Constructions of Quasi-Cyclic Measurement Matrices Based on Array Codes

Xin-Ji Liu *†, Shu-Tao Xia*†
*Graduate School at Shenzhen, Tsinghua University, Shenzhen, Guangdong, China
†Tsinghua National Laboratory for Information Science and Technology, Beijing, China
Email: liuxj11@mails.tsinghua.edu.cn, xiast@sz.tsinghua.edu.cn

***Abstract*—Recently, Dimakis, Smarandache, and Vontobel indicated that the parity-check matrices of good LDPC codes can be used as *provably* good measurement matrices for *compressed sensing* (CS) under *basis pursuit* (BP). In this paper, we consider the parity-check matrix $H(r,q)$ of the *array codes*, one of the most important kind of structured LDPC codes. The *spark*, i.e. the smallest number of linearly dependent columns in a matrix, of $H(2,q)$ and $H(3,q)$ are determined and two lower bounds of the sparks of $H(r,q)$ are given for $r \geq 4$. Moreover, we carry out numbers of simulations and show that many parity-check matrices of array codes and their submatrices perform better than the corresponding Gaussian random matrices. The proposed measurement matrices have perfect quasi-cyclic structures and can make the hardware realization convenient and easy, thus having great potentials in practice.**

I. Introduction

For a k*-sparse* signal $\boldsymbol{x} \in \mathbb{R}^n$ with at most k nonzero components, *compressed sensing* (CS) [1], [2] aims to recovering it from its linear measurements $\boldsymbol{y} = H\boldsymbol{x}$, where $H \in \mathbb{R}^{m\times n}$ is the *measurement matrix* with $m \ll n$. This can be done by solving the following l_0*-optimization* problem

$$\min ||\boldsymbol{x}||_0 \quad s.t. \quad H\boldsymbol{x} = \boldsymbol{y}. \tag{1}$$

Unfortunately, it is well-known that the problem (1) is NP-hard in general. The theory of CS indicates that by choosing appropriate measurement matrix H, the output $\hat{\boldsymbol{x}}$ of a convex relaxation of (1), the l_1*-optimization* (a.k.a. *basis pursuit*, BP),

$$\min ||\boldsymbol{x}||_1 \quad s.t. \quad H\boldsymbol{x} = \boldsymbol{y}, \tag{2}$$

is coincident with that of (1). Besides, some greedy algorithms for l_0-optimization, such as the *orthogonal matching pursuit* (OMP) [3] algorithm and its modifications, can also produce exact estimate $\hat{\boldsymbol{x}}$ of $\boldsymbol{x}$.

The construction of the measurement matrix H is one of the main concerns in CS and some criteria of selecting effective measurement matrices have been proposed. In their earlier and fundamental work [4], Donoho and Elad defined the *spark* of a matrix H as,

$$spark(H) \triangleq \min\{||\boldsymbol{w}||_0 : \boldsymbol{w} \in Nullsp^*_{\mathbb{R}}(H)\}, \tag{3}$$

where

$$Nullsp^*_{\mathbb{R}}(H) \triangleq \{\boldsymbol{w} \in \mathbb{R}^n : H\boldsymbol{w} = \boldsymbol{0}, \boldsymbol{w} \neq \boldsymbol{0}\}. \tag{4}$$

It is easy to know [5] that any k-sparse signal $\boldsymbol{x}$ can be exactly recovered by the l_0-optimization (1) if and only if

$$spark(H) > 2k. \tag{5}$$

Other useful criteria include the well-known *restricted isometry property* (RIP) [6] and the *nullspace property* (NSP) [7]. In this paper, we will use spark to help analyze the constructed measurement matrices as it is simpler to deal with.

Generally, constructing methods of measurement matrices involve random construction and deterministic construction. Many random matrices, such as Fourier matrices and Gaussian matrices, satisfy RIP with overwhelming probability. However, there is no guarantee that a specific realization of random matrices works and it costs large storage space. Deterministic measurement matrices can avoid these shortcomings. Among the deterministic constructions, binary 0-1 matrices from coding theory attract many attentions [8], [9]. Recently, Dimakis, Smarandache, and Vontobel [10] showed that the parity-check matrices of good LDPC codes can be used as *provably* good measurement matrices under basis pursuit. In addition, Lu, Kpalma and Ronisn [11] conducted a series of experiments and verified that measurement matrices constructed by PEG algorithm, which is the best known algorithm to generate parity-check matrices of good random LDPC codes, significantly outperform the corresponding Gaussian random matrices under OMP algoritm.

The main focus of this paper is to utilize the parity-check matrices of *array codes* [12], [13] and their submatrices to construct binary measurement matrices. An array code $\mathcal{C}(r,q)$ is a LDPC code defined by a parity-check matrix $H(r,q)$ which consists of $r \times q$ *circulant permutation* submatrices, where q is an odd prime, $1 \leq r \leq q$. For some small r and q, the exact values or upper and lower bounds of the *minimum distance* of $\mathcal{C}(r,q)$ have been determined [14], [16], [17]. For example, it has been proved that in [14] that $d(\mathcal{C}(2,q)) = 4$ and $d(\mathcal{C}(3,q)) = 6$. As for *girth*, [15, Th. 2.1] and [13, Th. 1] showed that $H(r,q)$ with $r \geq 3$ have girth 6. Besides, by removing certain specific columns of $H(r,q)$, we can avoid some short circles and increase the girth [13] and we call the corresponding codes as *shortened array codes of large girth*.

In this paper, we firstly consider the spark of the parity-check matrix of an (shortened) array code. With the help of the determination method of $d(C(2,q))$ and $d(C(3,q))$ in [14], we prove that $spark(H(2,q)) = 4$ and $spark(H(3,q)) = 6$. For $r \geq 4$, two lower bounds of the spark of the parity-check matrix H of an (shortened) array code are derived, one depends only on the minimum distance of the array code and the other depends on r and the girth g of H. Apart from the theoretical results, lots of experiments are carried out afterwards. Simulation results show that in many cases, the parity-check matrices of array codes and their submatrices perform better than Gaussian random matrices. In particular, when n is large, the measurement matrix $H \in \{0,1\}^{m\times n}$ based on array codes have good performance for m varying within a very wide range, which is very useful in practice.

II. Array Codes

Array code is an important class of structured LDPC codes introduced in [12] which has good performance. Let q be an odd prime, $\mathcal{I}$ be the $q\times q$ identity matrix and $P\neq\mathcal{I}$ be a $q\times q$ circulant permutation matrix. A *permutation matrix* is a square matrix with only entries 0's and 1's. In addition, each row and column have only a single 1. A *circulant matrix* means the ith row of this matrix is the cyclic shift of the (i-1)th or (i+1)th row by one position. For example, P is typically chosen to be

$$P=\begin{pmatrix} 0 & 1 & 0 & \cdots & 0\\ 0 & 0 & 1 & \cdots & 0\\ \vdots & \vdots & \vdots & \vdots & \vdots\\ 0 & 0 & 0 & \cdots & 1\\ 1 & 0 & 0 & \cdots & 0 \end{pmatrix}. \tag{6}$$

Suppose r is an integer in $[1,q]$, $\{a_0,a_1,\ldots,a_{r-1}\}$ is a sequence of r distinct integers in [0, q-1]. A *general* form of a parity-check matrix $H(r,q)$ which defines the *array code* $\mathcal{C}(r,q)$ was given in [13] as:

$$H(r,q)=\begin{pmatrix} P^{a_0\cdot 0} & P^{a_0\cdot 1} & \ldots & P^{a_0\cdot(q-1)}\\ P^{a_1\cdot 0} & P^{a_1\cdot 1} & \ldots & P^{a_1\cdot(q-1)}\\ \vdots & \vdots & \vdots & \vdots\\ P^{a_{r-1}\cdot 0} & P^{a_{r-1}\cdot 1} & \ldots & P^{a_{r-1}\cdot(q-1)} \end{pmatrix}. \tag{7}$$

$H(r,q)$ can be seen as a $r\times q$ "matrix" whose (i,j)th[1] "entry" is the $q\times q$ circulant permutation matrix $P^{a_{i-1}\cdot(j-1)}$. Each "row" ("column") of this "matrix" will be called a *block-row* (*block-column*) of $H(r,q)$. Obviously, $H(r,q)$ is (r,q)-*regular* with uniform column weight r and uniform row weight q.

If $\{a_0,a_1,\ldots,a_{r-1}\}$ forms an arithmetic progression, we call the corresponding array code a *proper array code* (PAC) and otherwise an *improper array code* (IAC). If $a_i=i$, then $H(r,q)$ will simply be

$$H_P(r,q)=\begin{pmatrix} \mathcal{I} & \mathcal{I} & \cdots & \mathcal{I}\\ \mathcal{I} & P & \cdots & P^{q-1}\\ \vdots & \vdots & \vdots & \vdots\\ \mathcal{I} & P^{r-1} & \ldots & P^{(r-1)\cdot(q-1)} \end{pmatrix}. \tag{8}$$

The array code $\mathcal{C}_P(r,q)$ defined by (8) is actually the original version presented in [12]. If $a_0=0$ and $a_i=2^{i-1}(\mod q)$ for $i\geq 1$, denote the corresponding parity-check matrix as $H_I(r,q)$ and the array code as $\mathcal{C}_I(r,q)$. It is easy to see that when $r=1,2,3$, $H_P(r,q)$ and $H_I(r,q)$ are equivalent. In this paper, we will use $\mathcal{C}_P(r,q)$ and $\mathcal{C}_I(r,q)$ defined above as the representatives of PAC and IAC, respectively. In addition, the term *array code* without any other modifiers in the rest may be used to mean a PAC or an IAC.

While constructing LDPC codes, parity-check matrices of large girth are preferable as they perform better under message passing decoding. In [13], Milenkovic, Kashyap and Leyba observed that for any array code with parity-check matrix H, the cycles in G_H are governed by certain homogeneous linear equations with integer coefficients. Consequently, by only retaining those columns indexed by integer sequences that do not contain solutions to the equations governing those cycles, one can avoid some short circles and obtain a shortened array code of large girth. It was shown in [13] that a shortened PAC with at least 3 block-columns has girth at most 8, while a shortened IAC of column weight at least 3 can have girth as large as 12.

[1]Throughout this paper, row and column indices of a matrix start from 1.

III. Measurement Matrices Based on Array Codes

In this section, we will use the parity-check matrices of array codes and their submatrices to construct a kind of binary and highly-structured measurement matrices for CS. Since any k-sparse signal $\boldsymbol{x}$ measured by H can be exactly recovered by l_0-optimization (1) if and only if $k<spark(H)/2$, spark is an important performance parameter of the measurement matrix. Therefore, firstly we make an analysis of the spark of array codes' parity-check matrices.

A. Analysis of Spark of Array Codes' Parity-check Matrices

Traditionally, a easily computable property, *coherence*, of a matrix is used to bound its spark [4]. For a matrix $H=(\boldsymbol{h}_1,\boldsymbol{h}_2,\cdots,\boldsymbol{h}_n)\in\mathbb{R}^{m\times n}$, its coherence $\mu(H)$ is defined as:

$$\mu(H)\triangleq\max_{1\leq i\neq j\leq n}\frac{|\langle\boldsymbol{h}_i,\boldsymbol{h}_j\rangle|}{||\boldsymbol{h}_i||_2||\boldsymbol{h}_j||_2}, \tag{9}$$

where $\langle\boldsymbol{h}_i,\boldsymbol{h}_j\rangle\triangleq\boldsymbol{h}_i^T\boldsymbol{h}_j$ denotes the inner product of vectors. Given the coherence of a matrix, it is shown in [4] that

$$spark(H)\geq 1+\frac{1}{\mu(H)}. \tag{10}$$

Consider a binary parity-check matrix H with uniform column weight r and girth $g\geq 6$, such as $H(r,q)$, the maximum inner product of any two distinct columns of H is $\lambda=1$. By (9), we have $\mu(H)=\frac{1}{r}$. Thus the lower bound (10) implies $spark(H)\geq 1+r$ and so for $H(r,q)$, we have

$$spark(H(r,q))\geq 1+r. \tag{11}$$

Actually, for a binary matrix with girth $g\geq 6$ and uniform column weight r, its spark can be further lower bounded.

Theorem 1: For a binary matrix H with girth $g\geq 6$ and uniform column weight r, its spark satisfies

$$spark(H)\geq 2\sum_{u=0}^{t+1}(r-1)^u \text{ with } t\triangleq\lfloor\frac{g-6}{4}\rfloor. \tag{12}$$

Proof: It has been proved in [18, Th. 3] that for any $\boldsymbol{w}\in Nullsp_{\mathbb{R}}^*(H)$ and any $j\in supp(\boldsymbol{w})\triangleq\{i:w_i\neq 0\}$, $|w_j|\leq\frac{||\boldsymbol{w}||_1}{C_0}$, where $C_0=2\sum_{u=0}^{t+1}(r-1)^u$. Obviously, it can be implied that $|supp(\boldsymbol{w})|\geq C_0$ and (12) follows directly. ■

Remark 1: The lower bound in (12) is tight, see Example 2–5 in [18].

Remark 2: The lower bound in (12) is better than that in (11) for all H with $g\geq 6$. For example, for H with $g=6$, (12) implies $spark(H)\geq 2r$ while (11) only implies $\geq 1+r$.

Suppose the minimum distance of the binary code $\mathcal{C}$ defined by parity-check matrix H is $d(\mathcal{C})$. The following theorem says that $spark(H)$ could also be lower bounded by $d(\mathcal{C})$.

Theorem 2: Let H be a binary matrix and $d(\mathcal{C})$ be the minimum distance of the binary code $\mathcal{C}$ defined by H. Then,

$$spark(H)\geq d(\mathcal{C}). \tag{13}$$

Proof: For a matrix $H=(\boldsymbol{h}_1,\boldsymbol{h}_2,\ldots,\boldsymbol{h}_n)\in\{0,1\}^{m\times n}$, there must exist a basis with elements all in $\mathbb{Q}$ of the null space of H since the basis of the null space of H can be computed by Gaussian elimination and the elements of H are either 0 or 1. Therefore, if $\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_n$ are linear dependent in $\mathbb{R}^n$, there must exist not all zero $c_1, c_2, \ldots, c_n$ with $c_i \in \mathbb{Q}$ for $1 \leq i \leq n$ such that $\sum_{i=1}^{n} c_i \boldsymbol{h}_i = \mathbf{0}$. By reduction of fractions to a common denominator, there exist not all zero $c_1', c_2', \ldots, c_n'$ with $c_i' \in \mathbb{Z}$ for $1 \leq i \leq n$ and such that $\sum_{i=1}^{n} c_i' \boldsymbol{h}_i = \mathbf{0}$. Besides, there must be a $1 \leq j \leq n$ such that c_j' is an odd number (if not, divide all c_i' by 2 simultaneously until there is an odd c_j'). By executing modulo 2 operation, there exist not all zero $c_1'', c_2'', \ldots, c_n''$ with $c_i'' \in \{0, 1\}$ for $1 \leq i \leq n$ such that $\sum_{i=1}^{n} c_i'' \boldsymbol{h}_i = \mathbf{0} \pmod 2$, i.e. $\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_n$ are linear dependent in $\{0, 1\}^n$. Consequently, (13) follows directly. ■

Remark 3: Let H be the $m \times (2^m - 1)$ parity-check matrix of a binary $[2^m - 1, 2^m - m - 1, 3]$ Hamming code $\mathcal{C}$. It is easy to see that $spark(H) = d(\mathcal{C}) = 3$, implying that the lower bound in (13) is tight.

Remark 4: Let H be the matrix defined in Example 2 in [18] and $\mathcal{C}$ be the code defined by H. Obviously, $spark(H) = 4 > d(\mathcal{C}) = 3$, so the lower bound in (13) is nontrivial.

Remark 5: For some small r and q, [17, Table I] lists the values of the minimum distances of some PACs defined by (8). Combine these values with (13), one can find that (13) seems better than (12). However, (12) is meaningful as the exact values or even bounds of the minimum distances of (shortened) array codes are difficult to obtain for large r and q and (12) can give us some insights into their spark.

For $r = 2$ and 3, PACs and IACs defined in Section II are equivalent and their $spark(H(r, q))$ can be obtained.

Theorem 3: Let q be an odd prime, for the parity-check matrix $H(r, q)$ defined in (8), we have

$$\begin{aligned} spark(H(2,q)) &= 4, \\ spark(H(3,q)) &= 6. \end{aligned}$$

Proof: From Theorem 1 or 2, $spark(H(2, q)) \geq 4$ and $spark(H(3, q)) \geq 6$. Therefore, we only need to give a $\boldsymbol{w}_1 \in Nullsp^*_{\mathbb{R}}(H(2, q))$ and a $\boldsymbol{w}_2 \in Nullsp^*_{\mathbb{R}}(H(3, q))$ such that $||\boldsymbol{w}_1||_0 = 4$ and $||\boldsymbol{w}_2||_0 = 6$. Similar to the search of a stopping set of size 4 from $H(2, q)$ and a stopping set of size 6 from $H(3, q)$ in [14], we can easily obtain the satisfied $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$. For example, set the first and second components of $\boldsymbol{w}_1$ to -1, the $(q + 2)$th and $(q^2 - q + 1)$th components to 1, and other components to 0, then $H(2, q)\boldsymbol{w}_1 = \mathbf{0}$. As for $\boldsymbol{w}_2$, set the first, second and $\frac{q^2+3q+6}{2}$th components to 1, the $(q+2)$th, $(q+3)$th and $\frac{q^2-q+2}{2}$th components to -1, then $H(3, q)\boldsymbol{w}_2 = \mathbf{0}$. ■

B. Measurement Matrices Directly Based on Array Codes

The parity-check matrix $H(r, q)$ of PACs or IACs can be directly used as $rq \times q^2$ measurement matrices for CS. In this part, we will give some simulation results on their performances. We will see that these measurement matrices are good in practice and their performance agrees with or even outperform that indicates by the above theoretical results.

All the simulations are performed under the same conditions as with [8], [9]. The upcoming figures show the percentages of perfect recovery ($\text{SNR}_{rec.} \geq 100$dB) when different sparsity orders are considered. For the generation of the k-sparse input signals, we first select the support uniformly at random and then generate the corresponding values independently by the standard normal distribution $\mathcal{N}(0, 1)$. The OMP algorithm is used to reconstruct the k-sparse input signals from the compressed measurements and the results are averaged over 5000 runs for each sparsity k. The percentages of perfect recovery of both the proposed matrix and Gaussian random matrix with the same sizes are shown in figures for comparisons. For the Gaussian random matrix each entry is also chosen *i.i.d.* from $\mathcal{N}(0, 1)$. The base submatrix P in $H(r, q)$ is chosen to be (6).

For any binary measurement matrix H with uniform column weight r and girth $g \geq 6$, OMP algorithm is supposed to exactly recover any k_1-sparse signal with $k_1 < (1 + r)/2$ [3, Corollary 3.6]. However, according to Theorem 1, l_0-optimization (1) is supposed to recover any k_2-sparse signal with $k_2 < r$ for $g = 6$ or 8 and even larger k_2 for larger g. The upper bound of k_2 is roughly at least twice of that of k_1, and we expect OMP algorithm, a greedy algorithm for l_0-optimization, can exactly recover signals with sparsity order $k < r$ instead of $k < (1+r)/2$ for $g = 6$ or $g = 8$ in practice, see the following examples.

Example 1: Let $r = 4$, $spark(H(4, q)) \geq 2r = 8$ for any odd prime q. We expect OMP algorithm can recover k-sparse signals with $k < r = 4$ exactly. Fig. 1 shows the perfect recovery percentages of the parity-check matrices of PAC and IAC with $r = 4$, $q = 17$, $q = 19$, $q = 23$ and their corresponding Gaussian random matrices under OMP.

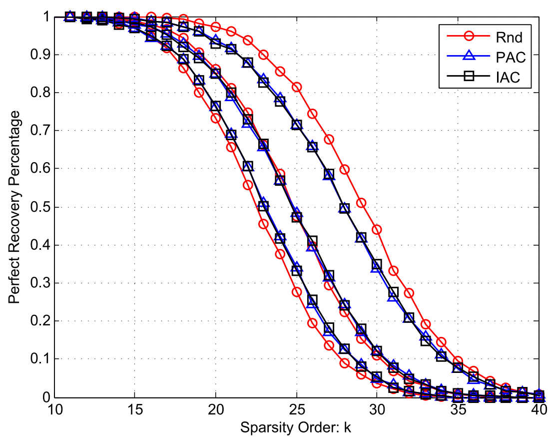

Fig. 1. Perfect recovery percentages of Gaussian random matrices of sizes 68×289, 76×361, 92×529 (red ○ from left to right), $H_P(4, 17)$, $H_P(4, 19)$, $H_P(4, 23)$ (blue △ from left to right) and $H_I(4, 17)$, $H_I(4, 19)$, $H_I(4, 23)$ (black □ from left to right).

From Fig. 1 we can easily see that:

- The parity-check matrix $H(4, q)$ lives up to and even performs better than our expectation. In practice, some signals with sparsity $k \geq r$ measured by $H(4, q)$ can also be recovered exactly by OMP.
- The performances of PAC and IAC defined in section II are almost the same.
- When q is small, $H(4, q)$ performs significantly better than the corresponding Gaussian matrix. However, let $r = 4$ be constant and increase q, the performance gain [2] of $H(4, q)$ is smaller than that of Gaussian matrices. When q increases to some extent, $H(4, q)$ will be worse than the corresponding Gaussian matrices.

Numbers of similar experiments have been conducted for $H(r, q)$ with other r and q, and the above phenomenon still exists. For the sake of the limits of space, simulations results for them are omitted here.

[2]Here, the performance gain means that given a sparsity order k of signals, the gain of the percentage of perfect recovery.

In the experiments in Fig. 1, we set $r = 4$ to be constant and let q vary. In the following, we will set q to be constant and let r vary.

Example 2: Set $q = 29$ to be constant, consider the PACs $H_P(4, q)$, $H_P(5, q)$, $H_P(6, q)$ and IACs $H_I(4, q)$, $H_I(5, q)$, $H_I(6, q)$, see Fig. 2.

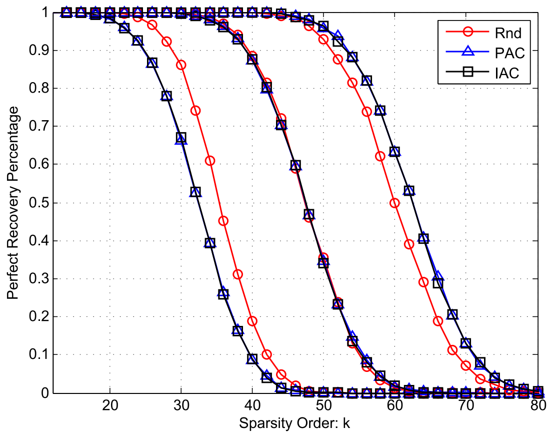

Fig. 2. Perfect recovery percentages of Gaussian random matrices of sizes 116×841, 145×841, 174×841 (red ○ from left to right), $H_P(4, 29)$, $H_P(5, 29)$, $H_P(6, 29)$ (blue △ from left to right) and $H_I(4, 29)$, $H_I(5, 29)$, $H_I(6, 29)$ (black □ from left to right).

It can be easily seen from Fig. 2 that:

- Similarly, all signals with $k < r$ measured by $H(r, 29)$ are exactly recovered and the maximum sparsity order of exactly recovered signals in practice is much larger than our estimate by Theorem 1.
- When r is small, measurement matrices based on array codes are a little worse than the corresponding Gaussian matrices while when r increases, the performance gain of $H(r, q)$ increases faster than that of Gaussian matrices and finally $H(r, q)$ will outperform Gaussian matrices.

Similarly, amounts of experiments have been done for other r and q, the same phenomenon can be observed.

It can be inferred from Fig. 1 and Fig. 2 that the relative performance of $H(r, q)$ and the corresponding Gaussian matrix may have something to do with the compression rate r/q. In order to investigate this assumption, we carried out a large number of experiments and identify those $H(r, q)$'s that perform not worse than Gaussian matrices for small r and q, see Table I [3]. We can imply from Table I that:

- No matter we fix r or q, there is a threshold of compression ratio r/q which is shown in italic (fix r) or bold (fix q) such that when the compression ratio is larger than that threshold, measurement matrices based on PAC will be better than Gaussian matrices.
- When the number of columns of $H_P(r, q)$ increases, i.e. q increases, the compression ratio r/q needed for $H_P(r, q)$ to outperforms Gaussian matrix decreases.

Similar results on IAC to the above can also be obtained and they are omitted for short of space.

Remark 6: The fact that the compression ratio r/q needed for $H(r, q)$ to be good decreases when q increases is of great significance in practice. One of the major superiorities of $H(r, q)$ over the random matrix relies on its great cost reduction of storage space when the number of columns is huge. In this situation, $H(r, q)$'s over a very large range of compression ratio will perform better than the corresponding Gaussian matrices, thus $H(r, q)$ can be applicable widely.

[3] The array code used to generate the data in this table is the PAC defined by (8).

C. Measurement Matrices Based on Shortened Array Codes

The previous part has illustrated the good performance of the full parity-check matrices of array codes. However, the number of columns of those matrices is fixed to be q^2, which means those matrices can only be used to measure a small class of signals. In the following, we will show that by removing several block-columns of those matrices they can be applied to much more signals without too much performance loss. The easiest way to remove block-columns is to remove the last several block-columns.

Example 3: Consider $H_P(5, 17)$, we can obtain two submatrices $H_P^{(13)}(5, 17)$ and $H_P^{(10)}(5, 17)$ by removing its last 4 and 7 block-columns, respectively. See Fig. 3 for the perfect recovery percentages of them and their corresponding Gaussian matrices.

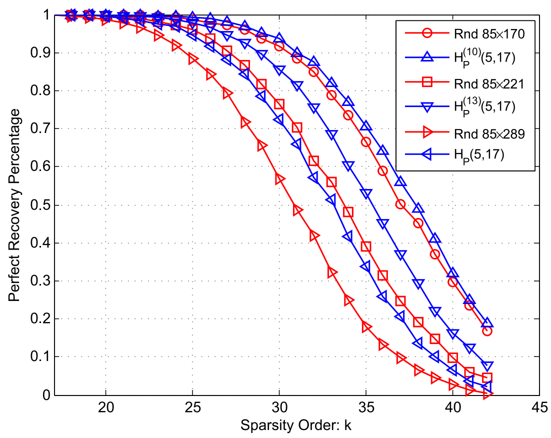

Fig. 3. Perfect recovery percentages of $H_P(5, 17)$, $H_P^{(13)}(5, 17)$ obtained by removing the last 4 block-columns of $H_P(5, 17)$, $H_P^{(10)}(5, 17)$ which retains the first 10 block-columns of $H_P(5, 17)$ and their corresponding Gaussian matrices.

It can be inferred from Fig. 3 that as long as $H(r, q)$ performs better than Gaussian matrices, many shortened submatrices of $H(r, q)$ will also have a good performance, though the performance gain over Gaussian matrices decreases when more block-columns are removed.

Large girth parity-check matrices are preferable in LDPC codes, so it is reasonable to expect them to perform better than small girth matrices in CS.

Example 4: Consider a PAC with parity-check matrix $H_P(3, 29)$, by taking its first 8 block-columns we can obtain a submatrix $H_P^{(8)}(3, 29)$ with girth 6. On the other hand, by taking block-columns $\{1, 2, 4, 5, 10, 11, 13, 14\}$ of $H_P(3, 29)$, we can get a submatrix $H_P^{(s)}(3, 29)$ with girth 8 according to [13, Table I]. Fig. 4 shows the perfect recovery percentages of $H_P^{(8)}(3, 29)$ and $H_P^{(s)}(3, 29)$.

Obviously, similar to the case in LDPC codes, $H_P^{(s)}(3, 29)$ with girth 8 performs better than $H_P^{(8)}(3, 29)$ with girth 6, which verifies that large girth has positive influence on the performance of binary measurement matrices.

TABLE I
THE COMPRESSION RATIO r/q OF MEASUREMENT MATRICES $H_P(r,q)$ WHICH PERFORM NO WORSE THAN THE CORRESPONDING GAUSSIAN MATRICES FOR SMALL r AND q.

r \ r/q \ q	11	13	17	19	23	29	31	37	41	43	47	53	59	61
2	**0.182**	***0.153***												
3	0.273	**0.231**												
4	0.364	0.308	**0.235**											
5	0.455	0.385	0.294	**0.263**	***0.217***									
6	0.545	0.462	0.353	0.316	0.261	**0.207**	**0.194**	**0.162**	**0.146**	***0.140***				
7	0.636	0.538	0.412	0.368	0.304	0.241	0.226	0.189	0.171	0.163	**0.149**	**0.132**	**0.119**	***0.115***

[1] Note that the compression ratios of matrices which perform worse than Gaussian matrices are not listed.

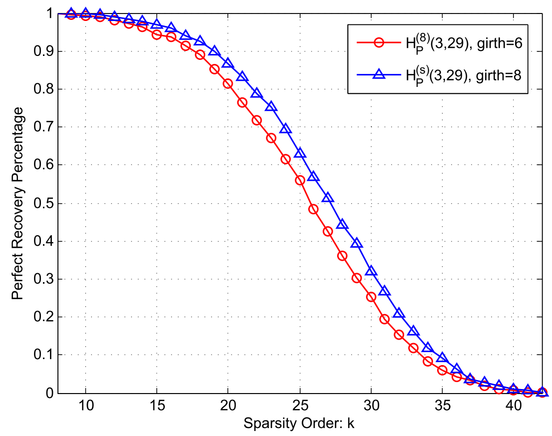

Fig. 4. Perfect recovery percentages of $H_P^{(8)}(3,29)$ with girth 6 obtained by taking the first 8 block-columns of $H_P(3,29)$ and $H_P^{(s)}(3,29)$ with girth 8 obtained by choosing the block-columns $\{1,2,4,5,10,11,13,14\}$ of $H_P(3,29)$.

IV. CONCLUSIONS

This paper has considered the application of parity-check matrices $H(r,q)$ of array codes into compressed sensing. The exact values of the sparks of $H(2,q)$ and $H(3,q)$ have been derived and two lower bounds of the spark of $H(r,q)$ or its submatrix for larger r have been given. The theoretical results about the spark of these matrices guarantee to some extent the good performance of them which has been verified by numbers of simulations. Besides the parity-check matrices of array codes themselves, many of their submatrices also show good performance, which enlarges the range of use of this kind of measurement matrices greatly. The binary measurement matrices constructed in this paper have perfect quasi-cyclic structures. Therefore, if used as measurement matrices, much storage space can be saved and the sampling process, even the recovering process will be much easier and simpler.

There seems to be a gap between theoretical and simulation results about the performance of the constructed matrices. Future work may include a better lower bound for the spark of these matrices so as to explain the simulation results better.

ACKNOWLEDGMENT

The authors wish to express their appreciation to the anonymous reviewers for their valuable suggestions and comments that helped to greatly improve the paper.

This research is supported in part by the 973 Program of China (2012CB315803), the Research Fund for the Doctoral Program of Higher Education of China (20100002110033), and the open research fund of National Mobile Communications Research Laboratory of Southeast University (2011D11). The author Shu-Tao Xia is also with the National Mobile Communications Research Laboratory of Southeast University of China.

REFERENCES

[1] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 489–509, Feb. 2006.

[2] D. L. Donoho,"Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[3] J. Tropp, "Greed is good: Algorithmic results for sparse approximation," *IEEE Trans. Inf. Theory*, vol. 50, no. 10, pp. 2231–2242, Oct. 2004.

[4] D. L. Donoho and M. Elad,"Optimally sparse representation in general (nonorthogonal) dictionaries via l^1 minimization," *Proc. Nat. Acad. Sci.*, vol. 100, no. 5, pp. 2197–2202, 2003.

[5] S. T. Xia, X. J. Liu, Y. Jiang and H. T. Zheng, "Sparks and deterministic constructions of binary measurement matrices from finite geometry," submitted, available online under http://arxiv.org/abs/1301.5952.

[6] E. J. Candès and T. Tao,"Decoding by linear programming," *IEEE Trans. Inform. Theory*, vol. 51, no. 12, pp. 4203–4215, Dec. 2005.

[7] W. Xu and B. Hassibi, "Compressed sensing over the Grassmann manifold: A unified analytical framework," in *Proc. 46th Allerton Conf. Commun., Control, Comput.*, Monticello, IL, Sep. 2008, pp. 562–567.

[8] A. Amini and F. Marvasti,"Deterministic construction of binary, bipolar, and ternary compressed sensing matrices," *IEEE Trans. Inf. Theory*, vol. 57, no. 4, pp. 2360–2370, Apr. 2011.

[9] S. Li, F. Gao, G. Ge, and S. Zhang, "Deterministic construction of compressed sensing matrices via algebraic curves," *IEEE Trans. Inf. Theory*, vol. 58, no. 8, pp. 5035–5041, Apr. 2012.

[10] A. G. Dimakis, R. Smarandache, and P. O. Vontobel,"LDPC codes for compressed sensing," *IEEE Trans. Inf. Theory*, vol. 58, no. 5, pp. 3093–3114, May 2012.

[11] W. Z. Lu, K. Kpalma and J. Ronisn, "Sparse binary matrices of LDPC codes for compressed sensing," in *Data Compression Conference* (DCC), Snowbird (Utah), United States, Apr. 2012.

[12] J. L. Fan, "Array codes as low-density parity-check codes," in *Proc.2nd Int. Symp. Turbo Codes*, Brest, France, Sept. 2000, pp. 545–546.

[13] O. Milenkovic, N. Kashyap and D. Leyba, "Shortened array codes of large girth," *IEEE Trans. Inf. Theory*, vol. 52, no. 8, pp. 3707–3722, Aug. 2006

[14] M. Esmaeili and M. J. Amoshahi, "On the stopping distance of array code parity-check matrices," *IEEE Trans. Inf. Theory*, vol. 55, no. 8, pp. 3488–3493, Aug. 2009.

[15] M. P. C. Fossorier, "Quasi-cyclic low-density parity-check codes from circulant permutation matrices," *IEEE Trans. Inf. Theory*, vol. 50, no. 8, pp. 1788–1794, Aug. 2004.

[16] M. Esmaeili, M. H. Tadayon, and T. A. Gulliver, "More on the stopping and minimum distances of array codes," *IEEE Trans. Commun.*, vol. 59, no. 3, pp. 750–757, Mar. 2011.

[17] E. Rosnes, M. A. Ambroze, and M. Tomlinson, "On the minimum/stopping distance of array low-density parity-check codes," submitted, available online under http://arxiv.org/abs/1205.2590.

[18] X. J. Liu, S. T. Xia, "Reconstruction guarantee analysis of binary measurement matrices based on girth," accepted by ISIT 2013, available online under http://arxiv.org/abs/1301.4926.pdf.