) \gg Q\left(\frac{(1+\alpha)d_{\min}}{\sqrt{2\sigma^2}}\right),$$

except for α small compared to 1. This remark implies that the error probability is not sensitive to α when α is large.

[Fig. 5 about here.]

The second observation is impact of α on the transmitter side. Without loss of generality, let us assume that the transmitter intends to send "□". With the PFC, the transmitter can choose in the PFC the "□" that minimizes the transmit power.

$$\boldsymbol{p}^*(\boldsymbol{s}) = \arg\min_{\boldsymbol{p}\in\mathcal{P}(\boldsymbol{s})} \|-\boldsymbol{H}^{-1}\boldsymbol{s} - \boldsymbol{H}^{-1}\boldsymbol{p}\|^2. \tag{11}$$

This is essentially a quantization problem where the minimum power can be seen as the quantization error of $-\boldsymbol{H}^{-1}\boldsymbol{s}$ in the irregular[3] lattice defined by $\boldsymbol{H}^{-1}$. As shown in Fig. 6, decreasing the value of α can reduce the size of quantization cell, which means a smaller quantization error (smaller required transmit power). On the other hand, increasing α provokes a larger transmit power in average.

[Fig. 6 about here.]

As a conclusion of the above observations, there is a trade-off between error probability at the receiver and the average transmit power for the generalized PFC scheme. This trade-off is realized by α. While the original PFC scheme fixes $\alpha = 1$, a relatively large value, intuition suggests that one should decrease α from 1. In the large α regime, the error probability is not sensitive to α, whereas the transmit power is sensitive to it. In this regime, the decrease of α saves the transmit power by paying a marginal penalty on the error probability. In the regime of small α, the error probability is very sensitive to α, whereas the transmit power is not sensitive to it. the decrease of α merely increases the error probability. The above arguments will be confirmed by the numerical results presented later on.

B. *Performance assessments*

By tuning the distance between replicated constellation, PFC performance are improved by 1.3dB for QPSK, 0.4dB for 16QAM and 0.3dB for 64QAM (see Fig. 7).

[Fig. 7 about here.]

V. Implementation of General Sphere Encoder

A. *General closest point problem*

For $\boldsymbol{G} \in \mathbb{R}^{m\times n}$ and $\boldsymbol{y} \in \mathbb{R}^m$, let us consider the minimization

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}\in\mathbb{Z}^n} \|\boldsymbol{y} - \boldsymbol{G}\boldsymbol{x}\|^2. \tag{12}$$

The set $\Lambda(\boldsymbol{G}) \triangleq \{\boldsymbol{G}\boldsymbol{x} : \boldsymbol{x} \in \mathbb{Z}^n\}$ is a n-dimensional lattice in $\mathbb{R}^m$. The search in (12) for the *closest lattice point* to a given point $\boldsymbol{y}$ has been widely investigated in lattice theory.

[3] It is a lattice only when $\boldsymbol{p}$ is in $c\mathbb{Z}^n$ with c being some scalar. Here, $\mathcal{P}$ defined by the PFC is not a subset of a lattice in the general case.

More recently, Agrell *et al* [15] proposed the use of the Schnorr-Euchner refinement of the Pohst enumeration [18] in the *closest lattice point* search. They further concluded, based on numerical results, that the Schnorr-Euchner enumeration is more efficient than the Viterbo-Boutros implementation [19].

In the rest of this section, we will first present briefly the principle of the Schnorr-Euchner strategy. A detailed version can be found in [15]. Then, it will be adapted to the construct a sphere encoder with PFC.

B. Closest lattice point search with Schnorr-Euchner algorithm

The Schnorr-Euchner strategy is in fact a combination of the Pohst strategy and the Babai nearest-plane algorithm [20]. Let us denote $\boldsymbol{G} \in \mathbb{R}^{m \times n}, m \geq n$ the generator matrix of lattice $\Lambda(\boldsymbol{G})$, $\boldsymbol{y} \in \mathbb{R}^m$ a vector to be decoded in the lattice $\Lambda(\boldsymbol{G})$.

1) Recursive formulation: Let us define $\boldsymbol{G}_n \triangleq \boldsymbol{G}$ and rewrite it as

$$\boldsymbol{G}_n = \begin{bmatrix} \boldsymbol{G}_{n-1} & \boldsymbol{v_n} \end{bmatrix}$$

where $\boldsymbol{G}_{n-1}$ is an $m \times (n-1)$ matrix consisting of the $n-1$ first columns of $\boldsymbol{G}$ and $\boldsymbol{v_n} = \boldsymbol{v}_{\|} + \boldsymbol{v}_{\perp}$, with $\boldsymbol{v}_{\|}$ and $\boldsymbol{v}_{\perp}$ in the column space and the null space of $\boldsymbol{G}_{n-1}$, respectively. The lattice points are defined by the following set

$$\begin{aligned}\Lambda(\boldsymbol{G}_n) &= \{\boldsymbol{G}_n \boldsymbol{u}_n : \ \boldsymbol{u}_n \in \mathbb{Z}^n\} \\ &= \bigcup_{u_n=-\infty}^{+\infty} \left\{\boldsymbol{c} + u_n \boldsymbol{v}_{\|} + u_n \boldsymbol{v}_{\perp} : \boldsymbol{c} \in \Lambda(\boldsymbol{G}_{n-1})\right\}\end{aligned}$$

which is basically a stack of $(n-1)$-dimensional translated sub-lattices from $\Lambda(\boldsymbol{G}_{n-1})$. The $(n-1)$-dimensional hyperplanes that contain these sub-lattices will be called $(n-1)$-dimensional *layers*. $u_n \in \mathbb{Z}$ can be seen as the index of the $(n-1)$-dimensional layer. We can write any point $\boldsymbol{y} \in \mathbb{R}^m$ in the following form

$$\boldsymbol{y} = \boldsymbol{v}_{\perp} \xi_n + \boldsymbol{G}_{n-1} \boldsymbol{\xi}_{n-1}$$

with $\xi_n \in \mathbb{R}$ and $\boldsymbol{\xi}_{n-1} \in \mathbb{R}^{n-1}$. Hence, searching the closest lattice point in $\Lambda(\boldsymbol{G}_n)$ to a given point $\boldsymbol{y}$ can be formulated as follows:

$$\begin{aligned}\hat{\boldsymbol{u}} &= \arg \min_{\boldsymbol{u}_n \in \mathbb{Z}^n} \|\boldsymbol{y} - \boldsymbol{G}_n \boldsymbol{u}_n\|^2 \\ &= \arg \min_{\boldsymbol{u}_n \in \mathbb{Z}^n} \left\{ d_n(u_n) + \|y_{n-1}(u_n) - \boldsymbol{G}_{n-1} \boldsymbol{u}_{n-1}\|^2 \right\} \qquad (13)\end{aligned}$$

where $d_n(u_n) = |\xi_n - u_n|^2 \|\boldsymbol{v}_{\perp}\|^2$ and $y_{n-1}(u_n) = \left(\boldsymbol{G}_{n-1}\boldsymbol{\xi}_{n-1} - \boldsymbol{v}_{\|} u_n\right)$ In the last equation, the first term stands for the distance between the given point $\boldsymbol{y}$ and the layer defined by u_n, while the second term is the distance between the projection of $\boldsymbol{y}$ on the translated $(n-1)$-sub-lattice and a point in this sub-lattice defined by $\boldsymbol{u}_{n-1}$.

2) Tree structure: Therefore, finding the closest lattice point in a n-dimensional lattice consists of finding the closest points in all translated $(n-1)$-dimensional sub-lattices. The recursive formulation implies a tree structure that depends only on the generator matrix $\boldsymbol{G}$ of the lattice.

A tree node at the i^{th} level is labeled by $(u_{n-i+1}, \ldots, u_n)$ and represents a specific translation of a $(n-i)$-dimensional sub-lattice. In particular, a leaf node is a lattice point. The weight of each tree node is defined by the distance between the given point $\boldsymbol{y}$ and the translated sub-lattice related to the node. Hence, the goal of the tree search is to find the leaf node with the smallest weight.

3) Enumeration with non-decreasing distance: The Schnorr-Euchner enumeration is a depth first tree search algorithm. A distinguished property of the enumeration is that the tree nodes in the same level are ordered with increasing weight. That is, a node with a smaller weight is always visited before the others. Since the weights of the tree nodes on the level i can be written as $w_i = w_{i-1} + d_i(u_i)$ differ only on $d_i(u_i)$, as implied by (13), ordering the nodes in terms of w_i is easy. Assuming $\xi_n \leq [\xi_n]$, it is obvious that

$$u_n = [\xi_n], [\xi_n] - 1, [\xi_n] + 1, \ldots \tag{14}$$

is a non-decreasing order in terms of w_i. The ordering for the case $\xi_n > [\xi_n]$ can be deduced similarly.

With this ordering constraint, the complexity is significantly reduced (see [15] for more details). The pseudo code of the algorithm is provided below.

C. Sphere encoder implementations for PFC

In this section, two implementations of the sphere encoder with PFC are issued. The first solution is the composition of 2 sphere encoders, and the others is unique sphere encoders based on a modified Schnorr Euchner. It is clear that using a unique Schnorr Euchner on the resulting lattice instead of the decompositions of 2 sphere encoders reduces significantly the complexity as shown in [21].

1) Solution 1: composition of sphere encoders: Since $\mathcal{P}_{\text{PFC}}(\boldsymbol{s})$ can be seen as a union of 2^{2K} shifted sub-lattices $2\tau\mathbb{Z}^K[i]$, finding the closest point in $\mathcal{P}_{\text{PFC}}(\boldsymbol{s})$ can be implemented by finding the closest points in each of the shifted sub-lattices and then taking the one with minimum distance. In each shifted sub-lattice, a standard sphere decoder can be used for the research. Therefore, the complexity of the composition is that of the sphere decoder multiplied by a factor 2^{2K}. This exponential time complexity becomes unacceptable for a large number of users K.

2) Solution 2: modified Schnorr-Euchner algorithm: A more efficient way is to modify the sphere decoder in such a way that it can work directly on the set $\mathcal{P}_{\text{PFC}}(\boldsymbol{s})$ that does not have the lattice property in general. As it will be shown in the following, this can be worked around since each dimension of any point in $\mathcal{P}_{\text{PFC}}(\boldsymbol{s})$ belongs to a manageable set (we will get back to this in the end of the section).

Algorithm 1 Schnorr-Euchner search for the closest lattice point

Require: a $m \times n$ real matrix $\boldsymbol{G}$ with $m \geq n$, a m-dimensional vector $\boldsymbol{y} \in \mathbb{R}^n$ to decode in the lattice $\Lambda(\boldsymbol{G})$;

Ensure: a n-dimensional vector $\hat{\boldsymbol{u}}$ ($\in \mathbb{Z}^n$) such that $\boldsymbol{G}\hat{\boldsymbol{u}}$ is the closest lattice point to $\boldsymbol{y}$.

```
 1: $k \leftarrow n$
 2: $\boldsymbol{G}_k \leftarrow \boldsymbol{G}$
 3: $\boldsymbol{G}_{k-1} \leftarrow \boldsymbol{G}_k(:, 1:k-1)$ and $\boldsymbol{v}_k \leftarrow \boldsymbol{G}_k(:, k)$
 4: $\boldsymbol{v}_\parallel \leftarrow (\boldsymbol{G}_{k-1}\boldsymbol{G}_{k-1}^{+})\boldsymbol{v}_k$, $\boldsymbol{v}_\perp \leftarrow \boldsymbol{v}_k - \boldsymbol{v}_\parallel$, and $x_k \leftarrow \frac{\boldsymbol{v}_\perp{}^\dagger \boldsymbol{y}}{\|\boldsymbol{v}_\perp\|^2}$
 5: $u_k \leftarrow$ round$(x_k)$
 6: step$_k \leftarrow$ sign$(x_k - u_k)$
 7: $\Delta_k \leftarrow \|\boldsymbol{v}_\perp\|(x_k - u_k)$
 8: dist$_k \leftarrow 0$
 9: while true do
10:     newdist $\leftarrow$ dist$_k + \Delta_k^2$
11:     if newdist $<$ bestdist then
12:         if $k \neq 1$ then                                  ▷ visiting intermediate node
13:             $k \leftarrow k - 1$                              ▷ keep going down the tree
14:             dist$_k \leftarrow$ newdist                        ▷ update current dist.
15:             $\boldsymbol{y} \leftarrow (\mathbf{I} - \boldsymbol{v}_\perp \boldsymbol{v}_\perp{}^{+})\,\boldsymbol{y}$          ▷ proj. to subspace
16:             $\boldsymbol{y} \leftarrow \boldsymbol{y} - \boldsymbol{v}_\parallel u_k$                   ▷ horizontal shift
17:             $\boldsymbol{G}_{k-1} \leftarrow \boldsymbol{G}_k(:, 1:k-1)$
18:             $\boldsymbol{v}_k \leftarrow \boldsymbol{G}_k(:, k)$
19:             $\boldsymbol{v}_\parallel \leftarrow (\boldsymbol{G}_{k-1}\boldsymbol{G}_{k-1}^{+})\boldsymbol{v}_k$
20:             $\boldsymbol{v}_\perp \leftarrow \boldsymbol{v}_k - \boldsymbol{v}_\parallel$ and $x_k \leftarrow \frac{\boldsymbol{v}_\perp{}^\dagger \boldsymbol{y}}{\|\boldsymbol{v}_\perp\|^2}$
21:             $u_k \leftarrow$ round$(x_k)$
22:             step$_k \leftarrow$ sign$(x_k - u_k)$
23:             $\Delta_k \leftarrow \|\boldsymbol{v}_\perp\|(x_k - u_k)$
```

As is shown in section V-B3, the advantage of the Schnorr-Euchner enumeration is its property of the non-decreasing ordering in each dimension. In the lattice case, each real dimension is a scaled version of the lattice $\mathbb{Z}$ and the ordering is easily done by (14). The implementation of the iterator is shown in algorithm 1 at line 5, 6, 21, 22, 28, 29, 37, and 38. These lines realize the updating of the next point to visit and the updating of the step. The rest of the algorithm is independent of the

Algorithm 2 Schnorr-Euchner search for the closest lattice point (Algorithm I cont'd)

```
24:         else                                          ▷ a leaf node has been visited
25:             $\hat{\boldsymbol{u}} \leftarrow \boldsymbol{u}$                                ▷ it is the best point for now
26:             bestdist ← newdist;
27:             $k \leftarrow k + 1$                              ▷ go back up, keep searching
28:             $u_k \leftarrow u_k + \text{step}_k$
29:             $\text{step}_k \leftarrow -\text{step}_k - \text{sign}(\text{step}_k)$
30:             $\Delta_k \leftarrow \|\boldsymbol{v}_\perp\|(x_k - u_k)$
31:         end if
32:     else                                   ▷ current point has a larger distance
33:         if $k = n$ then
34:             break                                                 ▷ return $\hat{\boldsymbol{u}}$
35:         else
36:             $k \leftarrow k + 1$                              ▷ go back up, keep searching
37:             $u_k \leftarrow u_k + \text{step}_k$
38:             $\text{step}_k \leftarrow -\text{step}_k - \text{sign}(\text{step}_k)$
39:             $\Delta_k \leftarrow \|\boldsymbol{v}_\perp\|(x_k - u_k)$
40:         end if
41:     end if
42: end while
```

structure of the set. As an abstraction, lines 5 and 6 can be replaced by the following lines

$$\text{Iter}(k).\text{update}(x_k),$$

$$u_k \leftarrow \text{Iter}(k).\text{next}().$$

Similarly, lines 21 and 22 are replaced by exactly the same lines above. On the other hand, lines 28 and 29 are replaced by

$$u_k \leftarrow \text{Iter}(k).\text{next}().$$

Again, the same line is used to replace lines 37 and 38. Note that there is one "Iterator" per dimension and it is updated only when we go down from a higher dimension, which is the case for lines 5 and 28. The update positions the point x_k and then determines the ordering for non-decreasing distance in the dimension k. Thus, the current point and the step between the current point and the next point are initialized each time an update is performed. The function next() sets the next point as current point, updates the step, and then returns the current point.

As a result, what we need is to re-implement the functions update() and next() for when the PFC is used in place of the conventional constellation. For convenience of exposition, we consider real lattices in the following. It can be shown that this assumption is without loss of generality because the complex signal model can be easily converted to a real signal model with the complex to real field embedding. Furthermore, the considered model is a scaled version of the original model in such a way that the distance between the adjacent constellation points is 1. The PFC without contraction is assumed in the first place. The necessary modifications with respect to the generalized PFC will be provided at the end of the section.

It should be noted that an initialization stage is needed for the iterators at the beginning of the proposed sphere encoding, as shown in algorithm 3.

Algorithm 3 Initialization of iterators: init(q, $\boldsymbol{s}$)

```
Require: an integer $q$ and a $2K \times 1$ vector $\boldsymbol{s}$ with $s_i \in [0, q-1]$
for $k = 1 : 2K$ do
    Iter$(k).q \leftarrow q$
    Iter$(k).s \leftarrow s_k$
end for
```

An implementation of the updating of an iterator is shown in algorithm 4. An example of a PFC of a BPSK constellation is illustrated in Fig. 8.

[Fig. 8 about here.]

In algorithm 4, two iterator modes have been defined according to the position of ξ. The iterator decides the next point to visit in each mode.

This is implemented in the function next(), as shown in algorithm 5.

In Fig. 9, the order in which the iterator visits the points is shown for the two modes.

[Fig. 9 about here.]

It is clear that the re-implementation of the iterator does not increase the order of the complexity of the Schnorr-Euchner algorithm. This is due to the manageable structure of the set in each real dimension, thanks to which the iterator can be implemented in low complexity.

The implementation of the sphere encoder with the generalized PFC is in exactly the same way as that with the original PFC. The only modification is on the iterator, since the PFC depends on α now. The new implementation of the functions init() and Iterator.update() is shown in algorithm 6 and algorithm 7.

Algorithm 4 Update an iterator: Iter.update(ξ)

```
Require: a real number $\xi$
Iter.counter $\leftarrow 0$
Iter.current $\leftarrow \xi$
$a \leftarrow \left\lfloor \frac{\xi}{2*\text{Iter}.q} \right\rfloor * 2 * \text{Iter}.q$
$b \leftarrow \left\lfloor \frac{\xi+2*\text{Iter}.s+1}{2*\text{Iter}.q} \right\rfloor * 2 * \text{Iter}.q - (2 * \text{Iter}.s + 1)$
$\tilde{a} \leftarrow \max\{a, b\}$, $\tilde{b} \leftarrow \min\{a, b\}$
Iter.$d_1 \leftarrow 2 * \text{Iter}.q - (\tilde{a} - \tilde{b})$
Iter.$d_2 \leftarrow \tilde{a} - \tilde{b}$
if $\xi \leq \tilde{b} + \text{Iter}.q$ or $\xi \geq \tilde{a} + \text{Iter}.q$ then
    Iter.mode $\leftarrow 1$
else
    Iter.mode $\leftarrow 0$
end if
if $\xi > \frac{\tilde{a}+\tilde{b}}{2} + \text{Iter}.q$ then
    Iter.step $\leftarrow \tilde{b} + 2 * \text{Iter}.q - \xi$
else
    Iter.step $\leftarrow \tilde{a} - \xi$
end if
```

Algorithm 5 Find the next point to visit: Iter.next()

```
Ensure: a real number
Iter.current $\leftarrow$ Iter.current $+$ Iter.step;
if Iter.mode $= 0$ then
    if Iter.counter mod $4 = 0$ then
        if Iter.counter $= 0$ then
            Iter.step $\leftarrow -(\text{sign(Iter.step)}) * \text{Iter.}d_1$
        else
            Iter.step$\leftarrow -$Iter.step $- (\text{sign(Iter.step)}) * \text{Iter.}d_1$
        end if
    else if Iter.counter mod $4 = 1$ then
        Iter.step $\leftarrow -$Iter.step $- (\text{sign(Iter.step)}) * \text{Iter.}d_2$
    else if Iter.counter mod $4 = 2$ then
        Iter.step $\leftarrow -$Iter.step $- (\text{sign(Iter.step)}) * \text{Iter.}d_2$
    else
        Iter.step $\leftarrow -$Iter.step $- (\text{sign(Iter.step)}) * \text{Iter.}d_1$
    end if
else
    if Iter.counter mod $2 = 0$ then
        Iter.step $\leftarrow (\text{sign(Iter.step)}) * \text{Iter.}d_2$
    else
        $f \leftarrow ((\text{Iter.counter} + 1)/2) * (\text{Iter.}d_1 + \text{Iter.}d_2) - \text{Iter.}d_2$
        Iter.step $\leftarrow -$Iter.step $- (\text{sign(Iter.step)}) * f$
    end if
end if
Iter.counter $\leftarrow$ Iter.counter $+ 1$
return Iter.current
```

Algorithm 6 Initialization of iterators: init(q, $\boldsymbol{s}$)

```
Require: an integer $q$ and a $2K \times 1$ vector $\boldsymbol{s}$ with $s_i \in [0, q-1]$
for $k = 1 : 2K$ do
    Iter$(k).q \leftarrow q$
    Iter$(k).s \leftarrow s_k$
    Iter$(k).\alpha \leftarrow \alpha$
    Iter$(k).Q \leftarrow q - 1 + \alpha$
end for
```

VI. Conclusions

In this paper, we present a new non-linear precoding technique for the multi-user MIMO downlink channel in the broadband wireless context. This new precoding scheme consists in choosing the set of perturbation vectors in a periodically flipped constellation rather than the replicated constellation as in the case of conventional sphere encoding scheme.

Algorithm 7 Update an iterator: Iter.update(ξ)

```
Require: a real number $\xi$
Iter.counter $\leftarrow 0$
Iter.current $\leftarrow \xi$
$a \leftarrow \left\lfloor \frac{\xi}{2*\text{Iter}.Q} \right\rfloor 2*\text{Iter}.Q$
$b \leftarrow \left\lfloor \frac{\xi+2*\text{Iter}.s+\text{Iter}.\alpha}{2*\text{Iter}.Q} \right\rfloor 2*\text{Iter}.Q - (2*\text{Iter}.s+\text{Iter}.\alpha)$
$\tilde{a} \leftarrow \max\{a,b\}$, $\tilde{b} \leftarrow \min\{a,b\}$
Iter.$d_1 \leftarrow 2*\text{Iter}.Q - (\tilde{a}-\tilde{b})$
Iter.$d_2 \leftarrow \tilde{a}-\tilde{b}$
if $\xi \leq \tilde{b}+\text{Iter}.Q$ or $\xi \geq \tilde{a}+\text{Iter}.Q$ then
    Iter.mode $\leftarrow 1$
else
    Iter.mode $\leftarrow 0$
end if
if $\xi > \frac{\tilde{a}+\tilde{b}}{2}+\text{Iter}.Q$ then
    Iter.step $\leftarrow \tilde{b}+2*\text{Iter}.Q-\xi$
else
    Iter.step $\leftarrow \tilde{a}-\xi$
end if
```

We show that this scheme, by enhancing the protection of outer constellation points, improves the performance when low order QAM constellations are used. For higher order QAM (e.g. 64QAM) constellations, the energy loss of the PFC compared to the standard periodically replicated constellation can be recovered using the generalized PFC with a finely tuned compression. By introducing the contraction on the constellation distance, we show that this can provide a better trade-off between the PFC resistance and the energy efficiency.

Finally, we propose a practical implementation for the sphere encoding with PFC based on the Schnorr-Euchner algorithm. The modified Schnorr-Euchner benefits from the manageable structure of the PFC to find the closest point in this type of lattice. Moreover, this modified algorithm takes into account the tuning parameter in order to adjust the distance between neighboring replicated and flipped constellations.

References

[1] S. Hamiti, "The draft IEEE 802.16m system description document," IEEE, Tech. Rep. IEEE 802.16m-08/003r4, July 2008.

[2] H. Holma and A. Toskala, *WCDMA for UMTS: HSPA Evolution and LTE.* John Wiley, 2007.

[3] I. E. Telatar, "Capacity of multi-antenna Gaussian channels," *European Transactions on Telecommunications*, vol. 10, pp. 585–595, 1999.

[4] G. Foschini and M. Gans, "On Limits of Wireless Communications in a Fading Environment when Using Multiple Antennas," *Wireless Personal Communications*, vol. 6, pp. 311–335, 1998.

[5] R. Ahlswede, "Multi-way communication channels," in *Proc. 2nd. Int. Symp. Information Theory*, 1971.

[6] G. Caire and S. Shamai, "On the achievable throughput of a multiantenna Gaussian broadcast channel," *IEEE Trans. on Information Theory*, vol. 49, pp. 1691–1706, July 2003.

[7] P. Viswanath and D. N. C. Tse, "Sum capacity of the vector Gaussian broadcast channel and uplink-downlink duality," *IEEE Trans. on Information Theory*, vol. 49, pp. 1912–1921, Aug. 2003.

[8] S. Vishwanath, N. Jindal, and A. Goldsmith, "Duality, achievable rates, and sum-rate capacity of Gaussian MIMO broadcast channels," *IEEE Trans. on Information Theory*, vol. 49, pp. 2658–2668, Oct. 2003.

[9] W. Yu and J. Cioffi, "Sum capacity of Gaussian vector broadcast channels," *IEEE Trans. on Information Theory*, vol. 50, pp. 1875–1892, Sept. 2004.

[10] H. Weingarten, Y. Steinberg, and S. Shamai, "The Capacity Region of the Gaussian Multiple-Input Multiple-Output Broadcast Channel," *IEEE Trans. on Information Theory*, vol. 52, pp. 3936–3964, Sept. 2006.

[11] U. Erez and S. Ten Brink, "A Close-to-Capacity Dirty Paper Coding Scheme," *IEEE Trans. on Information Theory*, vol. 51, pp. 3417–3432, Oct. 2005.

[12] S. Yang and J. C. Belfiore, "The impact of channel estimation error on the DPC region of the two-user Gaussian broadcast channel," in *43rd Annual Allerton Conference on Communication, Control, and Computing*, Monticello, IL, Sept. 2005.

[13] M. Taherzadeh, A. Mobasher, and A. K. Khandani, "Communication Over MIMO Broadcast Channels Using Lattice-Basis Reduction," *IEEE Trans. on Information Theory*, vol. 53, pp. 4567–4582, Dec. 2007.

[14] B. M. Hochwald, C. B. Peel, and A. Swindlehurst, "A vector-perturbation technique for near-capacity multiantenna multiuser communication part II: Perturbation," *IEEE Trans. on Communications*, vol. 53, no. 3, pp. 537–544, Mar. 2005.

[15] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, "Closest point search in lattices," *IEEE Trans. on Information Theory*, vol. 48, no. 8, pp. 2201–2214, Aug. 2002.

[16] M. Tomlinson, "New automatic equaliser employing modulo arithmetic," *Electronics Letters*, vol. 7, pp. 138–139, Mar. 1971.

[17] H. Harashima and H. Miyakawa, "Matched-transmission technique for channels with intersymbol interference," *IEEE Trans. on Communications*, vol. 20, no. 4, pp. 774–780, Aug. 1972.

[18] M. Pohst, "On the computation of lattice vectors of minimal length, successive minima and reduced basis with applications," *ACM SIGSAM*, vol. 15, pp. 37–44, 1981.

[19] E. Viterbo and J. Boutros, "A universal lattice code decoder for fading channel," *IEEE Trans. on Information Theory*, vol. 45, no. 10, pp. 1639–1642, July 1999.

[20] L. Babai, “On Lovász lattice reduction and the nearest lattice point problem,” *Combinatorica*, vol. 6, no. 1, pp. 1–13, 1986.

[21] B. Cerato, G. Masera and E. Viterbo, “Decoding the Golden Space-Time Trellis Coded Modulation,” *IEEE Comunication letters*, vol. 12, no. 8, pp. 569–571, August 2008.

List of Figures

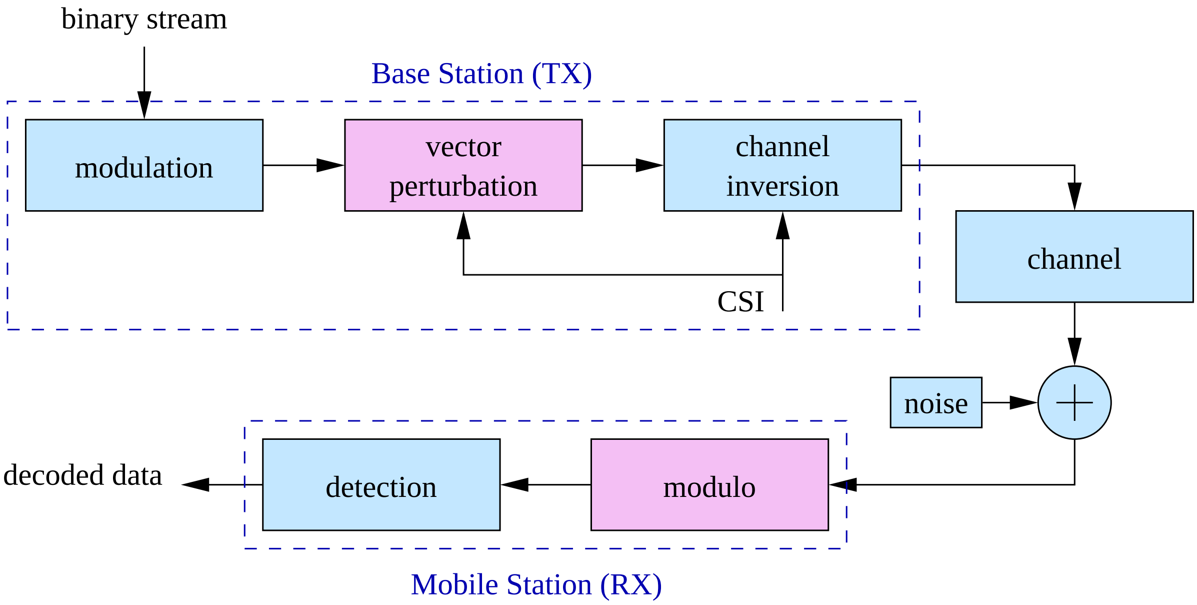
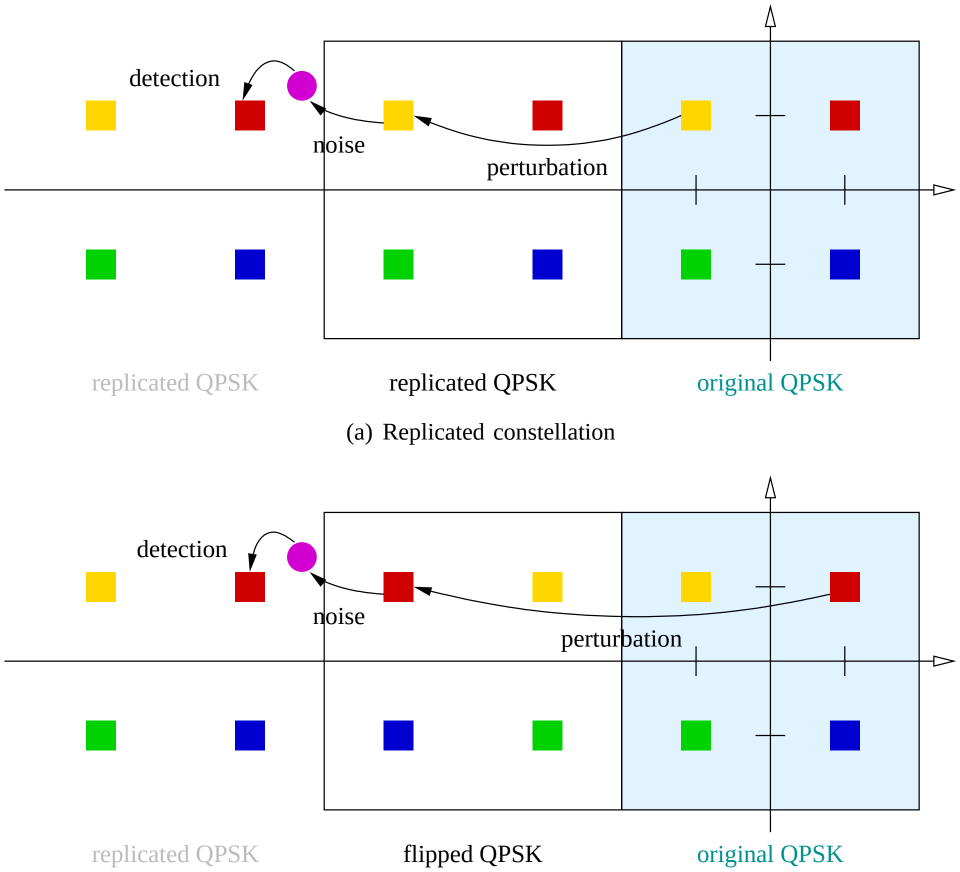


Fig. 1. Non-linear precoding with vector perturbation.



(a) Replicated constellation

(b) Periodically flipped constellation

Fig. 2. Extra protection provided by periodically flipped constellation.

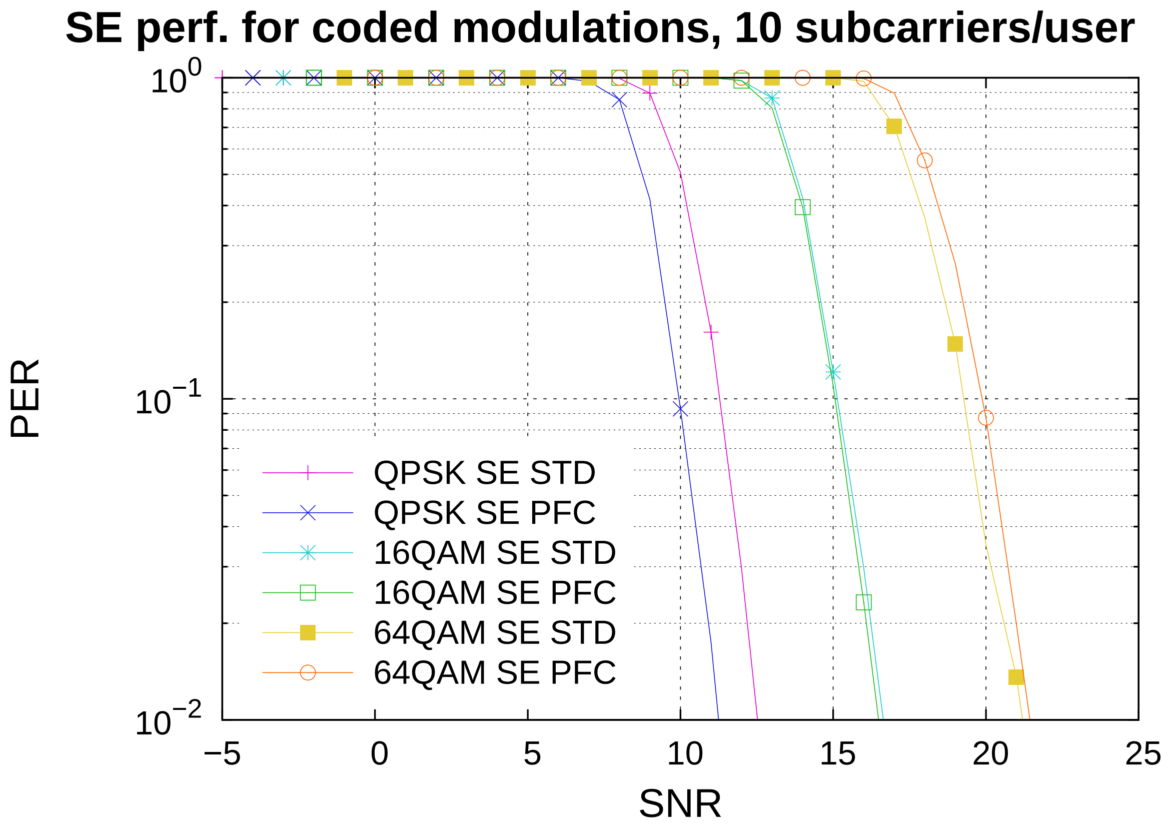


Fig. 3. Coded performance of PFC sphere encoder versus standard sphere encoder

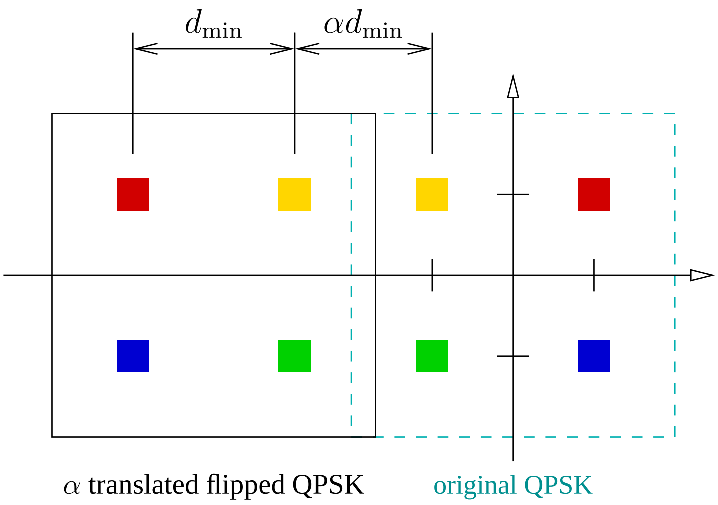


Fig. 4. Sphere Encoder with α-contraction.

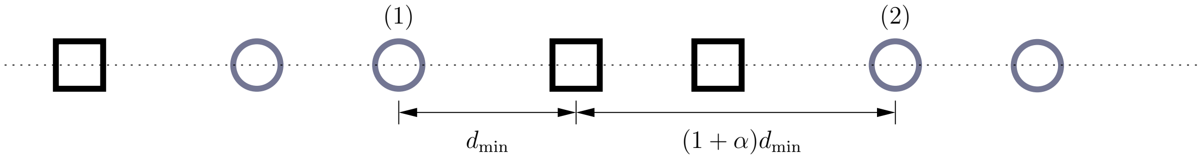


Fig. 5. PFC of BPSK: illustration of closest neighbors and the corresponding distances.

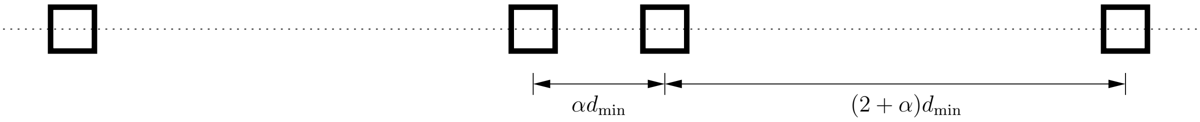


Fig. 6. PFC of BPSK: quantization cell.

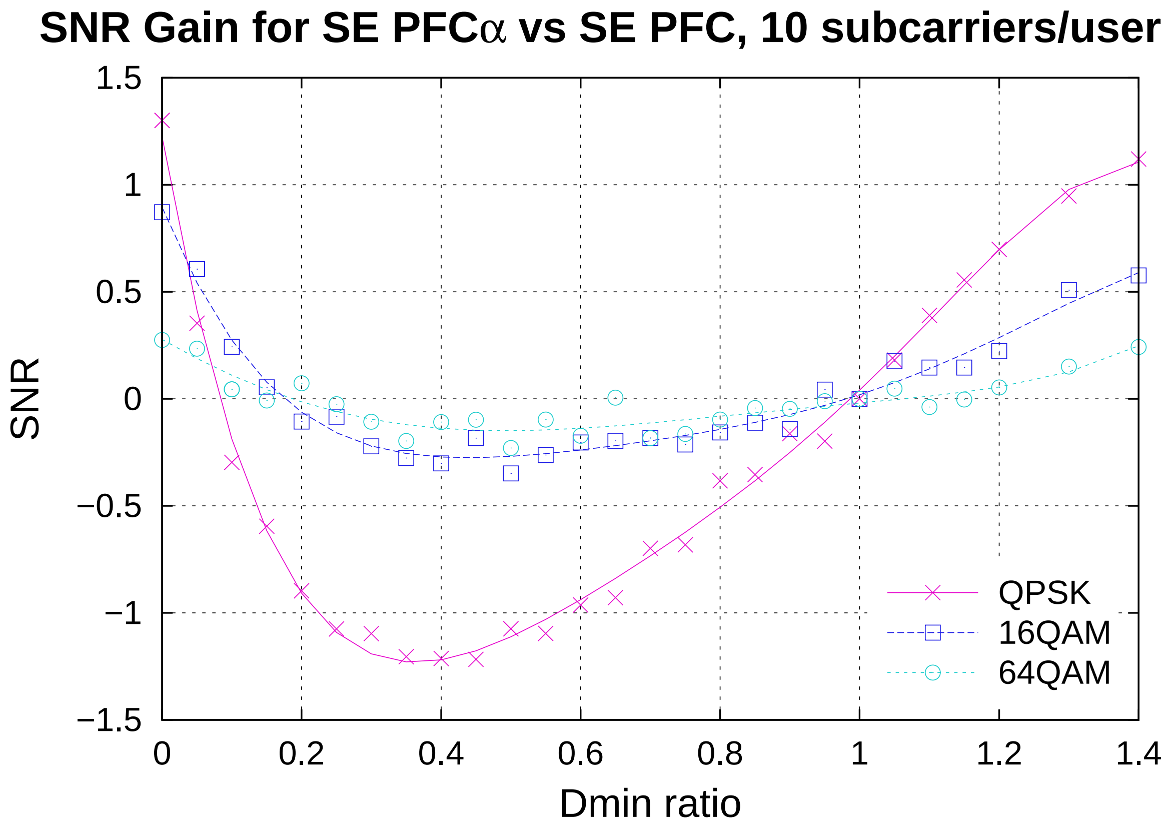


Fig. 7. SNR gain of PFC-α versus PFC at a PER target of 1%.

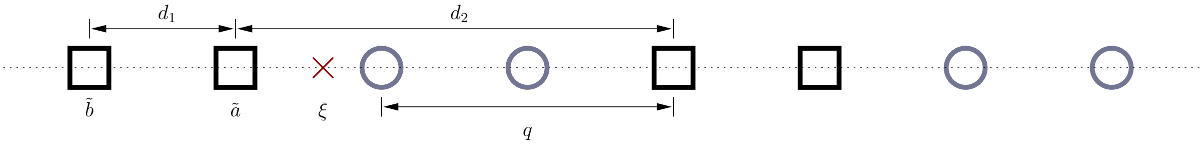


Fig. 8. An example of PFC with BPSK: updating the iterator.

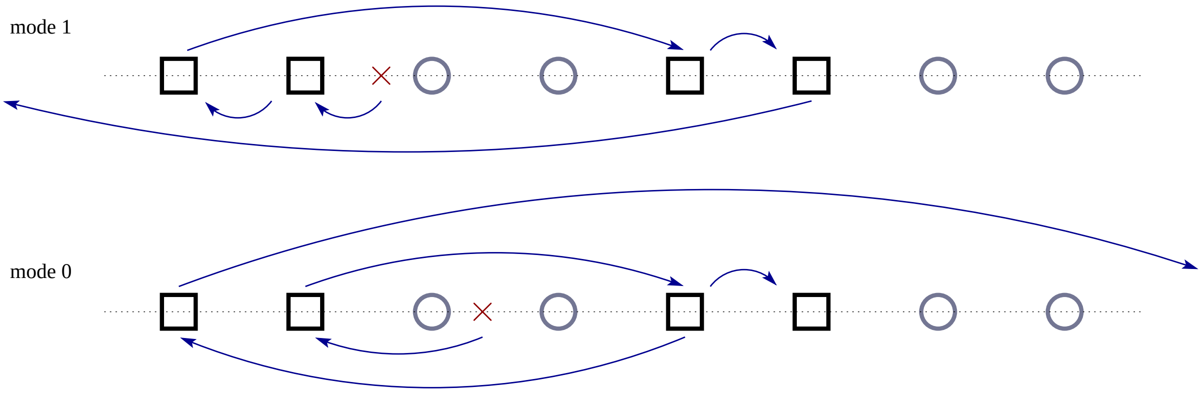


Fig. 9. An example of PFC with BPSK: visiting order.