Matrices with small Coherence using p-ary Block Codes

Arash Amini*, Vahid Montazerhodjat, and Farokh Marvasti, *Senior Member, IEEE*

***Abstract*—In contrast to the vast amount of literature in random matrices in the field of compressed sensing, the subject of deterministic matrix design is at its early stages. Since these deterministic matrices are usually constructed using the polynomials in finite Galois fields, the number of rows (number of samples) is restricted to some specific integers such as prime powers. In this paper, besides extending a previous matrix design based on the binary BCH codes to the p-ary codes, we introduce matrices with wide variety of options for the number of rows. Simulation results demonstrate that these matrices perform almost as well as random matrices.**

***Index Terms*—Compressed Sensing, p-ary BCH codes, Coherences.**

I. Introduction

The technique of compression while sampling, usually referred to as *Compressed Sensing*, has been the center of attention for at least half a decade [1]–[3]. In fact, the compressibility of the discrete data associated with an analog signal such as speech and image indicates that the sampling procedure is not as efficient as possible; i.e., instead of compressing the data after the sampling procedure, there should be a method to combine these two tasks (sampling and compression) in order to somehow decrease the rate.

In the field of discrete compressed sensing, we are interested in reconstructing 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 ($\mathbf{y}_{m\times 1}$). The measurement process is theoretically assumed to be linear in the form of $\mathbf{y}_{m\times 1} = \mathbf{\Phi}_{m\times n}\mathbf{x}_{n\times 1}$, where $\mathbf{\Phi}_{m\times n}$ is called the sensing matrix. Moreover, the vector $\mathbf{x}$ is assumed to be k-sparse which means that $\mathbf{x}$ has a sparse representation in a (known) unitary domain, namely, $\mathbf{x}_{n\times 1} = \mathbf{\Psi}_{n\times n}\mathbf{s}_{n\times 1}$ where $\mathbf{s}$ has at most k nonzero elements and $\mathbf{\Psi}$ is a unitary matrix. In this paper, we assume that $\mathbf{\Psi}$ is the identity matrix or equivalently, we are considering $\mathbf{\Phi}$ instead of $\mathbf{\Phi\Psi}$.

The two main problems in the discrete compressed sensing are the sampling and reconstruction tasks. The sampling part consists of designing a proper sensing matrix $\mathbf{\Phi}_{m\times n}$ with small enough m (number of samples) that preserves the main information conveyed by the original signal. The common solution is to use a random matrix of i.i.d. elements with $m \geq \mathcal{O}(k\log n)$ (we hereafter refer to this inequality as the bound in the random theory); the Gaussian distribution is probably the first studied case [1], however, a large class of distributions are investigated in [4]. The reconstruction challenge is to recover the original vector from an under-determined system of linear equations (m equations vs. n unknowns) with the additional sparsity constraint. This problem has a longer research history as it also appears in the source separation problems. Although the mentioned problem is intractable in general [5], under certain conditions, it is shown that ℓ_1 minimization (basis pursuit) can yield the desired result [2], [5]. Also the greedy algorithms such as matching pursuit and its variants, due to their reasonable computational complexity, are among the well-known techniques in this field [6].

Although a realization of a random sensing matrix, with high probability provides the possibility of perfect recovery for all k-sparse vectors with small enough value of k, there is currently no polynomial-time algorithm to verify this property for a given matrix. The main benefit of deterministic designs is that stable recovery of sparse vectors can be guaranteed without any probabilistic arguments. Among other advantages of the deterministic designs is the storage issue; to store a realization of a random matrix, all the elements should be kept in the memory and the process should be repeated each time a new realization is generated, while in deterministic designs, because of the special structure of the matrix, only a few parameters should be stored. Furthermore, deterministic matrices are likely (eg. the matrices introduced in this paper) to provide simplicity in both sampling and reconstruction processes.

One of the main tools for investigating the suitability of a given matrix as a sensing operator is the so called Restricted Isometry Property (RIP) introduced in [2]: the matrix $\mathbf{\Phi}_{m\times n}$ is said to satisfy the RIP of order k with constant $0 \leq \delta_k < 1$ if for every k-sparse vector $\mathbf{s}$, the following inequalities hold:

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

It should be mentioned that RIP is only a necessary condition that guarantees recovery; there are examples where RIP-less guarantees support special type of sensing matrices [7]–[10].

The *coherence* of a matrix defined as

$$\mu_{\mathbf{A}} \triangleq \max_{i\neq j} \frac{|\langle \mathbf{a}_i, \mathbf{a}_j\rangle|}{\|\mathbf{a}_i\|\cdot\|\mathbf{a}_j\|}, \tag{2}$$

where $\mathbf{a}_i, \mathbf{a}_j$ are different columns of $\mathbf{A}$, is one of the main tools for establishing the RIP in deterministic matrices. In

The authors are with the Advanced Communication Research Institute (ACRI), Electrical Engineering Department, Sharif University of Technology, Azadi Ave., Tehran, Iran e-mail: {arashsil,v_montazerhodjat}@ee.sharif.edu, marvasti@sharif.edu.

Manuscript received April 9, 2010; revised Nov. 10, 2010, Feb. 1, 2011 and Aug. 13, 2011.

This work is supported by the Iranian Telecommunication Research Center (ITRC).

A part of this work was presented at ICC2010.

plain words, matrices with normalized columns are guaranteed to satisfy an RIP order when the coherence is small [11], [12]. There is a well-known lower bound on the coherence of an $m \times n$ ($m < n$) matrix $\mathbf{A}$ known as Welch bound [13]:

$$\mu_{\mathbf{A}} \geq \sqrt{\frac{n}{m(n-m)}} \quad . \tag{3}$$

The above lower bound implies an upper bound on the provable RIP order of a matrix through coherence arguments. Unfortunately, this upper bound is proportional to $\sqrt{m}$ while for the random matrices, the upper bound scales like m; i.e., there is inherently a gap between the RIP orders that are guaranteed for deterministic designs and the ones predicted by the random matrices.

In [11], using the coherence arguments, DeVore has proposed $p^2 \times p^{r+1}$ binary matrices with coherence $\frac{r}{p}$ that satisfy the RIP of order k when $kr < p$. Exploiting the hash functions and extractor graphs, another class of binary matrices with $m = k2^{\mathcal{O}(\log\log n)^E}$ ($m \times n$ matrix with RIP of order k) has been introduced in [14]; here E is a constant larger than 1 which is involved in the construction of the extractor graphs (the best known guarantee is $E = 2$). In addition to the extractor graphs, expander graphs are also shown to be useful for sensing purposes [7], [15]. The authors in [16] have established a connection between Compressed Sensing (CS) and coding theory, specifically the second order Reed-Muller codes and have proposed a category of bipolar $2^l \times 2^{\frac{l(l+1)}{2}}$ deterministic sensing matrices; however, no lower bound on the RIP order of these matrices is proved. Some $m \times m^2$ complex-valued matrices have been investigated in [17] by taking advantage of chirp functions; although there is no guarantee for the RIP order of these matrices, in [18], a relaxed version of the RIP known as Statistical RIP (StRIP[1]), is shown to hold. In fact, a more general class of StRIP matrices are introduced in [18]: it is shown that if 1) the rows of a matrix $\mathbf{A}_{m\times n}$ are orthogonal and all the row sums are zero, 2) the columns of the matrix form a group under point-wise multiplication, and 3) the absolute value of the column sums except the all-one column, are upper bounded by $m^{1-0.5\eta}$ for $\eta > 0.5$, then the inequalities in (1) hold with high probability over all k-sparse vectors for the matrix $\frac{1}{\sqrt{m}}\mathbf{A}$ when $k < 1 + (n-1)\eta$ and $m \geq \left(\frac{c}{\delta^2} k \log n\right)^{\frac{1}{\eta}}$ for some constant c.

In [12], using coherence arguments and based on BCH codes, we have recently introduced $(2^l - 1) \times 2^{\mathcal{O}\left(2^{(l-j)}\frac{\ln j}{j}\right)}$ bipolar matrices with $\mu \leq \frac{2^{l-j}-1}{2^l-1}$. Although, the use of BCH codes in compressed sensing and dimensionality reduction has been already investigated (e.g., [18], [19]), the approach and results in [12] which are generalized in this paper, are different in that, there is no randomness involved, neither in the matrix nor in the type of recovery guarantees.

TABLE I
DETAILS OF THE CONSTRUCTED $m \times n$ MATRICES IN THIS PAPER. μ_{ub} DENOTES THE PROVEN UPPER BOUND ON THE COHERENCE WHILE k_{gr} IS THE SPARSITY ORDER UP TO WHICH THE PERFECT RECONSTRUCTION IS GUARANTEED.

m	$p^l - 1$
n	$p^{\mathcal{O}\left(p^{(l-r)}\frac{\log_p r}{r}\right)}$
μ_{ub}	$\frac{p}{2(p-1)}\frac{p^{l-r}-1}{p^l-1}$
k_{gr}	$\left\lfloor \frac{p-1}{p}\frac{p^l-1}{p^{l-r}-1} + 0.5 \right\rfloor$
elements	$\frac{e^{j\frac{2\pi}{p}a}}{\sqrt{m}}$ for $a \in \{0, 1, \ldots, p-1\}$
constraints	$1 < l \in \mathbb{N}$, $1 \leq r \leq l-1$ and p prime
Inequality form	$m \leq \mathcal{O}\left(k_{gr}\left(\log_p n\right)^{\frac{\log_p k_{gr}}{\log_p \log_p k_{gr}}}\right)$

Unlike the Devore's matrices for which the coherence is lower bounded by Johnson's bound (see [20] or [12] for the explanation of the bound) rather than the Welch bound, the coherence of the BCH-based matrices in [12] is relatively close to the Welch bound. However, the number of rows (m) in these matrices are restricted to the forms $2^l - 1$ (there are more options in Devore's design). In this paper, we generalize the utilization of the binary BCH codes to the use of p-ary codes (where p is a prime integer) and obtain $m \times n$ complex-valued sensing matrices; the details of these matrices are shown in Table I (the matrices in [12] are special cases when $p = 2$). This generalization, not only increases the possible options for m, but also results in matrices with a coherence closer to the Welch bound as p increases. We further broaden the achievable range of options by introducing two techniques for combining matrices with small coherence. The first is the Kronecker product which is of special interest for changing the number of rows. In the second method, we combine a binary matrix with fixed column weight and another matrix with fixed absolute value of the elements. This technique increases the number of columns in the binary matrix without increasing the number of rows or the coherence.

The rest of the paper is organized as follows: in Sec. II we explain how block codes, specially p-ary codes, can form sensing matrices with small coherence. Here we briefly review the concepts of the binary design in [12] and highlight the challenges for generalizing to p-ary codes. Section III describes a p-ary code design suitable for generating sensing matrices. The method is completely deterministic (no search is required) and is based on the generalized BCH codes. Due to the use of p-ary BCH codes, the number of rows in these matrices are restricted to the form $p^l - 1$ for some integer l; in Sec. IV, we show that by using the Kronecker product of these matrices, we can achieve matrices with more options on the number of rows. Other than the Kronecker product, we present a technique for combining binary and p-ary matrices which increases the number of columns without changing the number of rows. The simulation results in Sec. V confirm that the BCH-based matrices perform almost similar to the random matrices; here we consider different scenarios including real images. Finally, Sec. VI concludes the paper.

II. Complex Matrices via p-ary Linear Codes

In this section, we explain how block codes, specially p-ary codes, can form sensing matrices with small coherence.

[1] In the case of StRIP, for a given and fixed matrix, the inequalities in (1) hold with high probability if the support of the k-sparse vector is drawn uniformly at random from all the $\binom{n}{k}$ possible ways and the non-zero elements follow an independent and identical Gaussian distribution.

Since the approach is based on the one used for bipolar matrices introduced in [12], we briefly discuss the binary design concepts.

Assume that we are given a $(\tilde{n},\tilde{k})$ linear binary code[2] with the minimum distance $\tilde{d}_{min}$ such that the all-one vector ($\mathbf{1}_{\tilde{n}\times 1}$) is a valid codeword; due to the linearity of the code, all-zero vector ($\mathbf{0}_{\tilde{n}\times 1}$) is always a codeword. 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:

$$\begin{cases} d(\mathbf{c}_{\tilde{n}\times 1},\mathbf{0}_{\tilde{n}\times 1})\geq\tilde{d}_{min} \\ d(\mathbf{c}_{\tilde{n}\times 1},\mathbf{1}_{\tilde{n}\times 1})\geq\tilde{d}_{min} \end{cases}, \tag{4}$$

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. 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 codeword $\mathbf{a}$, the first case happens only when $\mathbf{b}=\mathbf{a}$ or $\mathbf{a}\oplus\mathbf{1}_{\tilde{n}\times 1}$; thus, all the possible $2^{\tilde{k}}$ codewords 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 exactly one vector from each set and putting them as the columns, and then converting 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 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 upper bounded by $\tilde{n}-2\tilde{d}_{min}$. Hence, the coherence of the matrix $\mathbf{A}$ when the columns are normalized by the factor $\frac{1}{\sqrt{\tilde{n}}}$ (all the columns have the same norm and thus, normalization is equivalent to scaling), is upper bounded by $\frac{\tilde{n}-2\tilde{d}_{min}}{\tilde{n}}$. Recalling a result from [6], [12], we know that it is possible to perfectly recover a k-sparse vector from noiseless measurements obtained by a sensing matrix with a coherence less than $\frac{1}{2k-1}$. Thus, the mentioned matrix $\mathbf{A}$ is guaranteed to recover k-sparse vectors for $k\leq\frac{\tilde{n}}{2(\tilde{n}-2\tilde{d}_{min})}+0.5$.

To generalize the above results to p-ary 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 codewords and unlike the binary case, does not give useful information about the differences 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 by -1 in the binary case. To solve the latter, we introduce complex matrices by converting the code elements into points on the unit circle in the complex plane while for the first challenge, 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 $\mathbf{0}_{\tilde{n}\times 1},\mathbf{1}_{\tilde{n}\times 1},\dots,(\mathbf{p}-\mathbf{1})_{\tilde{n}\times 1}$ are also codewords. 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 holds[3]:

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

$$\begin{cases} d(\mathbf{c}_{\tilde{n}\times 1},\mathbf{0}_{\tilde{n}\times 1})\geq\tilde{d}_{min} \\ d(\mathbf{c}_{\tilde{n}\times 1},\mathbf{1}_{\tilde{n}\times 1})\geq\tilde{d}_{min} \\ \vdots \\ d(\mathbf{c}_{\tilde{n}\times 1},(\mathbf{p}-\mathbf{1})_{\tilde{n}\times 1})\geq\tilde{d}_{min} \end{cases}, \tag{5}$$

which means that $\mathbf{c}_{\tilde{n}\times 1}$ contains at most $\tilde{n}-\tilde{d}_{min}$ from each of $\{0,1,\dots,p-1\}$. 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 $\sum_{i=0}^{p-1}N_i=\tilde{n}$ result in:

$$N_i=\tilde{n}-\sum_{j\neq i}N_j\geq\tilde{n}-(p-1)(\tilde{n}-\tilde{d}_{min}). \tag{6}$$

Hence

$$\underbrace{\tilde{n}-(p-1)(\tilde{n}-\tilde{d}_{min})}_{N_{min}}\leq N_i\leq\underbrace{\tilde{n}-\tilde{d}_{min}}_{N_{max}}, \tag{7}$$

which is equivalent to

$$\big|N_i-\frac{N_{min}+N_{max}}{2}\big|\leq\frac{N_{max}-N_{min}}{2}. \tag{8}$$

Similarly, we divide the set of code vectors into subsets of the form $\{\mathbf{a},\mathbf{a}\oplus\mathbf{1}_{\tilde{n}\times 1},\dots,\mathbf{a}\oplus(\mathbf{p}-\mathbf{1})_{\tilde{n}\times 1}\}$ and pick exactly one vector from each subset. In fact, we are looking for the representatives of the elements of the quotient group formed by dividing the group of all code vectors[4] by its subgroup $\{\mathbf{0}_{\tilde{n}\times 1},\dots,(\mathbf{p}-\mathbf{1})_{\tilde{n}\times 1}\}$. 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 codeword 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 $\{\mathbf{a},\mathbf{a}\oplus\mathbf{1}_{\tilde{n}\times 1},\dots,\mathbf{a}\oplus(\mathbf{p}-\mathbf{1})_{\tilde{n}\times 1}\}$. If we construct $\mathbf{A}_{\tilde{n}\times p^{\tilde{k}-1}}$ from $\tilde{\mathbf{A}}$ according to the following rule:

$$\tilde{\mathbf{A}}=[\tilde{a}_{\alpha\beta}]_{\alpha,\beta}\Rightarrow\mathbf{A}=\frac{1}{\sqrt{\tilde{n}}}\big[e^{j\frac{2\pi}{p}\tilde{a}_{\alpha\beta}}\big]_{\alpha,\beta}, \tag{9}$$

the coherence will be upper bounded by $\frac{p(p-1)\tilde{n}-p^2\tilde{d}_{min}}{2\tilde{n}}$.

Proof. First note that the columns of $\mathbf{A}$ all have unit norm:

$$\|\mathbf{a}_\beta\|=\left\|\frac{1}{\sqrt{\tilde{n}}}[e^{j\frac{2\pi}{p}\tilde{a}_{1,\beta}}\ \dots\ e^{j\frac{2\pi}{p}\tilde{a}_{\tilde{n},\beta}}]^T\right\|=1. \tag{10}$$

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 $\mathbf{c}=\tilde{\mathbf{a}}_\alpha\oplus-\tilde{\mathbf{a}}_\beta$. In

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

[3] For p-ary codes, $\oplus$ is the $mod\ p$ addition (element-wise).

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

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

$$\begin{aligned} |\langle \mathbf{a}_\alpha, \mathbf{a}_\beta \rangle| &= |\mathbf{a}_\beta^H \cdot \mathbf{a}_\alpha| = \frac{1}{\tilde{n}} \Big| \sum_{i=1}^{\tilde{n}} e^{j\frac{2\pi}{p}(\tilde{a}_{i,\alpha} - \tilde{a}_{i,\beta})} \Big| \\ &= \frac{\big|\sum_{i=1}^{\tilde{n}} e^{j\frac{2\pi}{p}c_i}\big|}{\tilde{n}} = \frac{\big|\sum_{i=0}^{p-1} N_i e^{j\frac{2\pi}{p}i}\big|}{\tilde{n}}. \end{aligned} \quad (11)$$

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

$$\Big|\sum_{i=0}^{p-1} N_i e^{j\frac{2\pi}{p}i}\Big| = \Big|\sum_{i=0}^{p-1} (N_i - \gamma) e^{j\frac{2\pi}{p}i}\Big| \leq \sum_{i=0}^{p-1} |N_i - \gamma|, \quad (12)$$

where we used the triangle inequality for the last part. Recalling inequalities (7) and (8) and by setting $\gamma = \frac{N_{min}+N_{max}}{2}$, we get:

$$\begin{aligned} \Big|\sum_{i=0}^{p-1} N_i e^{j\frac{2\pi}{p}i}\Big| &\leq p\, \frac{N_{max} - N_{min}}{2} \\ &= \frac{p(p-1)\tilde{n} - p^2 \tilde{d}_{min}}{2}, \end{aligned} \quad (13)$$

which demonstrates the following upper bound on the coherence of $\mathbf{A}$:

$$|\langle \mathbf{a}_\alpha, \mathbf{a}_\beta \rangle| \leq \frac{p(p-1)\tilde{n} - p^2 \tilde{d}_{min}}{2\tilde{n}}. \quad (14)$$

■

Remark 1: The best choice of γ in (12) which yields the least upper bound for the inner product is the median of the N_i's, not necessarily the mean value used in (13); however, the median is not a fixed value and thus, no deterministic upper bound will be derived.

Remark 2: To guarantee $\mu_{\mathbf{A}} < \frac{1}{2k-1}$ (a sufficient condition for perfect recovery of k-sparse signals [6], [12]) by using the upper bound in Theorem 1, we should have:

$$\frac{\tilde{d}_{min}}{\tilde{n}} > \frac{p-1}{p} - \frac{2}{p^2(2k-1)} \geq \frac{p-1}{p}\left(1 - \frac{4}{kp^2}\right). \quad (15)$$

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 implies that in order to increase the sparsity order k, we need to increase $\tilde{d}_{min}$. The existence and design of such matrices from p-ary 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. The BCH codes are a subclass of linear cyclic codes (sums and circular shifts of the code-words are also valid code-words) where the elements of the code vectors are taken from a finite field, namely $GF(p)$ (p is the field size and should be a prime power), and the length of the code-words are $\tilde{n} = p^{\tilde{m}} - 1$ for some integer $\tilde{m}$. Instead of the vector representation, BCH code-words are usually regarded as polynomials of degree $\tilde{n}-1$ over the Galois field (elements in the vector are considered as the polynomial coefficients). In this way, a p-ary $\tilde{n} \times 1$ vector is a valid codeword if its corresponding polynomial is divisible by a fixed polynomial $g(x) \in GF_P[x]$ referred to as the code generating polynomial. In order to have the cyclic property in the code, it is necessary and sufficient that $g(x)$ is a divisor of $x^{\tilde{n}} - 1$ [21]. Recalling a result from the Galois theory, we know [21]:

$$\prod_{\substack{r \in GF(p^{\tilde{m}}) \\ r \neq 0}} (x - r) = x^{p^{\tilde{m}}-1} - 1. \quad (16)$$

Thus, $g(x)$, which is a divisor of $x^{p^{\tilde{m}}-1} - 1$, should be equal to the product of a subset of $(x - r)$'s for $r \in GF(p^{\tilde{m}})$; 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 $GF(p^{\tilde{m}})$; 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 $\{\alpha^{i_1}, \dots, \alpha^{i_d}\}$ is a subset of the roots of $g(x)$ such that $i_1, \dots, i_d$ form an arithmetic progression, we have $\tilde{d}_{min} \geq d + 1$ [21]; if the vector $[c_1, \dots, c_{\tilde{n}}]^T$ is a nonzero codeword, we should have $g(x) | \sum_{j=1}^{\tilde{n}} c_j x^{j-1}$ and therefore[5]:

$$\underbrace{\begin{bmatrix} \alpha^{0\times i_1} & \alpha^{1\times i_1} & \dots & \alpha^{(\tilde{n}-1)\times i_1} \\ \alpha^{0\times i_2} & \alpha^{1\times i_2} & \dots & \alpha^{(\tilde{n}-1)\times i_2} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha^{0\times i_d} & \alpha^{1\times i_d} & \dots & \alpha^{(\tilde{n}-1)\times i_d} \end{bmatrix}}_{\mathbf{H}_{d\times\tilde{n}}} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_{\tilde{n}} \end{bmatrix} = \mathbf{0}_{d\times 1}. \quad (17)$$

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

In our code design approach, we choose $g(x)$ such that the set $\{\alpha^{p^{\tilde{m}-1}+\frac{p^l-1}{p-1}+1}, \alpha^{p^{\tilde{m}-1}+\frac{p^l-1}{p-1}+2}, \dots, \alpha^{p^{\tilde{m}}-2}\}$ is a subset of its roots for an integer $l < \tilde{m}$. Hence, 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

$$\begin{aligned} \tilde{d}_{min} &\geq p^{\tilde{m}} - p^{\tilde{m}-1} - \frac{p^l - 1}{p-1} - 1 \\ &= (p^{\tilde{m}} - 1)\left(1 - \frac{p^{\tilde{m}-1}}{p^{\tilde{m}} - 1} - \frac{p^l - 1}{(p^{\tilde{m}} - 1)(p-1)}\right) \\ &= \tilde{n}\left(\frac{p-1}{p} - \frac{p^{l+1} - 1}{p(p-1)(p^{\tilde{m}} - 1)}\right) \\ \Rightarrow \frac{\tilde{d}_{min}}{\tilde{n}} &\geq \frac{p-1}{p}\left(1 - \frac{p^{l+1} - 1}{(p-1)^2(p^{\tilde{m}} - 1)}\right). \end{aligned} \quad (18)$$

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

[5] $x|y$ implies y is divisible by x.

Except for the trivial cases, the polynomial $\prod_{i=0}^{|\mathcal{T}|-1}(x-\alpha^i)$ does not belong to $GF_p[x]$, which shows that the set of the roots is often a strict subset of $\mathcal{T}$. The following lemma is a helpful tool for identifying the roots of $h(x)$.

***Lemma** 1:* 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 sequence in $\mathcal{H}_{seq}^{(\tilde{m},l)}$. Now, we have:

$$\begin{cases} \mathcal{H}_{\tilde{m}}^{(l)} \subseteq \{0,1,\dots,p^{\tilde{m}-1}+\frac{p^l-1}{p-1}\} \\ \\ \prod_{i\in\mathcal{H}_{\tilde{m}}^{(l)}}(x-\alpha^i) \in GF_p[x] \end{cases} . \tag{19}$$

Proof. Let B be an element of $\mathcal{H}_{seq}^{(\tilde{m},l)}$ with the base-p representation as $(\overline{b_{\tilde{m}-1}\dots b_0})_p$. Since the sequence $(b_{\tilde{m}-1},\dots,b_0)$ is a member of $\mathcal{H}_{seq}^{(\tilde{m},l)}$, each of the b_i's is either 0 or 1. There are two cases:

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

$$\begin{aligned} (\overline{b_{\tilde{m}-1}\dots b_0})_p &\leq (\overline{011\dots 1})_p \\ &= \frac{p^{\tilde{m}-1}}{p-1} \leq p^{\tilde{m}-1}+\frac{p^l-1}{p-1}, \end{aligned} \tag{20}$$

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

$$\begin{aligned} (\overline{b_{\tilde{m}-1}\dots b_0})_p &\leq (\overline{1\underbrace{0\dots 0}_{\tilde{m}-l-1}\underbrace{1\dots 1}_{l}})_p \\ &= p^{\tilde{m}-1}+\frac{p^l-1}{p-1}. \end{aligned} \tag{21}$$

Thus, we have $\mathcal{H}_{\tilde{m}}^{(l)} \subseteq \{0,1,\dots,p^{\tilde{m}-1}+\frac{p^l-1}{p-1}\}$.

In addition, for the same B, we have:

$$\begin{aligned} pB &= (\overline{b_{\tilde{m}-1}\dots b_0 0})_p \\ &= b_{\tilde{m}-1}p^{\tilde{m}}+(\overline{b_{\tilde{m}-2}\dots b_0 0})_p \\ &\equiv b_{\tilde{m}-1}+(\overline{b_{\tilde{m}-2}\dots b_0 0})_p \pmod{p^{\tilde{m}}-1} \\ &\equiv (\overline{b_{\tilde{m}-2}\dots b_0 b_{\tilde{m}-1}})_p \pmod{p^{\tilde{m}}-1}. \end{aligned} \tag{22}$$

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

$$\begin{aligned} &\alpha^{pB}=\alpha^{B^{'}}\in\{\alpha^h\}_{h\in\mathcal{H}_{\tilde{m}}^{(l)}}, \\ \Rightarrow\ &\{\alpha^B,\alpha^{pB},\alpha^{p^2B},\dots,\alpha^{p^{\tilde{m}-1}B}\}\subseteq\{\alpha^h\}_{h\in\mathcal{H}_{\tilde{m}}^{(l)}}. \end{aligned} \tag{23}$$

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

$$\prod_i(x-\alpha^{p^iB})\in GF_p[x], \tag{24}$$

which finally results in $\prod_{i\in\mathcal{H}_{\tilde{m}}^{(l)}}(x-\alpha^i)\in GF_p[x]$. ■

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

$$h(x)\triangleq\prod_{h\in\mathcal{H}_{\tilde{m}}^{(l)}}(x-\alpha^h)\in GF(p)[x]. \tag{25}$$

One of the important conditions to be verified is whether $\mathbf{1}_{\tilde{n}\times 1}$ belongs to the set of codewords. 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

$$\begin{aligned} &\begin{cases} g(x)\big|x^{\tilde{n}}-1=(x-1)(1+x+\dots+x^{\tilde{n}-1}) \\ \\ gcd\big(g(x)\ ,\ x-1\big)=1 \end{cases} \\ &\Rightarrow g(x)\big|1+x+\dots+x^{\tilde{n}-1}, \end{aligned} \tag{26}$$

which confirms that $\mathbf{1}_{\tilde{n}\times 1}$ is a valid codeword. The other issue which should be considered is to choose the representatives from each of the sets $\{\mathbf{a},\mathbf{a}\oplus\mathbf{1}_{\tilde{n}\times 1},\dots\mathbf{a}\oplus(\mathbf{p}-\mathbf{1})_{\tilde{n}\times 1}\}$. Since $p\nmid\tilde{n}$, in each of these sets, the polynomial representation of exactly one of the codewords 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 [12]. Also, the additional factor of $x-1$ increases the lower bound on the minimum distance of the code by 1. Table II summarizes the matrix design steps.

To find the final size of the constructed sensing matrix, we should calculate the value $\tilde{k}$; similar to the discussions in [12], this value is equal to the size of the set $\mathcal{H}_{\tilde{m}}^{(l)}$. It is shown in [12] that $|\mathcal{H}_{\tilde{m}}^{(l)}|=\mathcal{O}\big(\gamma^{l+1}\big)$ where γ is the largest root of $x^{\tilde{m}-l-1}-x-1$. Thus, for the $m\times n$ constructed sensing matrix using this code, we have:

$$\begin{cases} m &= p^{\tilde{m}}-1 \\ \log_p n &= |\mathcal{H}_{\tilde{m}}^{(l)}|=\mathcal{O}\big(\gamma^{l+1}\big) \\ k_{gr} &\geq \frac{1}{2\mu_{\mathbf{A}}}\geq\frac{p-1}{p}\frac{p^{\tilde{m}}-1}{p^{l+1}-1}\approx p^{\tilde{m}-l-1} \end{cases} , \tag{27}$$

where k_{gr} represents the threshold for the sparsity order of the sparse vector, up to which we can guarantee the perfect reconstruction. Using the inequality $\ln\gamma\geq\frac{\ln\left(\tilde{m}-l-1\right)}{\tilde{m}-l-1}$ (see [12]) we can show $\gamma^{\frac{\log_p k_{gr}}{\log_p\log_p k_{gr}}}\geq p$. Therefore, we have:

$$m\leq\mathcal{O}\Bigg(k_{gr}(\log_p n)^{\frac{\log_p k_{gr}}{\log_p\log_p k_{gr}}}\Bigg). \tag{28}$$

Obviously, the upper bound for m in the random matrices ($m\leq\mathcal{O}\big(k_{gr}\log_p n\big)$) is much stronger than what we have proved here; not only is the power of $\log_p n$ greater than one in our case, but also it increases as the desired k_{gr} increases. To the best of our knowledge, no deterministic design is available yet which by means of RIP guarantees a fixed value c such that $m\leq\mathcal{O}\big(k_{gr}(\log n)^c\big)$. Since, the design principle in our matrices is coherence, it is logical to compare the resulting coherence with the Welch bound (3). For this purpose, we have reported the ratio of the achieved coherence to the Welch bound ($\frac{\mu_{BCH}}{\mu_{WB}}$) for some special BCH-based matrices in Table III. We have used the special case of $\tilde{m}=2l$, which results in $(p^{2l}-1)\times p^{3l}$ matrices; in fact, for each p we have $\mathcal{H}_{2l}^{(l)}=\{0\}\cup\{p^i\}_{i=0}^{2l-1}\cup\{p^{l+i}+p^i\}_{i=0}^{l-1}$. The results in Table III show that the achieved coherence tends to the Welch bound as p increases.

IV. Matrix Resizing

In this section, we introduce two methods to change the size of the previously discussed matrices. In the first method,

TABLE II
MATRIX DESIGN STEPS

For a given prime power p,

1) Choose the positive integer $\tilde{m}$ and set $m = p^{\tilde{m}} - 1$.
2) Choose an integer $0 \leq l \leq \tilde{m} - 1$. The value $\frac{p}{2(p-1)} \frac{p^{l+1}-1}{p^{\tilde{m}}-1}$ is an upper bound for the coherence of the final matrix.
3) Form the set $\mathcal{H}_{seq}^{(\tilde{m},l)}$ by finding all binary sequences of length $\tilde{m}$ such that each two 1's are circularly spaced by at least $\tilde{m} - l - 1$ zeros. Let $\mathcal{H}_{\tilde{m}}^{(l)}$ be the decimal numbers for which the base-p representation is a sequence in $\mathcal{H}_{seq}^{(\tilde{m},l)}$.
4) Define:
$$h(x) = \prod_{r \in \mathcal{H}_{\tilde{m}}^{(l)} \setminus \{0\}} (x - \alpha^r),$$
where α is one of the primitive elements of $GF(p^{\tilde{m}})$. Also set $n = p^{|\mathcal{H}_{\tilde{m}}^{(l)}|-1}$.
5) Put all the code vectors of the code defined by $h(x)$ and $g(x) = \frac{x^{p^{\tilde{m}}-1}-1}{h(x)}$ as different columns of the matrix $\tilde{\mathbf{A}}_{m \times n}$.
6) Define the final matrix as
$$\mathbf{A}_{m \times n} = \frac{1}{\sqrt{m}} \left[e^{j 2\pi \frac{\tilde{a}_{i,j}}{p}} \right],$$
where $\tilde{a}_{i,j}$'s are the elements of the matrix $\tilde{\mathbf{A}}$.

TABLE III
$\frac{\mu_{BCH}}{\mu_{WB}}$ FOR VARIOUS BCH-BASED $(p^{2l}-1) \times p^{3l}$ MATRICES.

	$p=2$	$p=3$	$p=5$	$p=7$
$l=1$	—	1.1863	1.1009	1.0709
$l=2$	1.1296	1.0549	1.0198	1.0102
$l=3$	1.0618	1.0184	1.0040	1.0015

by employing the binary matrices, we increase the number of columns (n) for fixed values of m and k; however, this increase does not change the order of $\mathcal{O}(k \log n)$. The second method is to change the number of rows (number of samples) which provides us with more options on the number of samples. A summary of these methods is given in Table IV.

A. Mixing with Binary Matrices

The design of binary matrices with small coherence, due to the non-negative nature of the elements, is a difficult task; the inner product of two binary vectors consists merely of non-negative terms which demonstrates the main difficulty in finding quasi-orthogonal binary vectors in relatively small dimensions. Hopefully, there are at least two known binary matrix structures: 1) Devore's matrices [11] with p^2 rows where p is a prime power and the weight of each column is p and 2) OOC-based matrices introduced in [12] for which the number of rows has slightly larger range of options.

***Lemma** 2:* Given a binary matrix $\mathbf{A}_{m \times n_1}$ whose columns each have w_m nonzero entries with coherence $\mu_{\mathbf{A}}$, and a $w_m \times n_2$ matrix $\mathbf{B}$ with coherence $\mu_{\mathbf{B}}$ for which the elements have the same absolute value, there exists a deterministic construction for an $m \times (n_1 n_2)$ matrix coherence $\mathbf{C}$ with $\mu_{\mathbf{C}} \leq \max\left(\mu_{\mathbf{A}}, \mu_{\mathbf{B}}\right)$ and normalized columns.

TABLE IV
SUMMARY OF THE RESIZING TECHNIQUES.

	Kronecker	Binary-Mixing
Inputs	$\mathbf{A}_{m_a \times n_a}$, $\mathbf{B}_{m_b \times n_b}$	$\mathbf{A}_{m_a \times n_a}$, $\mathbf{B}_{m_b \times n_b}$
Output	$\mathbf{C}_{(m_a m_b) \times (n_a n_b)}$	$\mathbf{C}_{m_a \times (n_a n_b)}$
Coherence	$\mu_{\mathbf{C}} \leq \max(\mu_{\mathbf{A}}, \mu_{\mathbf{B}})$	$\mu_{\mathbf{C}} \leq \max(\mu_{\mathbf{A}}, \mu_{\mathbf{B}})$
Constraints	—	$\mathbf{A}$ is binary with column-weight m_b and elements of $\mathbf{B}$ have similar absolute value.

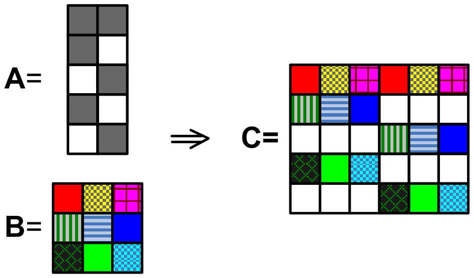


Fig. 1. The procedure of mixing with a binary matrix: $\mathbf{A}$ is a binary matrix with constant column weight and the elements of $\mathbf{B}$ have the same absolute value.

Proof Here, we explicitly construct $\mathbf{C}$ by mixing the two matrices. To form the l^{th} column of $\mathbf{C}$, we first write $l-1$ as $\alpha \cdot n_2 + \beta$, where $\alpha \in \{0, 1, \ldots, n_1 - 1\}$ and $\beta \in \{0, 1, \ldots, n_2 - 1\}$ (in fact, α and β are the quotient and remainder of $l-1$ by n_2, respectively). Let $i_1, \ldots, i_{w_m}$ be the indices of the nonzero locations in the $(\alpha+1)^{th}$ column of the matrix $\mathbf{A}$. Now the elements of the l^{th} column in $\mathbf{C}$ are:

$$\left\{ \begin{array}{lcl} c_{i_1,l} & = & b_{1,\beta+1} / \left(r_{\mathbf{B}} \sqrt{w_m} \right) \\ c_{i_2,l} & = & b_{2,\beta+1} / \left(r_{\mathbf{B}} \sqrt{w_m} \right) \\ & \vdots & \\ c_{i_{w_m},l} & = & b_{w_m,\beta+1} / \left(r_{\mathbf{B}} \sqrt{w_m} \right) \\ c_{s,l} & = & 0 \quad , \qquad s \notin \{i_1, \ldots, i_{w_m}\} \end{array} \right. , \tag{29}$$

where $[b_{1,\beta+1}, \ldots, b_{w_m,\beta+1}]^T$ is the $(\beta+1)^{th}$ column of $\mathbf{B}$ and $r_{\mathbf{B}}$ is the absolute value of the elements of $\mathbf{B}$. The schematic diagram of the above procedure is shown in Fig. 1.

To show the coherence property of $\mathbf{C}$, let $\mathbf{u}_{m \times 1}$ and $\mathbf{v}_{m \times 1}$ be the l_1^{th} and l_2^{th} columns of $\mathbf{C}$, respectively, where $l_1 - 1 = \alpha_1 \cdot n_2 + \beta_1$ and $l_2 - 1 = \alpha_2 \cdot n_2 + \beta_2$. It is trivial to check that $\|\mathbf{u}\| = \|\mathbf{v}\| = 1$. To investigate the inner product of the two vectors, we consider the following two cases:

1) $\alpha_1 \neq \alpha_2$, which means that $\mathbf{u}, \mathbf{v}$ are generated using different columns of $\mathbf{A}$ and therefore, they have different patterns of nonzero elements. Since the inner product of different columns of the binary matrix $\mathbf{A}$ are less than $\mu_{\mathbf{A}}$ ($w_m \mu_{\mathbf{A}}$ prior to column normalization), at most $w_m \mu_{\mathbf{A}}$ of the nonzero elements of $\mathbf{u}$ coincide with that of $\mathbf{v}$. Moreover, the absolute value of the nonzero elements in both $\mathbf{u}$ and $\mathbf{v}$ is $\frac{1}{\sqrt{w_m}}$ (normalized elements

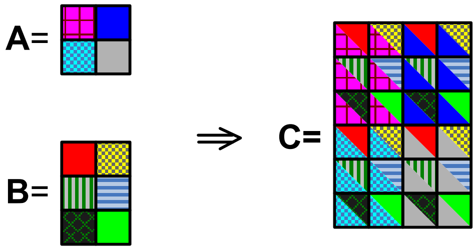


Fig. 2. Kronecker product of two matrices ($\mathbf{C} = \mathbf{A} \otimes \mathbf{B}$).

of $\mathbf{B}$). Consequently, we have:

$$\begin{aligned} |\langle \mathbf{u},\mathbf{v}\rangle| &= \Big|\sum_{i=1}^{n} u_i v_i\Big| \\ &\le \sum_{i=1}^{n} |u_i v_i| = w_m \mu_{\mathbf{A}} \left(\frac{1}{\sqrt{w_m}}\right)^2 \\ &= \mu_{\mathbf{A}}, \end{aligned} \tag{30}$$

2) $\alpha_1 = \alpha_2$, which means that $\mathbf{u}, \mathbf{v}$ are generated using the same column of $\mathbf{A}$ and therefore, their inner product is the same as the inner product of the respective columns in $\mathbf{B}$ (columns $\beta_1 + 1$ and $\beta_2 + 1$):

$$|\langle \mathbf{u},\mathbf{v}\rangle| = \frac{|\langle \mathbf{b}_{\beta_1+1}, \mathbf{b}_{\beta_2+1}\rangle|}{r_{\mathbf{B}}^2 w_m} \le \mu_{\mathbf{B}}. \tag{31}$$

Thus, $\mathbf{C}$ has normalized columns and its coherence is upper bounded by $\max\big(\mu_{\mathbf{A}}, \mu_{\mathbf{B}}\big)$. ■

Although this technique increases $n = n_1 n_2$ (dimension) for the same values of m (number of measurements) and μ (and consequently, k_{gr}), the order of $\log n$ and consequently $\frac{\log n}{m}$ is not improved:

$$\begin{aligned} &\log n = \log n_1 + \log n_2 \qquad \Rightarrow \\ &\mathcal{O}\big(\log n\big) = \max\big\{\mathcal{O}\big(\log n_1\big), \mathcal{O}\big(\log n_2\big)\big\}. \end{aligned} \tag{32}$$

B. Kronecker Product

Let $\mathbf{A}_{m_a \times n_a}$ and $\mathbf{B}_{m_b \times n_b}$ be two arbitrary matrices. Define:

$$\mathbf{C}_{m_a m_b \times n_a n_b} \triangleq \mathbf{A}_{m_a \times n_a} \otimes \mathbf{B}_{m_b \times n_b} \tag{33}$$

where $\otimes$ denotes the Kronecker product of the two matrices; i.e.:

$$c_{\eta,\theta} = a_{\gamma,\tau} b_{\rho,\nu}, \tag{34}$$

where $\eta = (\gamma - 1)m_b + \rho$, $\theta = (\tau - 1)n_b + \nu$, and γ, τ, ρ, ν are positive integers not exceeding m_a, n_a, m_b, n_b, respectively. Figure 2 shows the schematic diagram of the above Kronecker product.

***Lemma** 3:* Assume $\mathbf{C} = \mathbf{A} \otimes \mathbf{B}$.

(i) If $\mathbf{A}$ and $\mathbf{B}$ have normalized columns, $\mathbf{C}$ has also normalized columns.

(ii) $\mu_{\mathbf{C}} = \max\big\{\mu_{\mathbf{A}}, \mu_{\mathbf{B}}\big\}$.

(iii) If both $\mathbf{A}$ and $\mathbf{B}$ satisfy RIP of order k with constants $\delta_{k,\mathbf{A}}$ and $\delta_{k,\mathbf{B}}$, respectively, matrix $\mathbf{C}$ also satisfies RIP of order k with $\delta_{k,\mathbf{C}} \le \delta_{k,\mathbf{A}}\delta_{k,\mathbf{B}} + \delta_{k,\mathbf{A}} + \delta_{k,\mathbf{B}}$.

The proof of the first two statements can be found in [22], [23] and for the third, the reader is referred to [24].

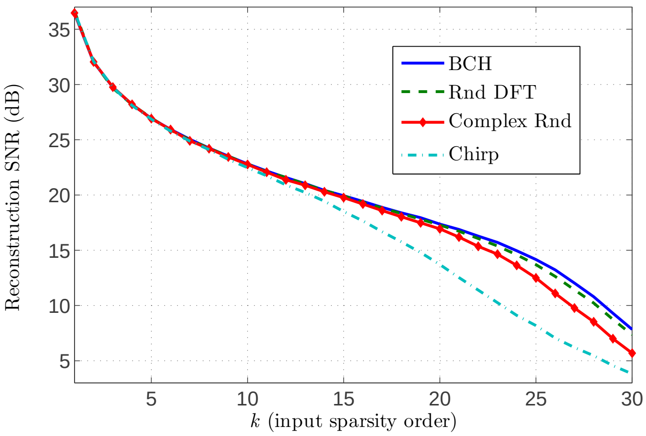


Fig. 3. The reconstruction SNR vs. sparsity order where the noisy compressed samples have SNR of 15 dB. The matrices are 80×729 and the coherence of the 3-ary BCH-based matrix ($p = 3$) is $\frac{1}{8}$.

Now the interesting result which can be achieved by the above operation is the generation of matrices with small coherence and arbitrary number of rows (number of samples). The method in [11] generates matrices with only prime power number of rows; however, using the Kronecker product, we can obtain matrices with number of rows as any product of the prime powers which obviously includes all the positive integers. The disadvantage of this method is the order decrease of $\frac{\log n}{m}$ for a fixed μ or k_{gr}:

$$\begin{aligned} \frac{\log n_c}{m_c} &= \frac{\log n_a + \log n_b}{m_a m_b} \\ &= \frac{1}{m_b}\frac{\log n_a}{m_a} + \frac{1}{m_a}\frac{\log n_b}{m_b}. \end{aligned} \tag{35}$$

Therefore, even if $\mathbf{A}$ and $\mathbf{B}$ are random matrices ($\frac{\log n_a}{m_a}$ and $\frac{\log n_b}{m_b}$ are close to a fixed multiple of k_{gr}^{-1}), the guaranteed performance of $\mathbf{C}$ is much worst than the random matrices; e.g., $\frac{\log n_c}{m_c} \to 0$ for large size values of $\mathbf{A}$ and $\mathbf{B}$, while for random matrices that guarantee the recovery of k_{gr}-sparse vectors with the same length, $\frac{\log n_{rnd}}{m_{rnd}}$ is close to a multiple of k_{gr}^{-1}.

V. Simulation Results

In this section, the performance of our proposed class of sensing matrices based on the p-ary BCH codes is compared to the performance of various types of sampling matrices including those proposed in [17] based on the chirp functions, random rows of the DFT matrix, and the realizations of complex-valued Gaussian random matrices.

In order to have a good performance evaluation for our matrices, we have implemented the matrices for three different cases of p (in the p-ary BCH codes); namely, $p = 3, 5, 7$. Moreover, to have a fair comparison, we have considered the same size, $(p^4 - 1) \times p^6$, for the BCH-based matrices, matrices formed by random DFT rows, complex random (independent real and imaginary parts) and chirp-based matrices (80×729, 624×15625, and 2400×117649 for $p = 3, 5, 7$, respectively).

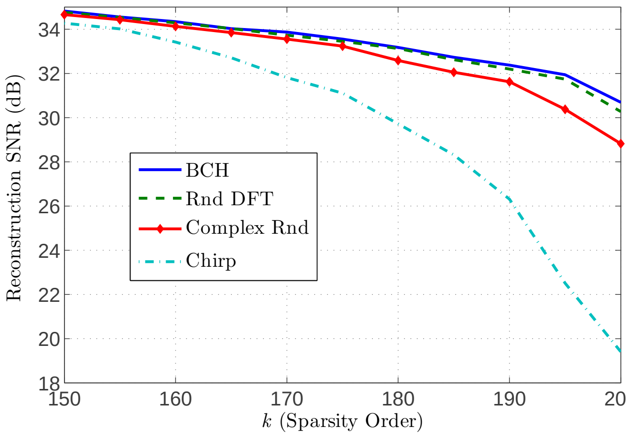


Fig. 4. The reconstruction SNR vs. sparsity order for the noisy compressed samples with the SNR of 30 dB. The matrices are 624 × 15625 and the coherence of the 5-ary BCH-based matrix ($p = 5$) is $\frac{1}{24}$.

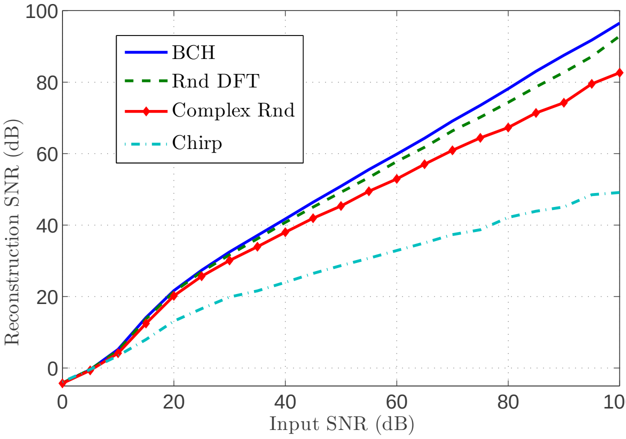


Fig. 5. The reconstruction SNR of a 25-sparse signal from its noisy compressed samples for various input SNRs. The matrices are 80 × 729 and the coherence of the 3-ary BCH-based matrix ($p = 3$) is $\frac{1}{8}$.

In the simulations, the original k-sparse vector is generated by first producing a realization of an $n \times 1$ vector of i.i.d. zero-mean complex-valued Gaussian random elements (independent real and imaginary parts) with $\sigma = 1$ and then setting $n - k$ of its elements to zero; the location of k nonzero elements is chosen uniformly at random among all $\binom{n}{k}$ possibilities. The compressed samples are generated by the mentioned rectangular matrices and then, the samples are subject to Additive White Gaussian Noise (AWGN) with different variances; we refer to the sample to noise power ratio as input SNR. Finally, the original k-sparse vector is reconstructed from the noisy samples using Orthogonal Matching Pursuit (OMP); the SNR of the reconstructed signal with respect to the original vector is referred to as the reconstruction SNR. It is shown in [6], [12] that the family of Matching Pursuit (MP) methods will perfectly recover the original k-sparse vector from the noiseless samples if the coherence of the sampling matrix is less than $\frac{1}{2k-1}$.

To have smooth curves, the results are averaged over 5000 different runs (500 runs for Fig. 4). It is worth mentioning that the coherence of the 80×729, 624×15625 and 2400×117649 BCH-based matrices are $\frac{1}{8}$, $\frac{1}{24}$ and $\frac{1}{48}$, respectively.

Figure 3 demonstrates the SNRs for the reconstruction of k-sparse input signals of size $n = 729$ where k varies from 1 to 30 and the compressed samples are corrupted by AWGN with $SNR = 15\ dB$, while Fig. 4 shows similar curves for input size $n = 15625$, where the sparsity orders $150 \leq k \leq 200$ and input $SNR = 30\ dB$ are considered. These figures show that the BCH-based matrices outperform all other designs, however, the difference between the performance of these matrices and those formed by random rows of the DFT matrix is negligible.

Figure 5 presents the reconstruction SNRs of 25-sparse 729×1 input signals where the compressed samples are subject to varying noise powers resulting in input SNRs ranging from 0 to 100 dB. This figure again confirms that the BCH-based matrices and those formed by random rows of the DFT matrix perform almost equally and better than the rest. Moreover, the obtained curves resemble linear trends between the input and reconstruction SNRs for high enough input SNRs.

In another simulation scenario, we have evaluated the maximum sparsity order of the input signals for which the signal can be recovered almost perfectly from the noiseless compressed measurements. Figures 6 and 7 show the recovery percentage ($SNR_{reconst} \geq 100\ dB$) at different input sparsity orders. The matrices in Fig. 6 are similar to those of Fig. 5 and 3; this figure shows a slight advantage of the BCH-based matrices to their closest competitor, the matrices formed by random rows of the DFT matrix. In Fig. 7 we have evaluated the performance of the matrices formed by mixing techniques in Sec. IV. As a representative of the binary mixing technique discussed in Sec. IV-A, we have combined the 64×512 binary matrix with column weight 8 using Devore's design (field size 8), and the 8×9 complex-valued matrix using ternary BCH codes; the result is a 64×4608 matrix with coherence 0.25. Also, random matrices (complex-valued random matrix and random rows of the DFT matrix) of size 64×4608 are used for this figure. Since the number of columns in the chirp-based matrices can not exceed the square of the number of rows, the size 64×4608 is not realizable in this design; therefore, we have considered the 75×4608 chirp-based matrix with coherence $\frac{1}{\sqrt{3}}$. For the Kronecker product technique discussed in Sec. IV-B, we have combined the 9×27 binary matrix with column weight 3 using Devore's design (field size 3), and the 7×64 bipolar matrix using binary BCH codes; the result is a 63×1728 matrix with coherence $\frac{5}{7}$. Although the binary-mixed matrix produces almost the same number of compressed samples for the inputs of 2.5 times larger than the Kronecker-mixed matrix, its performance which is close to the random matrices, is completely superior than the Kronecker-mixed one. Regarding the largest k value for which the recovery percentage is almost one, this matrix outperforms the chirp-based one as well.

To justify our claim regarding the fast implementation of the recovery algorithm when BCH-based matrices are used, we have compared the required time for recovering the sparse

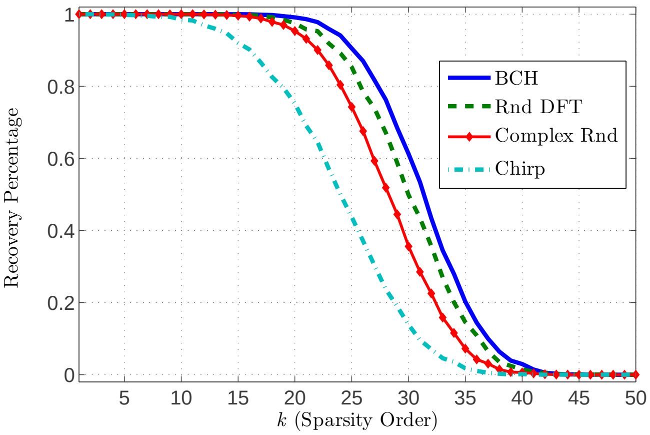


Fig. 6. The recovery percentage vs. sparsity order where the compressed samples are noiseless. The matrices are 80×729 and the coherence of the 3-ary BCH-based matrix ($p = 3$) is $\frac{1}{8}$.

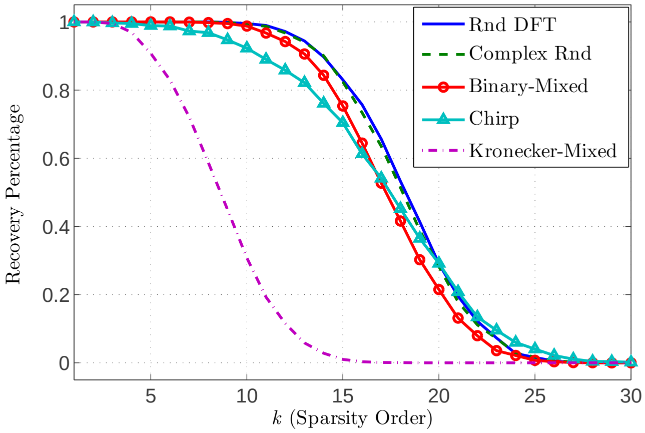


Fig. 7. The recovery percentage vs. sparsity order where the compressed samples are noiseless. The coherence of the binary-mixed (64×4608), Kronecker-mixed (63×1728) and chirp-based matrices (75×4608) are $\frac{1}{4}$, $\frac{5}{7}$ and $\frac{1}{\sqrt{3}}$,respectively. The random matrices are of size 64×4608

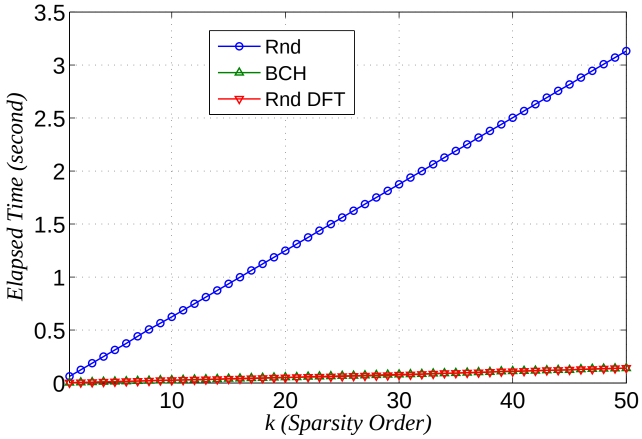
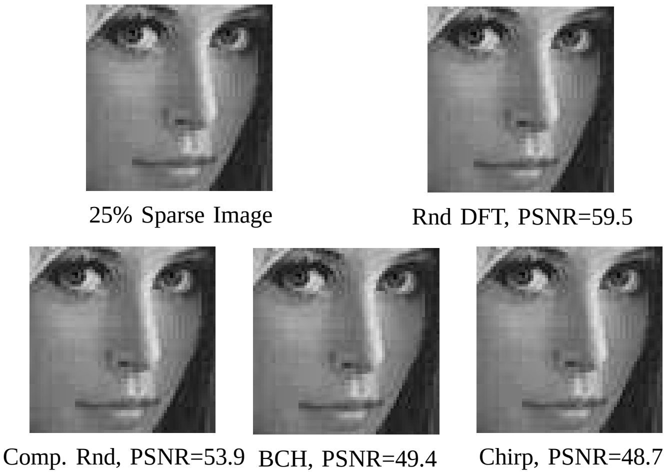


Fig. 8. Comparison of the required time for retrieving sparse 15625×1 signals utilizing a 5-ary BCH-based matrix, complex-valued random matrices and random sub-matrices of the DFT matrix all of size 624×15625 vs. sparsity order of the source signal.


25% Sparse Image

Rnd DFT, PSNR=59.5

Comp. Rnd, PSNR=53.9

BCH, PSNR=49.4

Chirp, PSNR=48.7

Fig. 9. Comparison of utilizing different sampling matrices in order to compress a 65×65 Lena image with sparsity of 25%. All the employed matrices are 2400×4225; from the 7-ary 2400×117649 BCH-based matrix which has the coherence $\frac{1}{48}$, we have kept only the first 4225 columns.

vectors with and without employing the circular characteristic of the columns in Fig. 8. This circular characteristic enables us to employ the FFT algorithm (m-point FFT) for finding the cross correlation of the samples' vector and the columns of the sensing matrix; for further details see [12]. Also, sub-matrices of the DFT matrix, although do not have the circular property in their columns, are special in that the required correlations can be found by a single n-point FFT operation. In Fig. 8, we have utilized the circular property of the BCH-based matrices and compared the required time for reconstructing the source signal to the reconstruction time when complex random matrices and sub-matrices of the DFT matrix are utilized; the results of this figure are obtained by considering 15625×1 original k-sparse vectors for $1 \leq k \leq 50$. The curves for the BCH-based matrix and sub-matrices of the DFT matrix almost coincide while the curve for the simple OMP method indicates a higher order of computational complexity. These curves reveal that for $k = 45$, the FFT-assisted methods are approximately 16 times faster than the simple one, which is remarkable.

In our last simulation, we use the Lena image of size 65×65. The original signal has been made sparse ($\frac{k}{n} = 0.25$) using Haar wavelet coefficients (discarding %75 of the coefficients). Figure 9 depicts the reconstructed images and their Peak Signal to Noise Ratios (PSNR) with respect to the sparse image; in this scenario, random matrices outperform the deterministic designs while 7-ary BCH-based matrix marginally outperform the chirp-based matrix.

VI. Conclusion

A new design for matrices with small coherence is investigated which results in complex-valued matrices (except for the special case of $p = 2$). The design is based on the previously studied link between coding theory and compressed sensing. The considered codes are generalized p-ary BCH codes that provide large minimum distances among the code vectors. The case of $p = 2$ (special case of bipolar matrices) was previously investigated. Simulation results confirm that the performance of the new matrices has reached the bounds of complex random compressed sensing while they outperform the chirp-type matrices. In addition, we have studied two mixing techniques for combining matrices with small coherence. More specifically, the Kronecker product is considered as a tool for generating sensing matrices with desirable number of rows while the other technique can increase the achievable number of columns.

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] 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.

[5] E. Candes and T. Tao, “Decoding by linear programming,” *IEEE Trans. Inform. Theory*, vol. 51, no. 12, pp. 4203–4215, Dec. 2005.

[6] J. Tropp, “Greed is good: algorithmic results for sparse approximation,” *IEEE Trans. on Inform. Theory*, vol. 50, no. 10, pp. 2231–2242, Oct. 2004.

[7] W. Xu and B. Hassibi, “Efficient compressive sensing with deterministic guarantees using expander graphs,” in *International Symposium on Information Theory (ISIT)*, 2007.

[8] F. Parvaresh and B. Hassibi, “Explicit measurements with almost optimal thresholds for compressed sensing,” in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2008.

[9] S. Jagabathula and D. Shah, “Inferring rankings under constrained sensing,” in *Neural Information Processing Systems (NIPS)*, 2008.

[10] X. Jiang, Y. Yao, and L. Guibas, “Stable identification of cliques with restricted sensing,” in *Neural Information Processing Systems (NIPS)*, 2009.

[11] 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.

[12] A. Amini and F. Marvasti, “Deterministic construction of binary, bipolar and ternary compressed sensing matrices,” *IEEE Trans. Inform. Theory*, vol. 57, no. 4, pp. 2360–2370, April 2011.

[13] T. Strohmer and R. W. Heath, “Grassmannian frames with applications to coding and communication,” *Applied and Computational Harmonic Analysis*, vol. 14, no. 3, pp. 257–275, May 2003.

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

[15] R. Berinde, A. C. Gilbert, P. Indyk, H. Karloff, and M. J. Strauss, “Combining geometry and combinatorics: A unified approach to sparse signal recovery,” in *Allerton Conference on Communication, Control, and Computing*, pp. 798–805.

[16] 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.

[17] L. Applebaum, S. D. Howard, S. Searle, 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.

[18] R. Calderbank, S. Howard, and S. Jafarpour, “Construction of a large class of deterministic sensing matrices that satisfy a statistical isometry property,” *IEEE Journal of Selected Topics in Sig. Proc.*, vol. 4, no. 2, pp. 358–374, April 2010.

[19] N. Ailon and E. Liberty, “Fast dimension reduction using rademacher series on dual bch codes,” in *ACM-SIAM symp. on Discrete Algorithms (SODA)*, 2008, pp. 215–224.

[20] S. M. Johnson, “A new upper bound for error-correcting codes,” *IRE Trans. Inform. Theory*, vol. 8, pp. 203–207, 1962.

[21] S. Lin and D. J. Costello, *Error Control Coding: Fundamentals and Applications*, 2nd ed. Prentice Hall: Englewood Cliffs, 2004.

[22] Y. Rivenson and A. Stern, “Compressed imaging with separable sensing operator,” *IEEE Sig. Proc. Letters*, vol. 16, no. 6, pp. 449–452, June 2009.

[23] S. Jokar and V. Mehrmann, “Sparse solutions to under-determined Kronecker product systems,” *Linear Algebra and its Applications*, vol. 431, no. 12, pp. 2437–2447, Dec. 2009.

[24] M. F. Duarte and R. G. Baraniuk, “Kronecker product matrices for compressive sensing,” in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2010.