ISIT'2013 1569719629

Quasi-Cyclic Regenerating Codes for Distributed Storage: Existence and Near-MSR Examples

Vignesh G and Andrew Thangaraj
Department of Electrical Engineering
Indian Institute of Technology Madras, Chennai, India
Email: andrew@ee.iitm.ac.in

***Abstract*—Regenerating codes for distributed storage systems promise significant improvements in the cost and maintenance requirements of large-scale data centers. Research in this area continues to define important new parameters and requirements that have the biggest impact in practice. One of the simplest requirements for a regenerating code is the so-called MSR property, which minimizes the number of bits downloaded during repair. Quasi-cyclic MSR codes are of particular interest, mainly for reducing the encoding and decoding complexity. However, quasi-cyclic MSR codes have not been studied in detail in the existing literature. In this work, we prove the negative result that quasi-cyclic MSR codes with no symbol extension do not exist if the number of systematic nodes is greater than or equal to 4. We provide several examples of quasi-cyclic near-MSR codes, which could be useful for reducing implementation complexity. We point out some interesting connections between zeros of quasi-cyclic codes and the MSR requirement, which are useful in the study of quasi-cyclic regenerating codes with symbol extension.**

I. Introduction

In a distributed storage system, bits of a single file are coded for error protection, split into several parts and each part is stored in a separate node or storage device. Suppose that there are a total of n nodes storing b bits each, and let k of them be systematic nodes. The coding is mainly to protect against node failures. In large distributed storage systems, failure of a single node is typical. Upon failure, a new node needs to be installed with the same data as the failed node - a process which is termed exact repair. To obtain the data, the new node connects to the surviving $n-1$ nodes and downloads some bits. The number of bits downloaded by the new node is a measure of the cost needed for repair. An upper bound for this cost is the size of the file equal to kb. However, by careful code design, the number of downloaded bits can be made as low as $(n-1)b/(n-k)$. Codes that aim to reduce the download cost for repair are termed regenerating codes.

The area of regenerating codes for distributed storage was introduced in [1]. In the past few years, there has been very active research in this area, as summarized in [2]. Important code designs, methods and bounds were first presented and explored in [3]. Code constructions using array codes are presented in [4]. The connection to linear algebraic "interference alignment" is particularly interesting, and has been explored further in [5]. Other algebraic code constructions include [6] and [7].

If the (n,k) regenerating code is over the finite field GF(2^m), then each node stores $\alpha = b/m$ symbols from GF(2^m). Typically, b is chosen such that α is an integer multiple of $n-k$, and $\alpha/(n-k)$ is termed the degree of symbol extension. A code that can achieve the lower bound of $(n-1)\alpha/(n-k)$ symbols for regeneration is termed a MSR code. The code is said to have no symbol extension if $\alpha = n-k$. As can be seen, the simplest regenerating codes do not have symbol extension. However, as shown in [3], the range of n and k for which MSR codes exist with no symbol extension is very limited. The constructions in [4] [5] produce MSR codes using $\alpha = O((n-k)^k)$ resulting in an exponential degree of symbol extension and very high complexity.

Code constructions for rate $k/n > 1/2$ are known to be particularly hard with existing solutions, except for a few cases, needing high complexity in terms of a large b or a large finite field alphabet. Since most of the known code constructions do not have cyclic structures, decoding complexity can be higher than that of standard Reed-Solomon codes. Cyclic constructions, which have a potential for reducing encoding and decoding complexity, have not received much attention, except for [7], [8]. In [7], quasi-cyclic regenerating codes for the case $n = 2k$ with the new node connecting to $k+1$ of the remaining nodes was considered. Code constructions were provided for some values of n and k, and a general existence result was proved.

In this work, we are concerned with the existence and possible constructions of quasi-cyclic regenerating codes. Our main result is that quasi-cyclic MSR codes with no symbol extension do not exist for $k \geq 4$. To prove this result, we use a parity-check matrix description for regenerating codes, and impose the requirements of quasi-cyclic structure [9]. The proof, though elementary, comprises several steps, and results from a careful juxtaposition of linear-algebraic alignment properties needed for MSR codes and the algebraic quasi-cyclic property.

We provide some examples of quasi-cyclic regenerating codes that are close to MSR, i.e., number of downloaded bits is close to the lower bound. In some of these cases, we consider symbol extension of small degree. Though these codes are not strictly MSR, they are close in terms of number of downloaded bits, and their encoding/decoding complexity is the same as that of comparable Reed-Solomon codes. Finally, we make some initial observations about quasi-cyclic MSR codes with symbol extension.

In comparison with prior work in this area, the novel aspects

are the use of the parity-check matrix description, which results in some significant simplifications. The results and construction examples for quasi-cyclic MSR codes are new to the best of our knowledge, and have been presented for the first time here.

II. System Model

We consider a distributed storage system, where a K-bit message is encoded into a N-bit codeword and stored in $n = \frac{N}{b}$ nodes with each node storing b bits. The code is constructed such that a *data collector*, interested in accessing the message, will be able to recover the message by connecting to any $k = \frac{K}{b}$ out of the n nodes, downloading $kb = K$ bits, and running a decoding algorithm. We will let $b = \alpha m$ (for some positive integers α and m), and view the bits stored in each node as a length-α vector over GF(2^m). We stick to characteristic-2 fields, though similar ideas extend to other fields. The vector stored in node i is denoted $c_i = [c_{i,1}\ c_{i,2}\ ...\ c_{i,\alpha}]$, $1 \leq i \leq n$ with coordinates $c_{i,j} \in$ GF(2^m). A codeword distributed over n nodes is denoted $c = [c_1\ c_2\ ...\ c_n]$ in the node-wise form. The set of all such codewords is denoted as the code C. The code C, when considered over the alphabet $A =$ GF$(2^m)^\alpha$, has block-length n and message-length $k = \frac{K}{b}$. For the data collector to be successful, the code C needs to be MDS over A. In this work, we will further assume that C is cyclic over A, i.e., if $c = [c_1\ c_2\ ...\ c_n] \in C$, then $[c_2\ c_3\ ...\ c_n\ c_1] \in C$. This will, as expected, require that $n|(2^m - 1)$. We will set $n = 2^m - 1$ in most examples.

When considered over the alphabet GF(2^m), the code C has block-length nm and message-length km. In this alphabet, the code C need not be MDS, but we will suppose that C is linear over GF(2^m). Now, since C is cyclic over A, we see that C is α-quasi-cyclic over GF(2^m), i.e., C is closed under a cyclic shift by α positions. Following the standard convention in the study of quasi-cyclic codes (see [9] and references thereon), a codeword $c = [c_1\ c_2\ ...\ c_n] \in C$ can be thought of as a concatenation of α vectors of length-n $\mathbf{c}_i = [c_{1,i}\ c_{2,i}\ ...\ c_{n,i}]$ for $i = 1, 2, ..., \alpha$. We will use the notation $c = [\mathbf{c}_1|\mathbf{c}_2|...|\mathbf{c}_\alpha]$ to denote this concatenation. Note that each vector $\mathbf{c}_j$ is stored over n nodes with one symbol $c_{i,j}$ stored in node i.

Using the structure results for quasi-cyclic codes from [9], C over GF(2^m) with codewords in the concatenated form $c = [\mathbf{c}_1|\mathbf{c}_2|...|\mathbf{c}_\alpha]$ has a parity-check matrix H of size $(n-k)\alpha \times n\alpha$ composed of block sub-matrices H_{ij}, for $1 \leq i, j \leq \alpha$. Each H_{ij} is circulant, in the sense that row r is a cyclic right shift by 1 of row $r-1$ for $r = 2, 3, \ldots$. The matrices H_{ii} are $(n-k) \times n$ parity check matrices of cyclic MDS codes over GF(2^m). The matrices H_{ij} are all-zero when $i < j$. However, H_{ij} can be non-zero for $i > j$. There is another additional constraint imposed upon these off-diagonal matrices. Considering $h_{ii}(x)$ to be the generator polynomial of the cyclic code with generator matrix H_{ii}, common roots of $h_{ii}(x)$ and $h_{jj}(x)$ must necessarily be roots of $h_{ij}(x)$ [8], [9].

We briefly describe regeneration in terms of the parity-check matrix, since it is non-standard in this area. In this work, we restrict ourselves to regenerating node n by accessing all remaining $n-1$ nodes. Since the code is cyclic over A, regeneration of any other node follows by a cyclic shift. For regenerating node n, we require α codewords from the dual code of C (over GF(2^m)), or the row-space of H, with some specific properties [8]. Let M be an $\alpha \times \alpha(n-k)$ matrix such that the rows of the product MH are, precisely, these α dual codewords. Denoting the i-th column of H as $H(i)$, the i-th column of MH is $MH(i)$. We form an $\alpha \times \alpha$ matrix M_i, $1 \leq i \leq n$, as

$$M_i = [MH(i)\ MH(i+n) \cdots MH(i+(\alpha-1)n)].$$

Note that for a codeword in the node-wise form $c = [c_1\ c_2\ ...\ c_n]$, we have $\sum_i M_i(c_i)^T = 0$.

For regeneration, we need M such that rank$(M_n) = \alpha$, i.e., M_n is invertible. The number of symbols over GF(2^m) that node i needs to send to node n for regeneration is precisely rank(M_i).

The code C is said have no symbol extension if $\alpha = n - k$. The code C is said to be Minimum Storage Regenerating (MSR) if there exists M such that rank$(M_i) = \alpha/(n-k)$ for $1 \leq i \leq n-1$ and rank$(M_n) = \alpha$. In particular, the MSR condition with no symbol extension ($\alpha = n - k$) requires M such that rank$(M_i) = 1$ for $1 \leq i \leq n-1$, and rank$(M_n) = n - k$.

In Sections III and IV, we prove the main result of this work. Since the proof involves several intertwined steps, we first provide a proof for the specific case of $n = 7$, $k = 4$ for the sake of clarity of exposition. This is followed by a generalization, which is brief.

III. Non-existence of (7,4) quasi-cyclic MSR code for $\alpha = 3$

For $n = 7$, $k = 4$ and $\alpha = n - k = 3$ (no symbol extension), a linear MSR code is known to exist [3]. We show, in this section, that a quasi-cyclic MSR code does not exist for the same parameters. The proof is by contradiction, and assumes characteristic-2 fields for simplicity. The same proof extends to other characteristics readily. So, we assume that there exists a $(7, 4)$ quasi-cyclic MSR distributed storage code C with a 9×21 parity-check matrix

$$H = \begin{bmatrix} H_{11} & \mathbf{0}_{3\times 7} & \mathbf{0}_{3\times 7} \\ H_{21} & H_{22} & \mathbf{0}_{3\times 7} \\ H_{31} & H_{32} & H_{33} \end{bmatrix}$$

Further, there exists a 3×9 matrix M for regeneration such that rank$(M_i) = 1$, $1 \leq i \leq 6$, and rank$(M_7) = 3$.

We know from the previous section that the 3×3 regenerative matrices M_i can be written as

$$M_i = [MH(i)\ MH(i+7)\ MH(i+14)].$$

For $1 \leq i \leq 6$, since rank$(M_i) = 1$, we have $\dim(N(M_i)) = 2$, where $N(\cdot)$ denotes the right nullspace for a matrix. Let $a_i = [a_{i1}\ a_{i2}\ a_{i3}]$ and $b_i = [b_{i1}\ b_{i2}\ b_{i3}]$ be a basis for $N(M_i)$. We see that

$$M_i(a_i)^T = M(a_{i1}H(i) + a_{i2}H(i+7) + a_{i3}H(i+14))^T = \mathbf{0}_{3\times 1},$$

or

$$\mathbf{a}_i = (a_{i1}H(i) + a_{i2}H(i+7) + a_{i3}H(i+14)) \in N(M).$$

Similarly,

$$\mathbf{b}_i = (b_{i1}H(i) + b_{i2}H(i+7) + b_{i3}H(i+14)) \in N(M).$$

Now, since the code C is MDS and has minimum distance 4 over $GF(2^m)^3$, we have that the set of columns

$$\{H(i), H(i+7), H(i+14) : i \in S\}$$

are linearly independent for any three-element subset $S \subset [1:7]$, where $[i:j]$ denotes the integer set $\{i, i+1, \ldots, j\}$. Further, since rank$(M_7) = 3$, we have that rank$(M) = 3$, and rank$(N(M)) = 6$. So, we have

Lemma *1:* For $S \subset [1:6]$ with $|S| = 3$, the set

$$B_S = \{\mathbf{a}_i, \mathbf{b}_i : i \in S\}$$

is a basis for $N(M)$.

The next lemma further clears up the structure of B_S.

Lemma *2:* For $1 \leq i \leq 6$, either $a_{i1} \neq 0$, or $b_{i1} \neq 0$.

Proof: We will prove, by contradiction, for $i = 1$. The proof for other cases is similar. The main idea used is that H_{ii} are 3×7 parity-check matrices of MDS codes. So, (1) any three of their columns are independent, and (2) any one of their columns can be written as a linear combination of three other columns.

Suppose $a_{11} = b_{11} = 0$. Writing $\mathbf{a}_4 \in N(M)$ in the basis $B_{\{1,2,3\}}$, and restricting to the first three positions, we have

$$a_{41}[H(4)]_{1:3} = \eta[H(2)]_{1:3} + \kappa[H(3)]_{1:3},$$

where η, κ are constants occurring in the linear combination, and an obvious notation has been used for the restriction. From the above, since the (7,4) code with parity-check matrix H_{11} is MDS, we have $a_{41} = 0$. Similarly, writing $\mathbf{b}_4$ in the basis $B_{\{1,2,3\}}$ and $\mathbf{a}_5, \mathbf{b}_5$ in $B_{\{1,2,3\}}$, we can show that $b_{41} = a_{51} = b_{51} = 0$. Now, using the basis $B_{\{1,4,5\}}$, we get that $a_{i1} = b_{i1} = 0$ for $1 \leq i \leq 6$. So, without loss of generality, we can set $a_i = [0\ 1\ 0]$ and $b_i = [0\ 0\ 1]$. Therefore, $H(15), H(16), H(17) \in N(M)$, which implies that $H(21) \in N(M)$, because $H(21)$ is a linear combination of $H(15), H(16), H(17)$.

Now, $H(21) \in N(M)$ contradicts

$$\text{rank}(M_7) = \text{rank}([MH(7)\ MH(14)\ MH(21)]) = 3,$$

and the proof is complete. ∎

Using Lemma 2, we let, without loss of generality, $a_{i1} = 1$ and $b_{i1} = 0$. With the above choice, we further have $b_{i2} \neq 0$. The proof of this is similar to that of Lemma 2, and we omit the details. So, we can further set, without loss of generality, $b_{i2} = 1$ and $a_{i2} = 0$, and we have, finally,

$$\begin{aligned} \mathbf{a}_i &= H(i) + a_{i3}H(i+14) \in N(M), \\ \mathbf{b}_i &= H(i+7) + b_{i3}H(i+14) \in N(M), \end{aligned}$$

for $1 \leq i \leq 6$. From the structure of $\mathbf{a}_i$ and $\mathbf{b}_i$, it is clear that any $\mathbf{b}_j$, when written as a linear combination of a basis B_S, only involves $\mathbf{b}_i$, $i \in S$. Writing $\mathbf{b}_4$ in the basis $B_{\{1,2,3\}}$ (which is, in fact, in terms of $\mathbf{b}_1$, $\mathbf{b}_2$ and $\mathbf{b}_3$), and restricting to the second three positions, we have

$$[H(11)]_{4:6} = c_1[H(8)]_{4:6} + c_2[H(9)]_{4:6} + c_3[H(10)]_{4:6}. \quad (1)$$

Now, $[H(i+7)]_{4:6}$, $1 \leq i \leq 7$, are the columns of H_{22}, which is a parity-check matrix of a cyclic MDS code. So, (1) becomes

$$H_{22}[c_1\ c_2\ c_3\ 1\ 0\ 0\ 0]^T = \mathbf{0}_{3\times 1},$$

and, we see that, $[c_1\ c_2\ c_3\ 1\ 0\ 0\ 0]$ is the unique generating codeword of the cyclic MDS code $< H_{22} >^{\perp}$ (for a matrix H, $< H >^{\perp}$ denotes the code with parity-check matrix H). So, we get that

$$\mathbf{b}_4 = c_1\mathbf{b}_1 + c_2\mathbf{b}_2 + c_3\mathbf{b}_3 \quad (2)$$

resulting in

$$\begin{aligned} &c_1H(8) + c_2H(9) + c_3H(10) + H(11) + \\ &c_1b_{13}H(15) + c_2b_{23}H(16) + c_3b_{33}H(17) + b_{43}H(18) = \mathbf{0}_{6\times 1}. \end{aligned} \quad (3)$$

From (3), we see that

$$[\mathbf{0}_{1\times 7} | c_1\ c_2\ c_3\ 1\ 0\ 0\ 0 | c_1b_{13}\ c_2b_{23}\ c_3b_{33}\ b_{43}\ 0\ 0\ 0] \in C. \quad (4)$$

Since H_{22} is the parity-check matrix of a cyclic code, we have

$$H_{22}[0\ c_1\ c_2\ c_3\ 1\ 0\ 0]^T = \mathbf{0}_{3\times 1}.$$

So, writing $\mathbf{b}_5$ in the basis $B_{\{2,3,4\}}$ (which is, in fact, in terms of $\mathbf{b}_2$, $\mathbf{b}_3$ and $\mathbf{b}_4$), we get

$$\mathbf{b}_5 = c_1\mathbf{b}_2 + c_2\mathbf{b}_3 + c_3\mathbf{b}_4. \quad (5)$$

Proceeding as before, we get that

$$[\mathbf{0}_{1\times 7} | 0\ c_1\ c_2\ c_3\ 1\ 0\ 0 | 0\ c_1b_{23}\ c_2b_{33}\ c_3b_{43}\ b_{53}\ 0\ 0\ 0] \in C.$$

Since C is quasi-cyclic, we get that

$$[\mathbf{0}_{1\times 7} | c_1\ c_2\ c_3\ 1\ 0\ 0\ 0 | c_1b_{23}\ c_2b_{33}\ c_3b_{43}\ b_{53}\ 0\ 0\ 0] \in C. \quad (6)$$

By a similar argument, we further have

$$[\mathbf{0}_{1\times 7} | c_1\ c_2\ c_3\ 1\ 0\ 0\ 0 | c_1b_{33}\ c_2b_{43}\ c_3b_{53}\ b_{63}\ 0\ 0\ 0] \in C. \quad (7)$$

Adding the codewords in (4) and (6), and the codewords in (6) and (7), we get that

$$\begin{aligned} &[c_1(b_{13}+b_{23})\ c_2(b_{23}+b_{33})\ c_3(b_{33}+b_{43})\ (b_{43}+b_{53})\ 0\ 0\ 0], \\ &[c_1(b_{23}+b_{33})\ c_2(b_{33}+b_{43})\ c_3(b_{43}+b_{53})\ (b_{53}+b_{63})\ 0\ 0\ 0] \end{aligned}$$

are minimum weight codewords of $< H_{33} >^{\perp}$ with the same support. Therefore, these codewords are proportional to each other (or they could be equal, which is dealt with later). This means that

$$\frac{b_{13}+b_{23}}{b_{23}+b_{33}} = \frac{b_{23}+b_{33}}{b_{33}+b_{43}} = \frac{b_{33}+b_{43}}{b_{43}+b_{53}} = \frac{b_{43}+b_{53}}{b_{53}+b_{63}}.$$

Now, we can always find a $b_{73} \in$ GF(2^m) such that

$$\frac{b_{53}+b_{63}}{b_{63}+b_{73}} = \frac{b_{43}+b_{53}}{b_{53}+b_{63}}.$$

The existence of such a b_{73} (if all b_{i3} are equal, then b_{73} is simply equal to one of them) would imply that

$$[\mathbf{0}_{1\times 7}|c_1\ c_2\ c_3\ 1\ 0\ 0\ 0|c_1b_{43}\ c_2b_{53}\ c_3b_{63}\ b_{73}\ 0\ 0\ 0] \in C,$$

which in turn implies that

$$[\mathbf{0}_{1\times 7}|0\ 0\ 0\ c_1\ c_2\ c_3\ 1|0\ 0\ 0\ c_1b_{43}\ c_2b_{53}\ c_3b_{63}\ b_{73}] \in C.$$

Hence, we see that $H(14) + b_{73}H(21) \in N(M)$, and, finally, we have the contradiction that $\text{rank}(M_7) < 3$.

Thus, it is not possible to construct a $(7, 4)$ MSR quasi-cyclic code.

IV. Non-existence of quasi-cyclic MSR codes for $\alpha = n - k$ and $k \geq 4$

Linear MSR codes with $\alpha = n - k$ (no symbol extension) do not exist if $n < 2k - 2$ [3]. In this section, we show that quasi-cyclic MSR codes with $\alpha = n-k$ do not exist for $k \geq 4$ with no regard to rate. The proof is similar in spirit to that in Section III that dealt with the special case of $n = 7$. So, we will be brief and focus mostly on the generalization steps.

The proof is by contradiction. So, we assume that there exists an (n, k) quasi-cyclic MSR distributed storage code C with an $(n-k)\alpha \times n\alpha$ parity-check matrix H composed of $(n-k)\times n$ block matrices H_{ij}, $1 \leq j \leq i \leq \alpha$. Further, there exists an $\alpha \times (n-k)\alpha$ matrix M for regeneration such that $\text{rank}(M_i) = 1$, $1 \leq i \leq n-1$, and $\text{rank}(M_n) = \alpha$.

For $1 \leq i \leq n-1$, let $a_{ij} = [a_{ij1}\ a_{ij2}\ \cdots\ a_{ij\alpha}]$, $1 \leq j \leq \alpha - 1$ be a basis for $N(M_i)$. We see that

$$\mathbf{a}_{ij} = (a_{ij1}H(i) + a_{ij2}H(i+n)+ \cdots + a_{ij\alpha}H(i+(\alpha-1)n)) \in N(M)$$

for $1 \leq j \leq \alpha - 1$. The generalization of Lemma 1 is immediate.

Lemma *3:* For $S \subset [1 : n-1]$ with $|S| = n - k$, the set

$$B_S = \{\mathbf{a}_{ij} : i \in S, 1 \leq j \leq \alpha - 1\}$$

is a basis for $N(M)$.

The generalization of Lemma 2 needs a few more arguments.

Lemma *4:* For each $i \in [1 : n-1]$, $a_{ij1} \neq 0$ for at least one $j \in [1 : \alpha - 1]$.

Proof: We will prove for $i = 1$, since the proof for any i is similar. Suppose $a_{1j1} = 0$ for all $1 \leq j \leq \alpha - 1$. Writing $\mathbf{a}_{n-k+l,j}$ in terms of vectors in $B_{[1:n-k]}$, we get $a_{n-k+l,j,1} = 0$ for $1 \leq j \leq \alpha - 1$ for $1 \leq l \leq k - 1$. Writing $\mathbf{a}_{ij}$, $2 \leq i \leq n-k$, in the basis B_S with $S = [1 : i-1] \cup [i+1 : n-k+1]$, we get that $a_{ij1} = 0$. Thus, $a_{ij1} = 0$ for all $i \in [1 : n-1]$, $j \in [1 : \alpha-1]$. For each i, the $\mathbf{a}_{ij}$, $1 \leq j \leq \alpha-1$, are linearly independent. So, we can now set

$$\mathbf{a}_{i,\alpha-1} = [\mathbf{0}_{1\times\alpha-1}\ 1], 1 \leq i \leq n-1,$$

which implies that $H(i+(\alpha-1)n) \in N(M)$ for $1 \leq i \leq n-1$. This results in $H(n\alpha) \in N(M)$, and the contradiction that $\text{rank}(M_n) < \alpha$. ∎

Now, proceeding as in Section III, we can set, without loss of generality,

$$a_{ij} = [\mathbf{0}_{1\times j-1}\ 1\ \mathbf{0}_{1\times\alpha-j-1}\ a_{ij\alpha}]$$

for $1 \leq i \leq n-1$, $1 \leq j \leq \alpha - 1$. We focus on $j = \alpha - 1$ and set, for $1 \leq i \leq n-1$,

$$\mathbf{b}_i = \mathbf{a}_{i,\alpha-1} = H(i + (\alpha-2)n) + b_{i\alpha}H(i + (\alpha-1)n), \quad (8)$$

where $b_{i\alpha} = a_{i,\alpha-1,\alpha}$.

Now, expressing $\mathbf{b}_{n-k+j}$, $1 \leq j \leq k-1$ in terms of $\mathbf{b}_j$, $\mathbf{b}_{1+j}$, …, $\mathbf{b}_{n-k-1+j}$, we have that

$$\begin{aligned}\mathbf{c}_j =&[\mathbf{0}_{1\times(\alpha-2)n}| \\ &c_1 \quad c_2 \quad \cdots\ c_{n-k} \quad 1 \quad \mathbf{0}_{k-1}| \\ &c_1b_{j\alpha}\ c_2b_{1+j,\alpha}\ \cdots\ c_{n-k}b_{n-k-1+j,\alpha}\ b_{n-k+j,\alpha}\ \mathbf{0}_{k-1}] \in C.\end{aligned} \quad (9)$$

Considering $\mathbf{c}_j + c_{j+1}$, $1 \leq j \leq k-2$, we get $k-2$ minimum weight codewords of $< H_{\alpha,\alpha} >^{\perp}$ with the same support. Since $k \geq 4$, there are at least two such codewords, and a similar argument as in Section III shows the existence of b_{nj} such that

$$H((\alpha-1)n) + b_{nj}H(n) \in N(M)$$

resulting in the contradiction that $\text{rank}(M_n) < \alpha$.

This completes the proof. So, the only interesting parameters for quasi-cyclic MSR codes with $k/n > 1/2$ and no symbol extension are $(3, 2)$ and $(4, 3)$ with $\alpha = 1$, and $(5, 3)$ with $\alpha = 2$. Of these, $(3, 2)$ and $(4, 3)$ are easily seen to be not possible. So, the (5,3) quasi-cyclic MSR code with $\alpha = 2$ reported in [8] is the only non-trivial one with no symbol extension.

V. Numerical Search for Codes

Since it is not possible to construct (7,4) quasi-cyclic MSR codes with $\alpha = 3$, we attempted to search (by computer) for quasi-cyclic codes that perform close to MSR. The goal is to find H and M such that $\text{rank}(M_7) = 3$, and $\beta_i = \text{rank}(M_i)$ for $1 \leq i \leq 6$ are either 1 or 2. We obtained one code over GF(8) for which $\beta_1 = \beta_2 = \beta_4 = 1$ and $\beta_3 = \beta_5 = \beta_6 = 2$. To specify the parity-check matrix H, we provide the first rows of H_{ij}, denoted $h_{ij}(x)$ in polynomial notation ($\gamma \in$ GF(2^3) is primitive):

$$\begin{aligned}
h_{11}(x) &= \gamma + \gamma^3x + \gamma^6x^2 + \gamma^6x^3 + x^4,\\
h_{21}(x) &= \gamma^6 + \gamma^5x + \gamma^5x^2 + \gamma^2x^3 + x^4,\\
h_{31}(x) &= 1 + \gamma^4x + \gamma^2x^2 + \gamma^4x^3 + x^4,\\
h_{22}(x) &= \gamma^5 + \gamma^2x + \gamma^3x^2 + \gamma^3x^4 + \gamma^6x^5 + \gamma^4x^6,\\
h_{32}(x) &= \gamma^2 + \gamma^3x^2 + \gamma x^3 + \gamma^4x^4 + \gamma^5x^5 + \gamma^6x^6,\\
h_{33}(x) &= \gamma^4x + x^2 + \gamma x^3 + \gamma^3x^4 + \gamma x^5 + \gamma^3x^6.
\end{aligned}$$

The three regenerative vectors are given by

$$\begin{array}{lll}
[\gamma^4 1 \gamma^3 \gamma^2 \gamma^5 0\ \gamma & | & \gamma^2\gamma\gamma^6\gamma^6 1\gamma^2\gamma^6 & | & \gamma^5 1\gamma^6\gamma\gamma^6 1\gamma^5]\\
[1\ 1\gamma^2\gamma^6\gamma^6\gamma^6\gamma^3 & | & \gamma^5\gamma\gamma^5 0\ 0\ \gamma\gamma^6 & | & \gamma 1\gamma^5 0\gamma^2\gamma^6\gamma^4]\\
[\gamma^2\gamma^4\gamma^4 00\gamma^5\gamma^6 & | & 1\gamma^5 1\gamma^3\gamma\ \gamma\ 0 & | & \gamma^3\gamma^4 1\gamma^5\gamma^2 0\gamma]
\end{array}$$

The above (7,4) code is an improvement over the code reported in [8].

We have found (7,4) quasi-cyclic codes that are close to MSR with symbol extension. For instance, with $\alpha = 9$, we get H and M with $\beta_i = 4$, $1 \leq i \leq 6$ and rank$(M_7) = 9$. Note that an MSR code would have $\beta_i = 3$. Similarly we have a $(7, 4)$, $\alpha = 12$ code with $\beta_i = 5$, $1 \leq i \leq 5$ and rank$(M_7) = 12$, while for an MSR code β_i would have been 4. All these codes are over GF(8).

For $(n, k) = (9, 5)$, $\alpha = 4$, we found a code over GF(64) such that $\beta_i = 2$, $1 \leq i \leq 8$, and rank$(M_9) = 4$. For $(n, k) = (7, 2)$, $\alpha = 5$ in $GF(2^3)$, we found a code with $\beta_i = 1$, $1 \leq i \leq 5$, $\beta_6 = 2$, rank$(M_7) = 5$. A summary of our findings by computer search are given in Table I.

(n,k)	α	β_i, $1 \leq i \leq n-1$	Field size
(7,5)	2	[1 1 1 2 1 2]	8
(7,4)	3	[1 1 2 1 2 2]	8
	9	[4 4 4 4 4 4]	
	12	[5 5 5 5 5 5]	
(7,3)	4	[1 1 1 1 2 2]	8
	8	[3 3 3 3 3 3]	
(7,2)	5	[1 1 1 1 1 2]	8
	10	[3 3 3 3 3 3]	
(9,5)	4	[2 2 2 2 2 2 2 2]	64

TABLE I
CODES FOUND BY COMPUTER SEARCH.

VI. CONCLUDING REMARKS

We proved the non-existence of quasi-cyclic MSR codes with no symbol extension when $k \geq 4$. The condition $k \geq 4$ is quite intriguing, since it validates the existence of $(5, 3)$ MSR quasi-cyclic codes discussed in [8], and also precludes $(7, 4)$ quasi-cyclic MSR codes, for which there exist linear codes. This makes the quasi-cyclic requirement much stronger than that of the MSR requirement. It also emphasises, strongly, the difficulty in obtaining quasi-cyclic MSR codes for rate≥ 0.5.

The analysis of quasi-cyclic MSR codes with symbol extension ($\alpha = \mu(n-k)$, $\mu = 2, 3, \ldots$) is an interesting problem. An important factor in this analysis is the nature of the roots of the generator polynomials of $< H_{ii} >^{\perp}$. We state, without proof, the requirements that we could derive for the existence of a $(7, 4)$ quasi-cyclic MSR code with $\alpha = 6$. The requirements are the following: (1) the spacing between the zeros of $< H_{44} >^{\perp}$ and $< H_{55} >^{\perp}$ should be the same, and (2) the spacing between the zeros of $< H_{33} >^{\perp}$ and $< H_{66} >^{\perp}$ should be the same. However, the remaining requirements are non-linearly coupled and require further study.

Going by the results of computer search, it appears that a significantly large symbol extension will be needed for quasi-cyclic MSR codes, if they exist at all. Therefore, near-MSR codes offer an interesting compromise from a complexity perspective.

REFERENCES

[1] A. Dimakis, P. Godfrey, Y. Wu, M. Wainwright, and K. Ramchandran, "Network coding for distributed storage systems," *Information Theory, IEEE Transactions on*, vol. 56, pp. 4539 –4551, sept. 2010.

[2] A. Dimakis, K. Ramchandran, Y. Wu, and C. Suh, "A survey on network codes for distributed storage," *Proceedings of the IEEE*, vol. 99, pp. 476 –489, march 2011.

[3] N. Shah, K. Rashmi, P. Kumar, and K. Ramchandran, "Interference alignment in regenerating codes for distributed storage: Necessity and code constructions," *Information Theory, IEEE Transactions on*, vol. 58, pp. 2134 –2158, april 2012.

[4] I. Tamo, Z. Wang, and J. Bruck, "Zigzag codes: MDS array codes with optimal rebuilding," *to appear in IEEE Transactions on Information Theory*, vol. abs/1112.0371, 2011.

[5] V. R. Cadambe, C. Huang, S. A. Jafar, and J. Li, "Optimal repair of MDS codes in distributed storage via subspace interference alignment," *CoRR*, vol. abs/1106.1250, 2011.

[6] F. Oggier and A. Datta, "Self-repairing codes for distributed storage - a projective geometric construction," in *Information Theory Workshop (ITW), 2011 IEEE*, pp. 30 –34, oct. 2011.

[7] B. Gaston, J. Pujol, and M. Villanueva, "Quasi-cyclic minimum storage regenerating codes for distributed data compression," in *Data Compression Conference (DCC), 2011*, pp. 33 –42, march 2011.

[8] A. Thangaraj and C. Sankar, "Quasicyclic MDS codes for distributed storage with efficient exact repair," in *Information Theory Workshop (ITW), 2011 IEEE*, pp. 45 –49, oct. 2011.

[9] K. Lally and P. Fitzpatrick, "Algebraic structure of quasicyclic codes," *Discrete Applied Mathematics*, vol. 111, no. 12, pp. 157 – 175, 2001. Coding and Cryptology.