Block Noncoherent Detection of Hexagonal QAM

Robby G. McKilliam[1], Daniel J. Ryan[2], I. Vaughan L. Clarkson[1] and Iain B. Collings[3]

[1] School of Information Technology and Electrical Engineering, The University of Queensland, QLD 4067, AUSTRALIA.
{robertm,v.clarkson}@itee.uq.edu.au

[2] Dept. Elec. and Telecommuns. (IET), Norwegian Univ. of Science and Technology (NTNU) danryan@ieee.org

[3] Wireless Technologies Laboratory, CSIRO ICT Centre, NSW 2113, AUSTRALIA iain.collings@csiro.au

***Abstract*—We consider block noncoherent detection of hexagonal quadrature amplitude modulation (QAM). We focus on hexagonal constellations generated from a *Voronoi code*. We find that these constellations are particularly well suited to noncoherent detection because they avoid most of the identifiability problems that occur with more traditional constellations. We describe a fast, approximate, noncoherent detection algorithm with statistical performance that is almost indistinguishable from an optimal, brute force approach, but with significantly smaller computational requirements. The performance is close to that of coherent detection when the *block length* is sufficiently large.**

***Index Terms*—Noncoherent detection, Hexagonal QAM, Voronoi code, lattice theory, lattice decoding.**

I. INTRODUCTION

This paper considers noncoherent detection of hexagonal quadrature amplitude modulation (QAM) transmitted in the block fading channel. Noncoherent transmission is applicable to systems exhibiting small coherence intervals where regular transmission of training symbols is wasteful. Such situations typically occur in mobile communications [1, 2]. Moreover, as indicated by Chen *et al.* [2] standard coherent detection is, in a sense, inherently suboptimal because it only uses the energy of a small number of training symbols for channel estimation, rather than also exploiting the (typically larger) energy in the unknown data symbols. Other applications of noncoherent detection include recovery from deep fades in pilot-symbol assisted modulation based schemes, eavesdropping, and non-data-aided channel estimation [1].

In block noncoherent detection it is assumed that the channel is constant over a number of consecutive data symbols called the *block length*. The channel and the data symbols are then estimated simultaneously. For phase-shift-keying (PSK) constellations the estimation problem is computationally simple, requiring only $O(N \log N)$ arithmetic operations [3, 4] or $O(N)$ operations [5] depending on the objective function used where N is the block length. These estimators rely heavily on the fact that PSK symbols have constant amplitude. The estimation problem is more complicated for amplitude varying (multi level) constellations such as square, rectangular and hexagonal QAM. The brute force approach requires complexity that is exponential in N. Motedayen and Anastasopoulos [6] describe a general approach that reduces the complexity to polynomial time. Specific details were presented for the cases of square QAM over a phase noncoherent channel (i.e. unknown channel phase, but known channel amplitude), and for PSK over a fading channel with coding. For the fully noncoherent channel (i.e. unknown phase and amplitude) with square and rectangular QAM, an exact algorithm requiring $O(N^3)$ operations and an approximate algorithm requiring $O(N^2 \log N)$ operations was described by Ryan *et. al.* [1]. The exact algorithm was latter improved to require only $O(N^2 \log N)$ operations in [7].

The existing literature mostly deals with block noncoherent detection of PSK and square QAM. There is less work on hexagonal QAM. It has been suggested that noncoherent detection of hexagonal QAM can be represented as a *simultaneous Diophantine approximation* problem [8, 9]. The detection problem can then be solved using general purpose algorithms such as the sphere decoder [10] or Babai's algorithm [11]. The sphere decoder is more accurate, but is exponentially complex in N. Babai's algorithm is less accurate but requires only $O(N^4)$ arithmetic operations. In this paper we take a different approach and derive an approximate algorithm that requires only $O(N^2 \log N)$ operations. The algorithm is similar to the approximate algorithm described for square QAM in [1] which is based on searching along 'lines' which pass through hypercubes. For our algorithm the hypercubes are replaced by polytopes that are the Cartesian product of N hexagons.

The block noncoherent detection problem is further complicated by identifiability (or ambiguity) issues. For example, the quadrature-PSK constellation is unchanged by a $\pi/2$ phase rotation. For PSK these identifiability issues are typically handled by differential encoding. Differential encoding can also by used for multi level QAM constellations. However, in this case, more complicated ambiguities can exist that are not resolved by differential encoding. These are known as *divisor ambiguities* and have been completely enumerated for the case of square QAM and some hexagonal constellations different to those in this paper [9, 12]. The divisor ambiguities lead to a lower bound on block error rate for noncoherent detection. The bound decreases quite rapidly as N increases and for sufficiently large N the effect of the lower bound is negligible. For the constellations we consider here the divisor ambiguities

The work of Daniel J. Ryan was supported by the Research Council of Norway (Grants 171133/V30 and 184973/S10).

Robby McKilliam is partly supported by a scholarship from the Wireless Technologies Laboratory, CSIRO ICT Centre, Sydney, Australia

are very simple to describe and there are very few of them. This removes the need for differential encoding and results in the lower bound being smaller than it is in [12].

The paper is organised as follows. In Section II we introduce the particular hexagonal constellations considered in this paper. These are generated from a Voronoi code [13]. In Section III we discuss the signal model for block noncoherent detection and derive an objective function based on least squares. We then describe the divisor ambiguities that exist for these constellations. In Section III-B we derive a fast, approximate, noncoherent detection algorithm. Section IV compares the performance of the algorithm with coherent detection (i.e when perfect channel knowledge is available) and also with the optimal brute force noncoherent detector. We find that the noncoherent detector performs similarly to coherent detection for sufficiently large block length and is almost indistinguishable from the brute force approach, but has significantly smaller computational complexity.

II. Hexagonal constellations and Voronoi codes

We consider hexagonal constellations generated from a *Voronoi code* (1). Such constellations were first described by Conway and Sloane [13] and have the advantage of being simple to encode and decode. We define the hexagonal lattice, A_2, lying in the complex plane, $\mathbb{C}$, as the set of all integer linear combinations of 1 and $\omega = {}^1\!/_2 + j\sqrt{3}/2$ where $j = \sqrt{-1}$. This coincides with the ring of Eisenstein integers.

The *Voronoi region* about a lattice point x is the region that is nearer (in Euclidean distance) to x than to any other lattice point. The Voronoi regions for A_2 are hexagons. We use $\mathcal{H}$ to denote the Voronoi region centered at the origin (Figure 1). It follows that $\mathcal{H} + x$ is the Voronoi region about x. The *relevant vectors* of E are those lattice points directly opposite the edges of $\mathcal{H}$. There are 6 relevant vectors given by $1, -1$, ω, $-\omega$, ω^* and $-\omega^*$ where * denotes the complex conjugate.

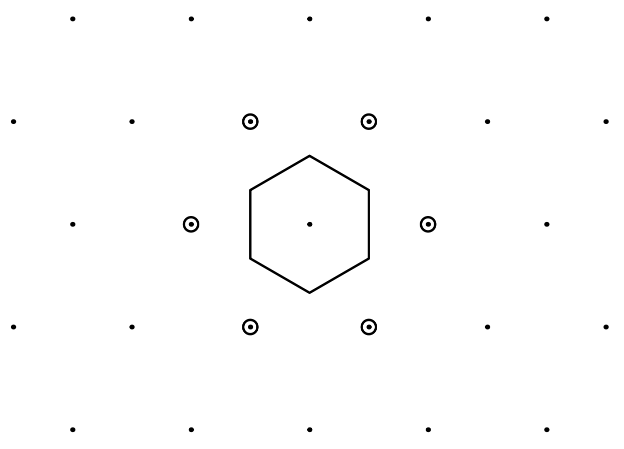

Fig. 1. Hexagonal lattice. The origin is the central lattice point. The Voronoi region $\mathcal{H}$ is marked by the hexagon about the origin. The relevant vectors are marked with circles.

The constellations we consider are given by

$$C(r, t) = (A_2 + t) \cap r\mathcal{H} \tag{1}$$

where $r \in \mathbb{Z}$ is a scale factor and $t \in \mathbb{C}$ is a translation. The number of constellation points is r^2. The translation is chosen to minimise the average energy of the constellation. In this paper we consider 3 constellations $C(4, {}^1\!/_4)$, $C(8, {}^3\!/_8)$ and $C(16, {}^7\!/_{16})$. This gives constellations with 16, 64 and 256 points respectively. We will refer to these as C_{16}, C_{64} and C_{256}. Figures 2 and 3 show C_{16} and C_{64}. The translations are calculated using the iterative scheme described in [13, page 822]. Conway and Sloane conjecture that C_{16} is the optimal 16 point constellation in terms of average energy [13]. Throughout this paper we will use C to refer to either C_{16}, C_{64} or C_{256}.

We can define a Voronoi region about a point, or *symbol*, in a constellation. We use $\mathrm{Vor}(x)$ to denote the Voronoi region about $x \in C$. Notice that for some of the $x \in C$ the Voronoi region is $\mathcal{H} + x$. However, for x lying on the boundary of the constellation $\mathrm{Vor}(x)$ is an open region. It is always the case that $\mathcal{H} + x \subseteq \mathrm{Vor}(x)$. For some $y \in \mathbb{C}$ we define the function $\mathrm{NearestPt}(y)$ to return the nearest (in Euclidean distance) symbol in C to y. It follows that $x = \mathrm{NearestPt}(y)$ if and only if $y \in \mathrm{Vor}(x)$[1].

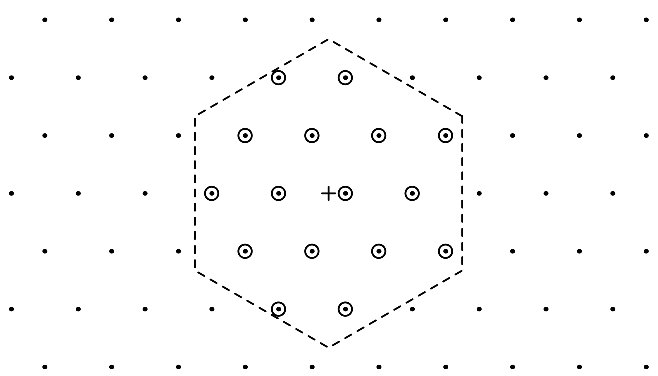

Fig. 2. Translated hexagonal lattice $A_2 - {}^1\!/_4$ (dots) and C_{16} (circles). The region inside the dashed hexagon is $4\mathcal{H}$. The origin is indicated by $+$.

III. Block noncoherent detection

We consider a vector, or *block*, of N symbols taken from a constellation C transmitted in a noisy channel

$$\mathbf{y} = h_0 \mathbf{x}_0 + \mathbf{w}.$$

Here $\mathbf{x}_0 \in C^N$ is the block of transmitted symbols, $\mathbf{y} \in \mathbb{C}^N$ is the received signal and $\mathbf{w} \in \mathbb{C}^N$ is a vector of complex noise. The channel is represented by $h_0 \in \mathbb{C}$ and is assumed to be constant for the duration of the block. The channel is

[1] There is a slight technical deficiency here. We actually require to define some of the faces of $\mathrm{Vor}(x)$ to be closed and some to be open. Ties in $\mathrm{NearestPt}(x)$ can then be broken accordingly.

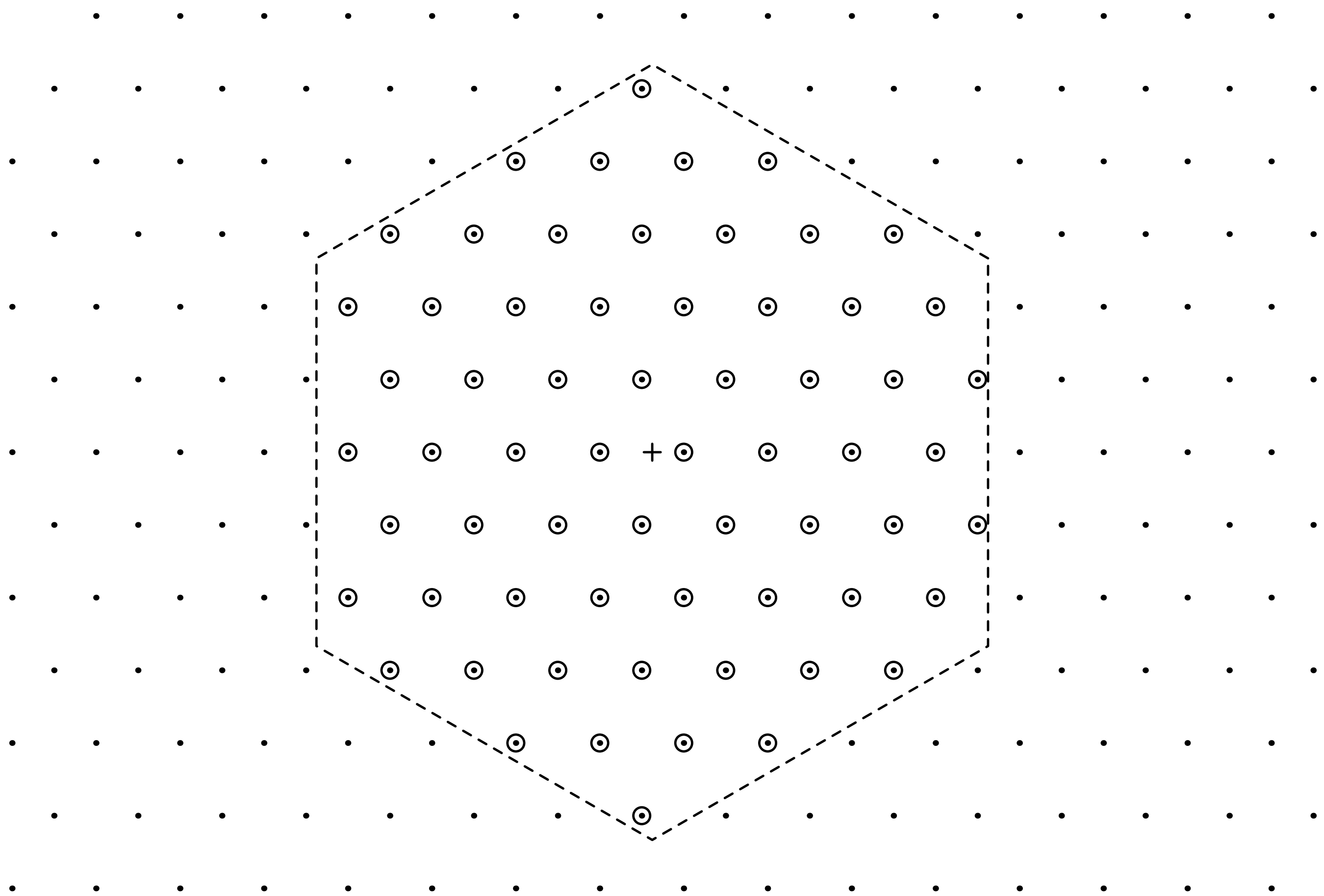

Fig. 3. Translated hexagonal lattice $A_2 - {}^{3}\!/\!{}_{8}$ (dots) and C_{64} (circles). The region inside the dashed hexagon is $8\mathcal{H}$. The origin is indicated by $+$.

unknown at the receiver. Our aim is to estimate $\mathbf{x}_0$ given $\mathbf{y}$. An obvious approach is the least squares estimator

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}\in C^N} \min_{h\in\mathbb{C}} \|\mathbf{y} - h\mathbf{x}\|^2. \tag{2}$$

By conditioning with respect to h it can be shown that [1, 14]

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}\in C^N} \frac{\left|\mathbf{x}^\dagger\mathbf{y}\right|^2}{\|\mathbf{x}\|^2} \tag{3}$$

where † indicates the Hermitian transpose. The objective function (3) is often referred to as the *generalised likelihood ratio test* in the noncoherent detection literature [1, 6, 14].

A. Divisor ambiguities

Inherent ambiguities exist in (3). Consider when $\mathbf{a} = [a, a, a, \dots]$ for some $a \in C$. Let $b = \gamma a$ for some $\gamma \in \mathbb{C}$ and $b \in C$. Then both $\mathbf{a}$ and $\mathbf{b} = \gamma\mathbf{a} = [b, b, b, \dots]$ are in C^N and

$$\frac{\left(\mathbf{a}^\dagger\mathbf{y}\right)^2}{\|\mathbf{a}\|^2} = \frac{\left(\gamma\mathbf{a}^\dagger\mathbf{y}\right)^2}{\|\gamma\mathbf{a}\|^2} = \frac{\left(\mathbf{b}^\dagger\mathbf{y}\right)^2}{\|\mathbf{b}\|^2}.$$

Thus our objective function (3) is unable to differentiate between $\mathbf{a}$ and $\mathbf{b}$. We say that $\mathbf{a}$ and $\mathbf{b}$ are *ambiguous*.

In [1, 12] the classes of ambiguous block were completely described and enumerated for the case of square QAM and some hexagonal constellations different to those here. It was shown that the ambiguities lead to a floor in the block error rate. We find that the problem is easier for our constellations. Intuitively this is due to the translation that is used in Voronoi codes breaking the symmetry of the constellations about the origin. For the remainder of the paper we will use a subscript, e.g. a_n, to indicate the nth symbol in the block $\mathbf{a}$.

Lemma 1. *There are no 4 symbols $a, b, c, d \in C$ with $a \neq b$, $c \neq d$ and $a \neq c$ such that $a/b = c/d$.*

Proof: By brute force enumeration over all $|C|^4$ combinations of a, b, c, d. ■

Lemma 2. *Choose $\mathbf{a}, \mathbf{b} \in C^N$ such that $\mathbf{a} = \gamma\mathbf{b}$ for some $\gamma \in \mathbb{C}\backslash 0$. Then $\mathbf{a} = [a, a, \dots, a]$ and $\mathbf{b} = [b, b, \dots, b]$, i.e. all elements in $\mathbf{a}$ (and $\mathbf{b}$) are the same symbol.*

Proof: Assume this is false. Then there exists an $a_n = \gamma b_n$ and $a_m = \gamma b_m$ such that $a_n \neq a_m$ and $a_n \neq b_n$ and $a_m \neq b_m$. But then $a_n/b_n = a_m/b_m$ contradicting Lemma 1. ■

From Lemma 2 it follows that there are only $|C|$ ambiguous blocks of type $\mathbf{a} = [a, a, \dots, a]$ where $a \in C$. One way to manage the ambiguities is to remove $|C|-1$ of the ambiguous blocks from the set of transmittable blocks. This slightly reduces the average bit rate. However, the remaining (no longer) ambiguous block can be chosen to be $\mathbf{a} = [a, a, \ldots, a]$ where a is the symbol closest to the origin. This slightly reduces the average energy. Another simple approach is to ignore the ambiguities. This results in a floor in the block error rate equal to $(|C|-1)/|C|^N$. If a block error occurs due to ambiguity, then every symbol in the block is received incorrectly. Therefore the same floor applies to the symbol error rate.

B. Algorithm

A brute force approach to computing $\hat{\mathbf{x}}$ is to test each of the $|C|^N$ possible blocks and return that which minimises (3). This is exponentially complex in N. Motivated by techniques suggested in [1, 7, 15] we derive an approximate algorithm that requires only $O(N^2 \log N)$ arithmetic operations. In Section IV we show that the approximate algorithm has statistical performance that is almost indistinguishable from the brute force approach and is substantially less computationally intensive.

Let $h^{-1} = \gamma e^{j\theta}$ where $\theta \in (0, 2\pi]$ and $\gamma \geq 0$. Then the sum of squares function (2) may be written as

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}\in C^N} \min_{\gamma \geq 0} \min_{\theta \in (0,2\pi]} \frac{1}{\gamma^2} \|\gamma e^{j\theta}\mathbf{y} - \mathbf{x}\|^2.$$

Geometrically $\hat{\mathbf{x}}$ can be viewed as the nearest block in angle to the plane of points defined by $\gamma e^{j\theta}\mathbf{y}$ where $\theta \in (0, 2\pi]$ and $\gamma \geq 0$. We can approximate $\hat{\mathbf{x}}$ by sampling θ at a number of values in $(0, 2\pi]$. That is

$$\hat{\mathbf{x}} \approx \arg \min_{\mathbf{x}\in C^N} \min_{\gamma \geq 0} \min_{\theta \in \Theta} \frac{1}{\gamma^2} \|\gamma e^{j\theta}\mathbf{y} - \mathbf{x}\|^2$$

where $\Theta = \{2\pi/L, 4\pi/L, \ldots, 2\pi\}$ and L is the number of samples. Fix $\theta \in \Theta$ and let $\mathbf{z} = e^{j\theta}\mathbf{y}$. Then, conditional on θ,

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}\in C^N} \min_{\gamma \geq 0} \frac{1}{\gamma^2} \|\gamma \mathbf{z} - \mathbf{x}\|^2. \tag{4}$$

Geometrically $\hat{\mathbf{x}}$ is the nearest block in angle to the line of points $\gamma\mathbf{z}$ for $\gamma \geq 0$. We will show how this can be efficiently computed.

For some $\mathbf{y} \in \mathbb{C}^N$ we will write $\mathrm{NearestPt}(\mathbf{y})$ to denote the block $\mathbf{x}$ such that $x_n = \mathrm{NearestPt}(y_n)$ for all $n = 1, 2, \ldots, N$. It follows that $\mathrm{NearestPt}(\mathbf{y})$ is the nearest block in Euclidean distance to $\mathbf{y}$. We will also use $\mathrm{Vor}(\mathbf{x})$ to denote the region of $\mathbb{C}^N$ that is nearer in Euclidean distance to $\mathbf{x}$ that any other block. It is easy to show that

$$\mathrm{Vor}(\mathbf{x}) = \mathrm{Vor}(x_1) \times \mathrm{Vor}(x_2) \times \cdots \times \mathrm{Vor}(x_N)$$

where $\times$ denotes the Cartesian product. It follows that $\mathrm{Vor}(\mathbf{x}) \supseteq \mathcal{H}^N + \mathbf{x}$. Let

$$B = \left\{\mathbf{x} \in C^N \mid \mathbf{x} = \mathrm{NearestPt}(\gamma\mathbf{z}), \gamma \geq 0\right\}.$$

Then (4) is equivalent to

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}\in B} \min_{\gamma \geq 0} \frac{1}{\gamma^2} \|\gamma \mathbf{z} - \mathbf{x}\|^2. \tag{5}$$

We can equivalently write B as

$$B = \left\{\mathbf{x} \in C^N \mid \gamma\mathbf{z} \in \mathrm{Vor}(\mathbf{x}), \gamma \geq 0\right\}.$$

Unfortunately B itself is difficult to compute due to some of the $\mathrm{Vor}(\mathbf{x})$ being complicated, open regions. Instead we use

$$\tilde{B} = \left\{\mathbf{x} \in C^N \mid \gamma\mathbf{z} \in \mathcal{H}^N + \mathbf{x}, \gamma \geq 0\right\} \subseteq B$$

and compute

$$\hat{\mathbf{x}} \approx \arg \min_{\mathbf{x}\in \tilde{B}} \min_{\gamma \geq 0} \frac{1}{\gamma^2} \|\gamma \mathbf{z} - \mathbf{x}\|^2.$$

This proves to be a reasonable approximation. Approximations of this type have been called *lattice decoding* in the literature [16]. We will show how the elements in $\tilde{B}$ can be found efficiently.

The first block in $\tilde{B}$ is given when $\gamma = 0$ and is denoted $\mathbf{x}^{[1]} = \mathrm{NearestPt}(\mathbf{0})$ where $\mathbf{0}$ is a vector of zeros. To compute the second block we observe where the line of points $\gamma\mathbf{z}$ intersects the boundary of $\mathcal{H}^N + \mathbf{x}^{[1]}$. This will occur when γz_n intersects $\mathcal{H} + x_n$ for some $n \in \{1, 2, \ldots, N\}$. The line then enters $\mathcal{H}^N + \mathbf{x}^{[1]} + g\mathbf{e}_n$ where g is a relevant vector of the hexagonal lattice and $\mathbf{e}_n$ is a vector of 0's except for a 1 in the nth position. The second block is then $\mathbf{x}^{[2]} = \mathbf{x}^{[1]} + g\mathbf{e}_n$. Similarly, the third block in $\tilde{B}$ can be found by observing where the line first intersects $\mathcal{H}^N + \mathbf{x}^{[2]}$. In this manner all the blocks in $\tilde{B}$ can be found. Figure 4 depicts this process in 1 dimension.

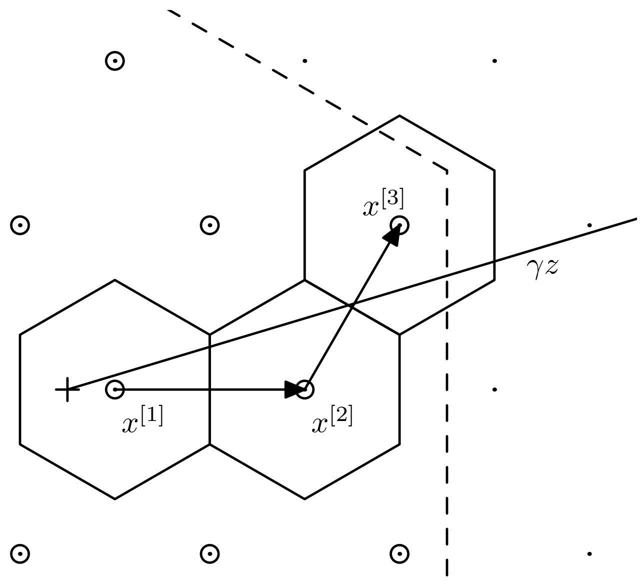

Fig. 4. Figure showing the line γz intersecting the Voronoi regions in the hexagonal lattice. The arrows show the relevant vectors that are added as γ increases.

We can also efficiently compute the objective function (3) for each element in $\tilde{B}$. Let $\mathbf{x}^{[t]} \in \tilde{B}$ and let $\alpha_t = \mathbf{y}^\dagger\mathbf{x}^{[t]}$ and $\beta_t = \|\mathbf{x}^{[t]}\|^2$ and

$$L_t = \frac{\left|\mathbf{y}^\dagger\mathbf{x}^{[t]}\right|^2}{\|\mathbf{x}^{[t]}\|^2} = \frac{|\alpha_t|^2}{\beta_t}$$

be the objective function evaluated at $\mathbf{x}^{[t]}$. Then $\mathbf{x}^{[t+1]} = \mathbf{x}^{[t]} + g\mathbf{e}_n$ and α_{t+1}, β_{t+1} and L_{t+1} can be computed recursively by

$$\alpha_{t+1} = \mathbf{y}^\dagger(\mathbf{x}^{[t]} + g\mathbf{e}_n) = \alpha_t + g y_n^* \tag{6}$$

$$\beta_{t+1} = \|\mathbf{x}^{[t]} + g\mathbf{e}_n\|^2 = \beta_t + 2\Re\left(g^* x_n^{[t]}\right) + 1 \tag{7}$$

where $\Re(\cdot)$ denotes the real part of it's argument.

Algorithm 1 now follows. Lines 8-15 compute a set S of 3-tuples $\{\gamma, n, g\}$ that are related to $\tilde{B}$. After S is sorted in ascending order of γ (Line 15) the $t+1$th block of $\tilde{B}$ can be computed as $\mathbf{x}^{[t+1]} = \mathbf{x}^{[t]} + g\mathbf{e}_n$ where $\{\gamma, n, g\} = S_t$ where S_t denotes the tth element of S. The function

$$\{\gamma, g\} = \mathrm{nextHexagonalBoundary}(z_n, v)$$

computes the greatest value of γ where the line γz_n intersects the boundary of $\mathcal{H} + v$. It also returns the corresponding relevant vector g. This is achieved by testing each of the 6 edges of $\mathcal{H} + v$ (see Figure 4).

Lines 16-2 compute the objective function for each block in $\tilde{B}$ using the recursive operations (6) and (7). The value of γ and θ that correspond to the best block is stored on line 22 so that the block can be recovered on line 23 using $\mathrm{NearestPt}(\cdot)$. $\mathrm{NearestPt}(\cdot)$ can be computed efficiently using the algorithm in [17]. On line 22, notice that $\hat{\gamma} = \gamma + \epsilon$. The ϵ is added to avoid rounding problems due to $\gamma\mathbf{z}$ being on the boundary of the Voronoi region. In practice ϵ can be set so that $\hat{\gamma}$ is halfway between consecutive boundary crossings.

Lines 3 to 6 require $O(N)$ arithmetic operations or less. It can be shown that the loops starting on lines 9 and 16 require $O(|S|)$ arithmetic operations. The sort operation on line 15 requires $O(|S| \log |S|)$ operations. As $|S| \geq N$ then the computational complexity inside the main loop on line 2 is $O(|S| \log |S|)$. As the main loop iterates L times the algorithm requires $O(L|S| \log |S|)$ operations. In practice we find that $L = rN$ gives good results. It can be shown that $|S| \leq 2rN$ by observing that A_2 is the union of two translated rectangular lattices. The algorithm then requires $O(r^2 N^2 \log(rN))$ operations.

Input: $\mathbf{y} \in \mathbb{C}^N$

1 $L = -\infty$
2 **for** $\theta \in \{2\pi/L, 4\pi/L, \ldots, 2\pi\}$ **do**
3     $\mathbf{z} = e^{j\theta}\mathbf{y}$
4     $\mathbf{x} = \mathrm{NearestPt}(\mathbf{0})$
5     $\alpha = \mathbf{y}^\dagger\mathbf{x}$
6     $\beta = \|\mathbf{x}\|^2$
7   **if** $L < |\alpha|^2/\beta$ **then** $L = |\alpha|^2/\beta$, $\hat{\gamma} = 0$, $\hat{\theta} = \theta$
8     $S = \emptyset$
9   **for** $n \in \{1, 2, \ldots, N\}$ **do**
10         $v = x_n$
11   **while** $v \in r\mathcal{H}$ **do**
12             $\{\gamma, g\} = \mathrm{nextHexagonalBoundary}(z_n, v)$
13   add $\{\gamma, n, g\}$ to S
14             $v = v + g$
15   sort S in ascending order of γ
16   **for** $t \in \{1, 2, \ldots, |S|\}$ **do**
17         $\{\gamma, n, g\} = S_t$
18         $\alpha = \alpha + y_n^* g$
19         $\beta = \beta + 2\Re\{x_n^* g\} + 1$
20         $x_n = x_n + g$
21   **if** $x_n \notin r\mathcal{H}$ **then** break for loop
22   **if** $L < |\alpha|^2/\beta$ **then** $L = |\alpha|^2/\beta$, $\hat{\theta} = \theta$, $\hat{\gamma} = \gamma + \epsilon$
23 $\hat{\mathbf{x}} = \mathrm{NearestPt}(\hat{\gamma}e^{j\hat{\theta}}\mathbf{y})$
24 **return** $\hat{\mathbf{x}}$

Algorithm 1: Algorithm to find the nearest point

IV. Simulations

In this section we present Monte Carlo simulations displaying the performance of the noncoherent detector. For the simulations the noise, $\mathbf{w}$, is independent and identically distributed complex Gaussian noise with real and imaginary parts having variance σ^2. The noise power is $N_0 = 2\sigma^2$. The channel, h_0, is assumed to be complex Gaussian with real and imaginary parts having variance 1.

Figure 5 shows the symbol error rate (SER) of the detector with the C_{16} constellation for varying E_b/N_0 where E_b is the energy per bit. Unsurprisingly, the performance of the detector improves as N increases. The performance is close to that of coherent hexagonal QAM when $N = 100$. The approximate noncoherent detector performs almost identically to the optimal brute force detector when $N = 3$. It is computationally infeasible to run the brute force detector for $N > 3$. In these simulations we have not removed ambiguous blocks. The SER therefore reaches a floor at $(|C|-1)/|C|^N$. When $N = 3$ this floor is ≈ 0.003662. In the plot it appears that the curve for $N = 3$ reaches a floor at ≈ 0.015. It is actually the case that the curve is still deceasing and reaches the true error floor when $E_b/N_0 \approx 180\,\mathrm{dB}$. Figure 6 shows the SER for the C_{64} constellation. The results are similar to the C_{16} constellation.

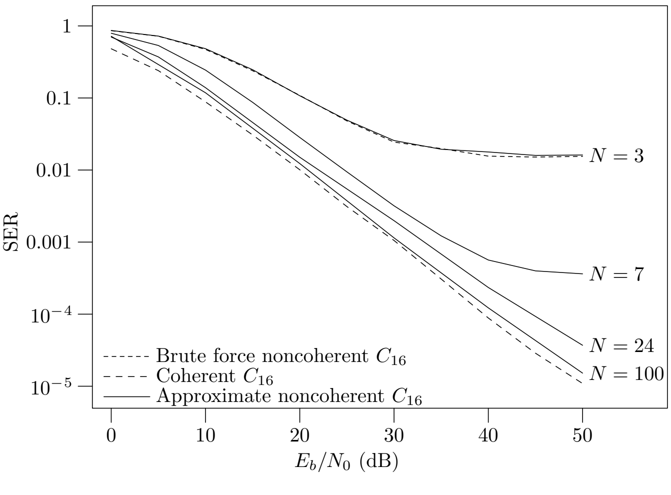

Fig. 5. Symbol error rate versus E_b/N_0 for C_{16}.

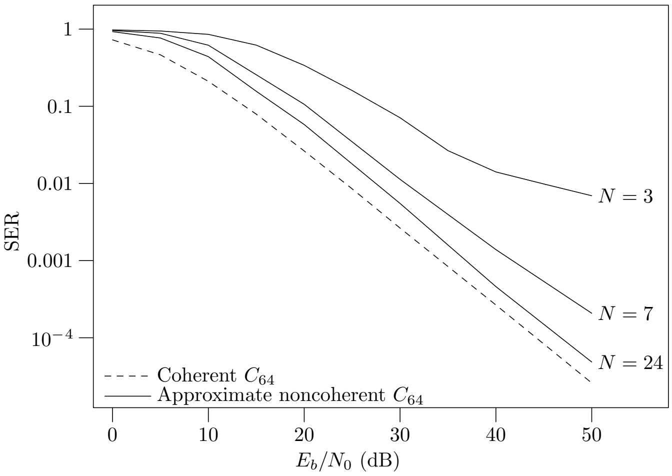

Fig. 6. Symbol error rate versus E_b/N_0 for C_{64}.

V. Discussion and Conclusion

The C_{16}, C_{64} and C_{256} constellations and the fast noncoherent detection algorithm have some interesting properties that we shall discuss before concluding. Firstly, it is worth noting that the fast detection algorithm can easily be applied to other hexagonal constellations such as those in [8, 9, 12]. It is also interesting that the ambiguity problem is less severe for C_{16}, C_{64} and C_{256} than for square QAM, M-PSK ($M \geq 4$) and other hexagonal constellations which lose at least 2 bits per block due to differential encoding. Differential encoding is not necessary for C_{16}, C_{64} and C_{256}. Intuitively this is due to the translation that is used in Voronoi codes breaking the symmetry of the constellations about the origin. There is no reason that a similar translation could not be applied to, say, square QAM, or even PSK to obtain a similar effect. This would, however, increase the average energy of these constellations, whereas for a Voronoi code, the translation is made in an attempt to minimise the average energy. In this paper we have not discussed coded transmission of the constellations, however, it is possible to combine the techniques here with other coding schemes as is done in [2, 6].

References

[1] D. J. Ryan, I. V. L. Clarkson, and I. B. Collings, "GLRT-optimal noncoherent lattice decoding," *IEEE Trans. Signal Process.*, vol. 55, pp. 3773–3786, July 2007.

[2] R. Chen, R. Koetter, D. Agrawal, and U. Madhow, "Joint demodulation and decoding for the noncoherent block fading channel: a practical framework for approaching Shannon capacity," *IEEE Trans. Commun.*, vol. 51, no. 10, pp. 1676–1689, Oct. 2003.

[3] K.M. Mackenthun, "A fast algorithm for multiple-symbol differential detection of MPSK," *IEEE Trans. Commun.*, vol. 42, pp. 1471–1474, Feb/Mar/Apr 1994.

[4] W. Sweldens, "Fast block noncoherent decoding," *IEEE Comms. Letters*, vol. 5, no. 4, pp. 132–134, Apr. 2001.

[5] R. G. McKilliam, I. V. L. Clarkson, D. J. Ryan, and I. B. Collings, "Linear-time block noncoherent detection of PSK," *Proc. Internat. Conf. Acoust. Speech Signal Process.*, pp. 2465–2468, Apr. 2009.

[6] I. Motedayen-Aval and A. Anastasopoulos, "Polynomial-complexity noncoherent symbol-by-symbol detection with application to adaptive iterative decoding of turbo-like codes," *IEEE Trans. Commun.*, vol. 51, no. 2, pp. 197–207, Feb 2003.

[7] R. G. McKilliam, D. J. Ryan, I. V. L. Clarkson, and I. B. Collings, "An improved algorithm for optimal noncoherent QAM detection," *Proc. Australian Commun. Theory Workshop*, pp. 64–68, Jan. 2008.

[8] D. J. Ryan, *Lattice theory and algorithms for wireless communications*, PhD disseration, April 2008.

[9] D.J. Ryan, I.V.L. Clarkson, and I.B. Collings, "Blind detection of hexagonal QAM in fading channels," *Proc. Internat. Sympos. Signal Process. Appl.*, vol. 1, pp. 279–282, August 2005.

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

[11] L. Babai, "On Lovasz lattice reduction and the nearest lattice point problem," *Combinatorica*, vol. 6, pp. 1–13, 1986.

[12] D. J. Ryan, I. V. L. Clarkson, and I. B. Collings, "Blind detection of PAM and QAM in fading channels," *IEEE Trans. Inform. Theory*, vol. 52, no. 3, pp. 1197–1206, March 2006.

[13] J. Conway and N. Sloane, "A fast encoding method for lattice codes and quantizers," *IEEE Trans. Inform. Theory*, vol. 29, no. 6, pp. 820–824, Nov 1983.

[14] D. Warrier and U. Madhow, "Spectrally efficient noncoherent communication," *IEEE Trans. Inform. Theory*, vol. 48, no. 3, pp. 651–668, Mar. 2002.

[15] D. J. Ryan, I. V. L. Clarkson, and I. B. Collings, "Maximum-likelihood noncoherent lattice decoding of QAM," *Proc. Internat. Conf. Acoust. Speech Signal Process.*, vol. 4, pp. 189–192, 2006.

[16] U. Erez and R. Zamir, "Achieving $1/2 \log(1 + SNR)$ on the AWGN channel with lattice encoding and decoding," *IEEE Trans. Inform. Theory*, vol. 50, no. 10, pp. 2293–2314, Oct. 2004.

[17] W. H. Mow, "Fast decoding of the hexagonal lattice with applications to power efficient multi-level modulation systems," *Singapore ICCS/ISITA*, pp. 370–373, Nov 1992.