RIP-fulfilling Complex-Valued Matrices

Arash Amini, Vahid Montazerhodjat, Farokh Marvasti

Advanced Communications Research Institute (ACRI)
Department of Electrical Engineering, Sharif University of Technology, Tehran
Email: {arashsil, v_montazerhodjat}@ee.sharif.edu, marvasti@sharif.edu

***Abstract*— Although the theoretical results in the field of compressed sensing show that large classes of random matrices fulfill the so called Restricted Isometry Property (RIP) with high probability, only a few deterministic matrix designs are known. In this paper, we generalize one of the recent schemes based on binary BCH codes to p-ary codes which are useful for construction of complex sampling matrices. Though the design approach is similar, due to the use of p-ary codes (with p a prime power) and then complex matrices, the results are not similar. The new matrices are of the size $(p^a-1)\times p^b$ using a prime power p; the previous BCH structures are the special cases for $p=2^1$ which means that the new matrices provide more options in the number of samples.**

I. Introduction

Compressed Sensing (CS), a new framework for simultaneous sampling and compression of sparse signals, has been the center of immense attention for the past few years [1]–[3]. The linear CS theory addresses reliable reconstruction of a k-sparse $n\times 1$ source vector, namely $\mathbf{x}_{n\times 1}$, from its linear projections onto an m-dimensional subspace ($m\ll n$) which constitute an $m\times 1$ measurement vector ($\mathbf{y}_{m\times 1}$). In other words, m linear equations arranged in the form of $\mathbf{y}_{m\times 1}=\mathbf{\Phi}_{m\times n}\mathbf{x}_{n\times 1}$ are dealt in the CS field, where the matrix $\mathbf{\Phi}_{m\times n}$ is called the sampling (sensing) matrix.

The research work carried out in the CS area follows two major goals; first, proposing some efficient methods to retrieve the sparse source vector from the measurements with the minimum probability of error, and second, devising some sampling matrices mainly in order to have as minimum number of samples as possible. It has been proved that the general solution for the first concern is intractable [4]; however, under certain conditions, almost exact solutions for k-sparse source vectors can be achieved if the number of samples (m) exceeds the bound $\mathcal{O}(k\log(n/k))$. In this case, the previously mentioned linear equation system can be solved by the common ℓ_1-minimization methods (known as Basis Pursuit) to obtain sparse solutions [2], [4]. On the other hand, the second objective in the CS field, i.e., proposing sensing matrices has recently been investigated to have near minimal number of samples [5]–[8].

The so called Restricted Isometry Property (RIP) first introduced in [2] is a sufficient condition which entails the closeness of the acquired solution by the ℓ_1-minimization methods to the exact k-sparse solution, i.e., stable recovery. Simply speaking, the RIP of order k asserts the introduced transformation by the sampling matrix $\mathbf{\Phi}$ nearly preserves the ℓ_2-norm of all k-sparse vectors; i.e., a matrix $\mathbf{\Phi}_{m\times n}$ satisfies the RIP of order k with constant $0\le\delta_k<1$ if for every k-sparse vector $\mathbf{x}$, the following inequalities hold.

$$\forall \mathbf{x}_{n\times 1}: k\text{-sparse} \qquad 1-\delta_k\le\frac{\|\mathbf{\Phi x}\|_{\ell_2}^2}{\|\mathbf{x}\|_{\ell_2}^2}\le 1+\delta_k \tag{1}$$

Although it has been proved that the random sampling matrices satisfy the RIP with high probability [9], it is preferred to utilize RIP-fulfilling deterministic sensing matrices. This point is a direct result of the fact that exploiting random sampling matrices requires huge amount of storage and even if the storage is not the bottleneck, there are many cases in which implementing random sensing matrices in the hardware is infeasible. Moreover, deterministic matrices are likely to provide simplicity in the both sampling and reconstruction processes.

Recently, a few deterministic designs are investigated [5]–[8], [10], [11]. In [5], DeVore has proposed $p^2\times p^{r+1}$ binary matrices (prior to the column normalization) which satisfy the RIP of order k where $kr<p$. Exploiting the hash functions and extractor graphs, another class of binary matrices (prior to the column normalization) has been introduced in [10]. The authors in [6] has established a connection between CS and coding theory, specifically the second order Reed-Muller codes and have proposed a category of $2^l\times 2^{\frac{l(l+1)}{2}}$ deterministic sensing matrices. Some $m\times m^2$ complex-valued matrices have been investigated in [7] by taking advantage of chirp functions. Also, some bound-achieving matrices have recently been introduced in [11] which obey the statistical RIP (STRIP)[1] rather than the well-known RIP. There is a very recent work [8] in which by using Optical Orthogonal Codes (OOC) and binary BCH codes, the authors have devised some binary, bipolar, and ternary matrices (prior to the column normalization). The bipolar matrices are based on the binary BCH codes which is the main theme of this paper.

In this paper, we generalize the utilization of the binary BCH codes in [8] to the use of p-ary codes (with p a prime integer) to generate sampling matrices. This will result in a complex-valued class of $(p^a-1)\times p^b$ matrices which broadens the choices for the number of samples.

In this paper, in order to avoid confusion between the common parameters in the CS field and coding theory, the

[1] In the case of STRIP, the sampling matrix would satisfy the RIP with high probability.

associated parameters with the coding field have been marked by the tilde sign; e.g., $\tilde{n}$ represents the block length in the coding theory while n denotes the number of elements in the source vector.

II. Complex Matrices via p-ary Linear Codes

In this section, we show the link between the p-ary codes and RIP-fulfilling complex matrices. Since the approach is based on the one used for bipolar matrices introduced in [8], we briefly discuss the binary design concepts first.

Assume that we are given a $(\tilde{n}, \tilde{k})$ linear binary code with the minimum distance $\tilde{d}_{min}$ such that the all-one vector $(\mathbf{1}_{\tilde{n}\times 1})$ is a valid code word; due to the linearity of the code, all-zero vector $(\mathbf{0}_{\tilde{n}\times 1})$ is always a code word. Now for all pairs of code vectors such as $\mathbf{a}_{\tilde{n}\times 1}, \mathbf{b}_{\tilde{n}\times 1}$ with $\mathbf{c}_{\tilde{n}\times 1} \triangleq \mathbf{a} \oplus \mathbf{b}$ ($\oplus$ denotes the bitwise XOR operation) one of the following statements is true:

1) $\mathbf{c}_{\tilde{n}\times 1} = \mathbf{0}_{\tilde{n}\times 1}$ or $\mathbf{1}_{\tilde{n}\times 1}$.
2) $\mathbf{c}_{\tilde{n}\times 1} \neq \mathbf{0}_{\tilde{n}\times 1}$ and $\mathbf{c}_{\tilde{n}\times 1} \neq \mathbf{1}_{\tilde{n}\times 1}$, therefore:

$$\begin{cases} d(\mathbf{c}_{\tilde{n}\times 1}, \mathbf{0}_{\tilde{n}\times 1}) \geq \tilde{d}_{min} \\ d(\mathbf{c}_{\tilde{n}\times 1}, \mathbf{1}_{\tilde{n}\times 1}) \geq \tilde{d}_{min} \end{cases} \tag{2}$$

which means that $\mathbf{c}_{\tilde{n}\times 1}$ contains at least $\tilde{d}_{min}$ and at most $\tilde{n}-\tilde{d}_{min}$ number of ones (or zeros). In other words, $\mathbf{a}$ and $\mathbf{b}$ differ at least in $\tilde{d}_{min}$ and at most in $\tilde{n}-\tilde{d}_{min}$ bits.

For a given code word $\mathbf{a}$, the first case happens only when $\mathbf{b} = \mathbf{a}$ or $\mathbf{a} \oplus \mathbf{1}_{\tilde{n}\times 1}$; thus, all the possible $2^{\tilde{k}}$ code words can be paired ($\mathbf{a}$ with $\mathbf{a} \oplus \mathbf{1}_{\tilde{n}\times 1}$) into $2^{\tilde{k}-1}$ sets such that only the second case happens for two vectors from different sets. Now assume that we form a matrix by selecting one of the vectors in each set and convert all the zeros in the matrix into -1 ($\mathbf{A}_{\tilde{n}\times 2^{\tilde{k}-1}}$). The columns of $\mathbf{A}$ consist solely of ± 1 and as mentioned above, each two columns differ by at least $\tilde{d}_{min}$ and at most $\tilde{n}-\tilde{d}_{min}$ elements. Consequently, the absolute value of the inner product of each two distinct columns is upperbounded by $\tilde{n}-2\tilde{d}_{min}$. It is shown in [8] that by normalizing the columns of $\mathbf{A}$ (all the columns have the same norm and thus, normalization is equivalent to scaling), we obtain an $\tilde{n} \times 2^{\tilde{k}-1}$ matrix which obeys RIP of order $k < \frac{\tilde{n}}{\tilde{n}-2\tilde{d}_{min}} + 1$ with the constant $\delta_k = (k-1)\big(1 - 2\frac{\tilde{d}_{min}}{\tilde{n}}\big)$.

To generalize the mentioned results to p-ary (opposed to binary) codes, there are two difficulties: 1) the definition of $\tilde{d}_{min}$ in p-ary codes just reveals the number of unequal locations in two code words and does not give useful information about the differences (compare it to the binary case where unequality reveals almost everything) and 2) to have a matrix with low inner product among its columns, we need a transformation on the elements such as replacement of the zeros with -1 in the binary case. To solve the latter, we introduce complex matrices by converting the code elements into points on the unit circle while for the former, instead of pairing the code vectors, we have to define larger sets.

Let $\mathcal{C}(\tilde{n}, \tilde{k}; p)$ be a linear p-ary code over $GF(p)$ where p is a power of a prime integer with the minimum distance $\tilde{d}_{min}$ such that $\mathbf{1}_{\tilde{n}\times 1}$ is a valid code vector. Due to the linearity of the code, all the vectors $\mathbf{0}_{\tilde{n}\times 1}, \mathbf{1}_{\tilde{n}\times 1}, \ldots, (\mathbf{p}-\mathbf{1})_{\tilde{n}\times 1}$ are also code words. Similar to the binary case, for each two code vectors $\mathbf{a}_{\tilde{n}\times 1}$ and $\mathbf{b}_{\tilde{n}\times 1}$ with $\mathbf{c}_{\tilde{n}\times 1} \triangleq \mathbf{a} \oplus -\mathbf{b}$, one of the following statements hold[2]:

1) $\mathbf{c} = \mathbf{0}_{\tilde{n}\times 1}$ or $\mathbf{1}_{\tilde{n}\times 1}$ or $\ldots$ or $(\mathbf{p}-\mathbf{1})_{\tilde{n}\times 1}$
2) $\mathbf{c} \notin \{\mathbf{0}_{\tilde{n}\times 1}, \mathbf{1}_{\tilde{n}\times 1}, \ldots, \text{or } (\mathbf{p}-\mathbf{1})_{\tilde{n}\times 1}\}$; therefore

$$\begin{cases} d(\mathbf{c}_{\tilde{n}\times 1}, \mathbf{0}_{\tilde{n}\times 1}) \geq \tilde{d}_{min} \\ d(\mathbf{c}_{\tilde{n}\times 1}, \mathbf{1}_{\tilde{n}\times 1}) \geq \tilde{d}_{min} \\ \vdots \\ d(\mathbf{c}_{\tilde{n}\times 1}, (\mathbf{p}-\mathbf{1})_{\tilde{n}\times 1}) \geq \tilde{d}_{min} \end{cases} \tag{3}$$

which means that $\mathbf{c}_{\tilde{n}\times 1}$ contains at most $\tilde{n}-\tilde{d}_{min}$ from each of $\{0, 1, \ldots, p-1\}$. Let N_i $(0 \leq i \leq p-1)$ represent the number of occurrences of the element i in the vector $\mathbf{c}_{\tilde{n}\times 1}$. The inequalities $N_i \leq \tilde{n}-\tilde{d}_{min}$ together with $\sum_{i=1}^{p-1} N_i = \tilde{n}$ results in:

$$N_i = \tilde{n} - \sum_{j\neq i} N_j \geq \tilde{n} - (p-1)(\tilde{n}-\tilde{d}_{min}) \tag{4}$$

Hence

$$\underbrace{\tilde{n} - (p-1)(\tilde{n}-\tilde{d}_{min})}_{N_{min}} \leq N_i \leq \underbrace{\tilde{n}-\tilde{d}_{min}}_{N_{max}} \tag{5}$$

which is equivalent to

$$\big|N_i - \frac{N_{min}+N_{max}}{2}\big| \leq \frac{N_{max}-N_{min}}{2} \tag{6}$$

Similar to pairing in the binary codes, we divide the code vectors into sets of the form $\{\mathbf{a}, \mathbf{a}\oplus\mathbf{1}_{\tilde{n}\times 1}, \ldots, \mathbf{a}\oplus(\mathbf{p}-\mathbf{1})_{\tilde{n}\times 1}\}$ and pick exactly one vector from each set. In fact we are looking for representatives of the elements of the quotient group formed by dividing the group of all code vectors[3] to its subgroup $\{\mathbf{0}_{\tilde{n}\times 1}, \ldots, (\mathbf{p}-\mathbf{1})_{\tilde{n}\times 1}\}$. The following theorem summarizes the main results.

***Theorem** 1:* Let $\mathcal{C}(\tilde{n}, \tilde{k}; p)$ be a linear p-ary code over $GF(p)$ for a prime power p with the minimum distance $\tilde{d}_{min}$ such that $\mathbf{1}_{\tilde{n}\times 1}$ is a valid code word and let $\tilde{\mathbf{A}}_{\tilde{n}\times p^{\tilde{k}-1}}$ be the matrix generated by selecting exactly one vector from each set of $\{\mathbf{a}, \mathbf{a}\oplus\mathbf{1}_{\tilde{n}\times 1}, \ldots, \mathbf{a}\oplus(\mathbf{p}-\mathbf{1})_{\tilde{n}\times 1}\}$. Construct $\mathbf{A}_{\tilde{n}\times p^{\tilde{k}-1}}$ from $\tilde{\mathbf{A}}$ as:

$$\tilde{\mathbf{A}} = [\tilde{a}_{\alpha\beta}]_{\alpha,\beta} \Rightarrow \mathbf{A} = \frac{1}{\sqrt{\tilde{n}}}\big[e^{j\frac{2\pi}{p}\tilde{a}_{\alpha\beta}}\big]_{\alpha,\beta} \tag{7}$$

Now $\mathbf{A}$ satisfies RIP of order $k < \frac{2\tilde{n}}{p(p-1)\tilde{n}-p^2\tilde{d}_{min}} + 1$ with the constant $\delta_k = (k-1)\frac{p(p-1)\tilde{n}-p^2\tilde{d}_{min}}{2\tilde{n}}$.

Proof. First note that the columns of $\mathbf{A}$ are all normal:

$$\|\mathbf{a}_\beta\| = \left\| \frac{1}{\sqrt{\tilde{n}}}[e^{j\frac{2\pi}{p}\tilde{a}_{1,\beta}} \ \ldots \ e^{j\frac{2\pi}{p}\tilde{a}_{\tilde{n},\beta}}]^T \right\| = 1 \tag{8}$$

Let $\mathbf{a}_\alpha, \mathbf{a}_\beta$ be two different columns of $\mathbf{A}$ and let $\tilde{\mathbf{a}}_\alpha, \tilde{\mathbf{a}}_\beta$ be the corresponding columns in $\tilde{\mathbf{A}}$ with $\mathbf{c} = \tilde{\mathbf{a}}_\alpha \oplus -\tilde{\mathbf{a}}_\beta$.

[2]For p-ary codes, $\oplus$ is the element-wise addition $mod\ p$

[3]Algebraic group with respect to the operation $\oplus$

Moreover, assume that the element i ($0 \le i \le p-1$) is repeated N_i times in $\mathbf{c}$. For the inner product of $\mathbf{a}_\alpha, \mathbf{a}_\beta$ we have:

$$\begin{aligned} |\langle \mathbf{a}_\alpha, \mathbf{a}_\beta \rangle| &= |\mathbf{a}_\beta^H . \mathbf{a}_\alpha| = \Big| \sum_{i=1}^{\tilde{n}} e^{j\frac{2\pi}{p}(\tilde{a}_{i,\alpha} - \tilde{a}_{i,\beta})} \Big| \\ &= \Big| \sum_{i=1}^{\tilde{n}} e^{j\frac{2\pi}{p} c_i} \Big| = \Big| \sum_{i=0}^{p-1} N_i e^{j\frac{2\pi}{p} i} \Big| \end{aligned} \quad (9)$$

Since $e^{j\frac{2\pi}{p}}$ is the root of $1 + x + \cdots + x^{p-1}$, for all values of γ we have:

$$\Big| \sum_{i=0}^{p-1} N_i e^{j\frac{2\pi}{p} i} \Big| = \Big| \sum_{i=0}^{p-1} (N_i - \gamma) e^{j\frac{2\pi}{p} i} \Big| \le \sum_{i=0}^{p-1} |N_i - \gamma| \quad (10)$$

Using inequalities (5) and (6), by setting $\gamma = \frac{N_{min} + N_{max}}{2}$ we get:

$$\begin{aligned} \Big| \sum_{i=0}^{p-1} N_i e^{j\frac{2\pi}{p} i} \Big| &\le p \, \frac{N_{max} - N_{min}}{2} \\ &= \frac{p(p-1)n - p^2 d_{min}}{2} \end{aligned} \quad (11)$$

which demonstrates the following upper bound on the inner product of the columns of $\mathbf{A}$:

$$|\langle \mathbf{a}_\alpha, \mathbf{a}_\beta \rangle| \le \frac{p(p-1)\tilde{n} - p^2 \tilde{d}_{min}}{2} \quad (12)$$

Similar to the arguments in [8], we conclude the mentioned RIP order and constant from the above inequality and the proof in complete ■

Remark 1: The best choice of γ in (10) which yields the lowest upper bound for the inner product is the median of the N_i's, not necessarily the used value; however, the median is not a fixed value and thus, no deterministic upper bound will be achieved.

Remark 2: For a desired RIP order of $k > 1$ using the code length $\tilde{n}$, we should have (Theorem 1):

$$\frac{\tilde{d}_{min}}{\tilde{n}} > \frac{p-1}{p} - \frac{2}{p^2(k-1)} \ge \frac{p-1}{p}(1 - \frac{8}{kp^2}) \quad (13)$$

Hence, $\tilde{d}_{min}$ should be close to $\frac{p-1}{p}\tilde{n}$; i.e., for large values of p, $\tilde{d}_{min}$ is almost the same as $\tilde{n}$. this means that we need linear codes with very large minimum distances. The existence of such codes will be shown in the next section.

III. p-ary Code Design

Due to the existence of a lower bound on the minimum distance of the BCH codes, we focus on the generalized p-ary BCH codes with large minimum distances. Again the design approach is similar to the binary case.

The code vectors of a p-ary BCH code are vectors of length $\tilde{n} = p^{\tilde{m}} - 1$ for an integer $\tilde{m}$ with the elements in $GF(p)$ such that if the vectors are regarded as polynomials of degree $\tilde{n} - 2$, they are divisible by a specific polynomial $g(x) \in GF(p)[x]$ known as code generating polynomial. The main concept behind this choice of $\tilde{n}$ is the design tools in the field $GF(p^{\tilde{m}})$; in fact, recalling a result from the Galois theory, we know:

$$\prod_{\substack{r \in GF(p^{\tilde{m}}) \\ r \ne 0}} (x - r) = x^{p^{\tilde{m}} - 1} - 1 \quad (14)$$

The code generating polynomial for these codes is chosen in such a way that

$$g(x) \big| x^{p^{\tilde{m}} - 1} - 1 \quad (15)$$

Thus, considering (14), it is evident that all the roots of $g(x)$ lie in $GF(p^{\tilde{m}})$; i.e., $g(x)$ can be decomposed into linear factors in this field. This feature is helpful in designing the polynomial by determining its roots.

Let α be a primitive root of the field $GF(p^{\tilde{m}})$; hence, all the nonzero elements of the field can be written in the form α^l where l is a nonnegative integer number. An important result in BCH codes is that if $\alpha^{i_1}, \ldots, \alpha^{i_d}$ is a subset of the roots of $g(x)$ such that $\{i_1, \ldots, i_d\}$ form an arithmetic progression, we have $\tilde{d}_{min} \ge d+1$; if the vector $[c_1, \ldots, c_{\tilde{n}}]^T$ is a nonzero code word, we should have $g(x) | \sum_{j=1}^{\tilde{n}} c_j x^{j-1}$ and therefore:

$$\underbrace{\begin{bmatrix} \alpha^{0\times i_1} & \alpha^{1\times i_1} & \ldots & \alpha^{(\tilde{n}-1)\times i_1} \\ \alpha^{0\times i_2} & \alpha^{1\times i_2} & \ldots & \alpha^{(\tilde{n}-1)\times i_2} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha^{0\times i_d} & \alpha^{1\times i_d} & \ldots & \alpha^{(\tilde{n}-1)\times i_d} \end{bmatrix}}_{\mathbf{H}_{d\times\tilde{n}}} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_{\tilde{n}} \end{bmatrix} = \mathbf{0}_{d\times 1} \quad (16)$$

Since $\{i_1, \ldots, i_d\}$ form an arithmetic progression, each $d \times d$ submatrix of $\mathbf{H}$ is a Vandermond matrix; thus, each d selection of the columns are linearly independent which means that at least $d+1$ elements in $[c_1, \ldots, c_{\tilde{n}}]^T$ should be nonzero (the lower bound on the minimum distance).

In our code design problem, we choose $g(x)$ such that the set $\{\alpha^{p^{\tilde{m}-1} + \frac{p^l-1}{p-1}+1}, \alpha^{p^{\tilde{m}-1} + \frac{p^l-1}{p-1}+2}, \ldots, \alpha^{p^{\tilde{m}}-2}\}$ be a subset of its roots for an integer $l < \tilde{m}$. In this way, there exists at least an arithmetic progression of length $p^{\tilde{m}} - p^{\tilde{m}-1} - \frac{p^l-1}{p-1} - 2$ among the powers of α in the roots of $g(x)$. Consequently, we have

$$\begin{aligned} \tilde{d}_{min} &\ge p^{\tilde{m}} - p^{\tilde{m}-1} - \frac{p^l - 1}{p-1} - 1 \\ &= (p^{\tilde{m}} - 1)\left(1 - \frac{p^{\tilde{m}-1}}{p^{\tilde{m}} - 1} - \frac{p^l - 1}{(p^{\tilde{m}} - 1)(p-1)}\right) \\ &= \tilde{n}\left(\frac{p-1}{p} - \frac{p^{l+1} - 1}{p(p-1)(p^{\tilde{m}} - 1)}\right) \\ \Rightarrow \frac{\tilde{d}_{min}}{\tilde{n}} &\ge \frac{p-1}{p}\left(1 - \frac{p^{l+1} - 1}{(p-1)^2(p^{\tilde{m}} - 1)}\right) \end{aligned} \quad (17)$$

To find such a generating polynomial, we construct a polynomial $h(x) \in GP(p)[x]$ (parity check polynomial) without any repeated root such that the roots of $h(x)$ form a subset of $\{\alpha^0, \alpha^1, \ldots, \alpha^{p^{\tilde{m}-1} + \frac{p^l-1}{p-1}}\}$. Now $g(x) \triangleq \frac{x^{p^{\tilde{m}}-1} - 1}{h(x)}$ satisfies all the mentioned requirements for the code generating

polynomial. The reason that $h(x)$ can not contain all the mentioned roots is that the coefficients of the parity check polynomial must belong to $GF(p)$ rather than $GF(p^{\tilde{m}})$.

Let $\mathcal{H}_{seq}^{(\tilde{m},l)}$ be the set of all binary sequences of length $\tilde{m}$ such that each two 1's are circularly spaced by at least $\tilde{m}-l-1$ zeros in between. Furthermore, let $\mathcal{H}_{\tilde{m}}^{(l)}$ be the set of all decimal numbers for which the base-p representation coincides with a sequences in $\mathcal{H}_{seq}^{(\tilde{m},l)}$. We show that $\mathcal{H}_{seq}^{(\tilde{m},l)} \subseteq \{0,1,\ldots,p^{\tilde{m}-1}+\frac{p^l-1}{p-1}\}$. Let B be an element of $\mathcal{H}_{seq}^{(\tilde{m},l)}$ with the base-p representation as $(\overline{b_{\tilde{m}-1}\ldots b_0})_p$. Since the sequence $(b_{\tilde{m}-1},\ldots,b_0)$ is a member of $\mathcal{H}_{seq}^{(\tilde{m},l)}$, each of the b_i's is either 0 or 1. We have two cases:

1) $b_{\tilde{m}-1}=0$, therefore

$$\begin{aligned}(\overline{b_{\tilde{m}-1}\ldots b_0})_p &\leq (\overline{011\ldots1})_p \\ &= \frac{p^{\tilde{m}-1}}{p-1} \leq p^{\tilde{m}-1}+\frac{p^l-1}{p-1} \quad (18)\end{aligned}$$

2) $b_{\tilde{m}-1}=1$, therefore, the following $\tilde{m}-l-1$ digits should be zero: $b_{\tilde{m}-2}=\cdots=b_l=0$

$$\begin{aligned}(\overline{b_{\tilde{m}-1}\ldots b_0})_p &\leq (\overline{1\underbrace{0\ldots0}_{\tilde{m}-l-1}\underbrace{1\ldots1}_{l}})_p \\ &= p^{\tilde{m}-1}+\frac{p^l-1}{p-1} \quad (19)\end{aligned}$$

In addition, for the same B, we have:

$$\begin{aligned}pB &= (\overline{b_{\tilde{m}-1}\ldots b_0 0})_p \\ &= b_{\tilde{m}-1}p^{\tilde{m}}+(\overline{b_{\tilde{m}-2}\ldots b_0 0})_p \\ &\equiv b_{\tilde{m}-1}+(\overline{b_{\tilde{m}-2}\ldots b_0 0})_p \pmod{p^{\tilde{m}}-1} \\ &\equiv (\overline{b_{\tilde{m}-2}\ldots b_0 b_{\tilde{m}-1}})_p \pmod{p^{\tilde{m}}-1} \quad (20)\end{aligned}$$

According to the circular property of $(b_{\tilde{m}-1},\ldots,b_0)$, $B^{'}=(\overline{b_{\tilde{m}-2}\ldots b_0 b_{\tilde{m}-1}})_p$ should be also included in $\mathcal{H}_{\tilde{m}}^{(l)}$, hence

$$\begin{aligned}&\alpha^{pB}=\alpha^{B^{'}}\in\{\alpha^h\}_{h\in\mathcal{H}_{\tilde{m}}^{(l)}}, \\ \Rightarrow\ &\{\alpha^B,\alpha^{pB},\alpha^{p^2B},\ldots,\alpha^{p^{\tilde{m}-1}B}\}\subseteq\{\alpha^h\}_{h\in\mathcal{H}_{\tilde{m}}^{(l)}} \quad (21)\end{aligned}$$

In fact the set of $\{\alpha^{p^iB}\}_i$ is the set of conjugates of α^B with respect to the field $GF(p)$, therefore

$$\prod_i (x-\alpha^{p^iB})\in GF(p)[x] \quad (22)$$

This confirms that the following construction for $h(x)$ fulfills all the required conditions:

$$h(x)\triangleq\prod_{h\in\mathcal{H}_{\tilde{m}}^{(l)}}(x-\alpha^h)\in GF(p)[x] \quad (23)$$

It should be noticed that since $\mathcal{H}_{\tilde{m}}^{(l)}\subseteq\{0,1,\ldots,p^{\tilde{m}-1}+\frac{p^l-1}{p-1}\}$, the roots of the above $h(x)$ belong to $\{\alpha^0,\ldots,\alpha^{p^{\tilde{m}-1}+\frac{p^l-1}{p-1}}\}$ which was a required condition.

To find the final size of the constructed matrix, we should calculate the value of $\tilde{k}$; similar to the discussions in [8], this value is the same as the size of the set $\mathcal{H}_{\tilde{m}}^{(l)}$. It is shown in [8] that $|\mathcal{H}_{\tilde{m}}^{(l)}|=\mathcal{O}\Big(\big(\frac{\tilde{m}-l}{2}+1\big)^{\frac{l}{\tilde{m}-l}}\Big)$.

One of the most important conditions to be checked is that whether $\mathbf{1}_{\tilde{n}\times1}$ is a valid code word. Since the base-p representation of 0 satisfies the required conditions of $\mathcal{H}_{seq}^{(\tilde{m},l)}$, $1=\alpha^0$ is one of the roots of $h(x)$ which implies that $gcd\big(g(x),x-1\big)=1$. Due to the definition of $g(x)$ we know

$$\begin{aligned}&\begin{cases} g(x)\big|x^{\tilde{n}}-1=(x-1)(1+x+\cdots+x^{\tilde{n}-1}) \\ gcd\big(g(x)\ ,\ x-1\big)=1 \end{cases} \\ &\Rightarrow g(x)\big|1+x+\cdots+x^{\tilde{n}-1} \quad (24)\end{aligned}$$

which confirms that $\mathbf{1}_{\tilde{n}\times1}$ is a valid code word. the other issue which should be considered is to choose the representatives from each of the sets $\{\mathbf{a},\mathbf{a}\oplus\mathbf{1}_{\tilde{n}\times1},\ldots\mathbf{a}\oplus(\mathbf{p}-\mathbf{1})_{\tilde{n}\times1}\}$. Since $p\nmid\tilde{n}$, in each of these sets, the polynomial representation of exactly one of the code words is divisible by $x-1$. Hence, if instead of $g(x)$ we use $(x-1)g(x)$, all the desired conditions are fulfilled. In addition, by this choice of the code generating polynomial, the cyclic property of the original code is preserved which is a useful tool for reducing the complexity of the reconstruction method [8]. Also the additional factor of $x-1$ increases the lower bound on the minimum distance of the code by 1.

It is shown in [8] that the family of Matching Pursuit (MP) methods will perfectly recover the original k-sparse vector from the noiseless samples if the sampling matrix obeys RIP of order $2k$. This is a general result which also applies for the matrices in this paper.

IV. Simulation Results

In this section, we compare the performance of our proposed sampling matrices based on the p-ary BCH codes with the performance of the matrices introduced in [7] as the two members of the complex-valued sensing matrices. We have also investigated the improvements achieved by our method compared to random sampling matrices. In order to have a fair comparison, we have focused on the special case of matrices designed for $n=3^7$ and $k=3$ based on the 3-ary BCH codes (with the primitive polynomial represented by the vector [2 0 0 1 1]$\in GF(3)$) which results in a class of matrices of size 80×729. Therefore, we have generated matrices of size 80×729 based on the chirp functions, 3-ary BCH codes and of random structure and then, have compared their performances. It is noteworthy that the Orthogonal Matching Pursuit (OMP) has been exploited to reconstruct the source vector from the measurements and for each figure in the following, the results are obtained by averaging over 5000 different runs with different input signals.

Figure 1 presents the reconstruction SNRs for the input signals with different sparsity orders. The samples, prior to reconstruction, have been subject to Additive White Gaussian Noise (AWGN) resulting in an input SNR of 12 dB at the beginning of the reconstruction step. This figure clearly represents that the source vector is perfectly recovered for the

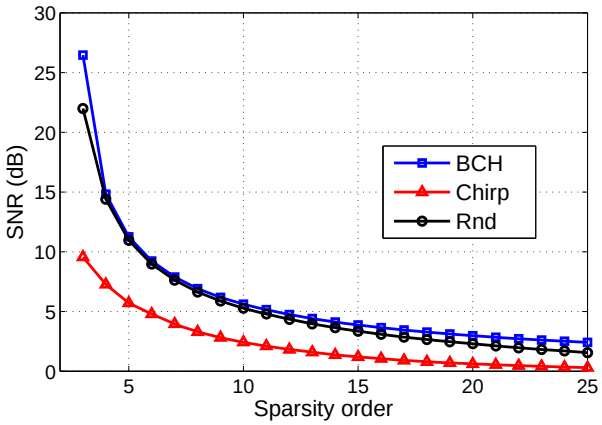

Fig. 1. The reconstruction SNR vs. different sparsity orders where the noisy compressed samples have SNR of 12 dB. The BCH-based satisfy the RIP of order 3.

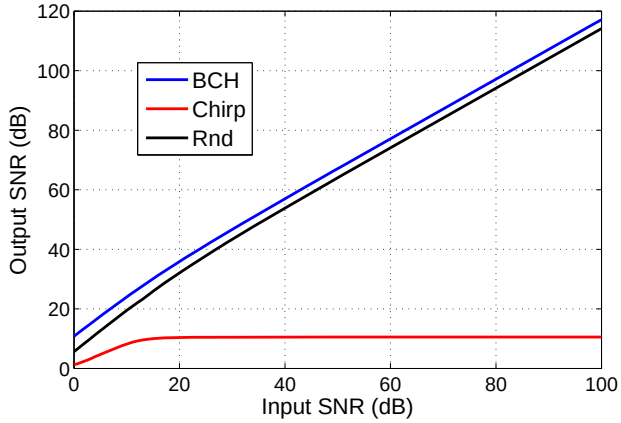

Fig. 2. The reconstruction SNR of a 3-sparse signal from its noisy compressed samples with different input SNRs. The BCH matrix satisfy the RIP of order 3.

sparsity order of 3 in the cases the sensing matrix design is based on the BCH codes or when the matrix is random. However, for the matrix produced by the chirp functions, a much weaker performance in reconstructing the original vector can be observed. The robustness of the proposed method against increasing sparsity orders can be inferred from this figure just by considering its performance over different sparsity orders. Moreover, it is worth mentioning that the proposed method results in greater SNRs than the other two methods over different sparsity orders.

To manifest the performance of the proposed BCH-based matrices over different SNRs, Fig. 2 has been included. In this case, the input signal is a 3-sparse signal and after sampling, the compressed samples are corrupted by AWGN of different powers, resulting in a range of SNRs. The chirp and BCH matrices have been designed to have the RIP of order 3. The proposed matrices in this paper together with the random matrices achieve increasing reconstruction SNRs by increasing the input SNR, while the chirp-based matrices reach a saturated level. Again, in this figure, the superior performance of the BCH-based matrices is clearly recognized.

V. Conclusion

A new design for the RIP-fulfilling matrices is investigated which in general, results in complex-valued matrices. The design is based on the previously studied link between coding theory and compressed sensing. The considered codes are generalized p-ary BCH codes which provide large minimum distance between the code vectors. For the case of $p = 2$, the special case of bipolar matrices occurs which was previously investigated. Simulation results confirm that the performance of the new codes is slightly better than the random i.i.d. Gaussian matrices while they completely outperform the chirp-type matrices which are the common candidate for the complex sampling matrices.

References

[1] D. Donoho, "Compressed sensing," *IEEE Trans. Inform. Theory*, vol. 52, no. 4, pp. 1289–1306, April 2006.

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

[3] E. Candes and T. Tao, "Near optimal signal recovery from random projections: Universal encoding strategies," *IEEE Trans. Inform. Theory*, vol. 52, no. 12, pp. 5406–5425, Dec. 2006.

[4] ——, "Decoding by linear programming," *IEEE Trans. Inform. Theory*, vol. 51, no. 12, pp. 4203–4215, Dec. 2005.

[5] R. A. DeVore, "Deterministic construction of compressed sensing matrices," *Journal of Complexity*, vol. 23, no. doi:10.1016/j.jco.2007.04.002, pp. 918–925, March 2007.

[6] S. D. Howard, A. R. Calderbank, and S. J. Searle, "A fast reconstruction algorithm for deterministic compressive sensing using second order reed-muller codes," in *IEEE Conf. on Inform. Sciences and Systems (CISS2008)*, 2008.

[7] L. Applebauma, S. D. Howardb, S. Searlec, and R. Calderbank, "Chirp sensing codes: Deterministic compressed sensing measurements for fast recovery," *Applied and Computational Harmonic Analysis*, vol. 26, no. 2, pp. 283–290, March 2009.

[8] A. Amini and F. Marvasti, "Deterministic construction of binary, bipolar and ternary compressed sensing matrices," *preprint arXiv:0908.2676*, Aug. 2009.

[9] R. Baraniuk, M. Davenport, R. DeVore, and M. B. Wakin, "A simple proof of the restricted isometry property for random matrices," *Constr. Approx.*, vol. 28, no. 3, pp. 253–263, Dec. 2008.

[10] P. Indyk, "Explicit constructions for compressed sensing of sparse signals," in *ACM-SIAM symp. on Discrete Algorithms*, 2008, pp. 30–33.

[11] R. Calderbank, S. Howard, and S. Jafarpour, "Deterministic compressive sensing with groups of random variables," *http://www.dsp.ece.rice.edu/files/cs/strip-more.pdf*, 2009.