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

A Project Report

submitted by

VIGNESH.G

in partial fulfilment of the requirements
for the award of the degree of

MASTER OF TECHNOLOGY & BACHELOR OF TECHNOLOGY

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY MADRAS.

THESIS CERTIFICATE

This is to certify that the thesis titled **Quasi-Cyclic Regenerating Codes for Distributed Storage: Existence and Near-MSR Examples**, submitted by **Vignesh.G**, to the Indian Institute of Technology, Madras, for the award of the degree of **Dual Degree (B.Tech in Electrical Engineering, with an M.Tech in Communications)**, is a bona fide record of the research work done by him under our supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Dr. Andrew Thangaraj
Research Guide
Associate Professor
Dept. of Electrical Engineering
IIT-Madras, 600 036

Place: Chennai

Date: 16th May 2013

ACKNOWLEDGEMENTS

I would like to thank Dr.Andrew Thangaraj, who started off by being my faculty advisor, and has been a friend, philosopher and guide (quite literally) to me over the last five years. His role in my growth at IITM cannot be quantified in words. I would like to thanks Shreyas and Prasad for have been with me through thick and thin over the last five years. I would like to thank Vasuki, CJR, Shaileshh and Karthikram for have been a wonderful support system to me over my first four years at IITM. I would like to thank Sudharshan, Bhargava, Siddharth, NPS, Numaan and MRB for have been a huge part of what I would call the defining part of my life at IITM. I would like to thank Shruthy for have stood by and supported me during some tough times in my final year. I would like to thank Midhun for all those wonderful discussions, and walks to Tiffanys, and for have been my conscience and made me a more tolerant person. I would like to thank the department of Electrical Engineering for my experiences at E2A, and my team of four, hardworking folks who have become great friends with time, consisting of Syed, Sohini, Rohan and Saptarshi. In this regard I would also like to thank Dr.Enakshi Bhattacharya, Dr. Nitin Chandrachoodan, Dr. Anjan Chakravorty, Dr. Aniruddhan Sankaran, Vikram Srinivas and Praneeth Kumar. I would like to thank Jaichu, Manda, Baby, Sreeki, Rakesh, RK, Jana, Surya, Srini, Dhev, BS, Prem, Hari, Dhari, BKa, SG, Srinath, Shyam Krish, Nishaanth, Purnima, Aparna, Glucon, Ryali, Shyam, Kabra and all others who have made these five years at IITM memorable for me in some way or the other. I would like to thank Dr.Srikrishna Bhashyam, Dr. Radhakrishna Ganti, Dr. Devendra Jalihal, Dr.Giridhar, Varsha,

Archana, Sadhana, Nishidh, and all others who made the DD-Communications experience extra special. I would also like to thank faculty and friends who have inspired me at various levels which has kept me pushing, some of them including Dr.Bhaskar Ramamurthi, Dr. Shanthi Pavan, Dr.Suresh Govindaarajan, the late Ayush Joshi, Rakesh Misra, Varun Saravanan amongst others. I would like to thank my parents Mr.G.R.Ganapathi Subramanian, and Durga Ganapathi Subramanian, and my brother Heramba Kumar for have provided me with all the freedom one could hope for, imbibed in me free thinking and making such a wonderful home to always fall back onto. I would like to thank this wonderful campus of IIT Madras, right in nature's lap, it is something I will definitely miss. Finally I would like to thank the Almighty, Lord Mahaganapathi, without whom none of this was possible.

ABSTRACT

KEYWORDS: Regenerating codes ; Quasi-cyclic; Minimum Storage Regeneration; Maximum Distance Separable.

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.

TABLE OF CONTENTS

LIST OF TABLES

ABBREVIATIONS

MSR	Minimum Storage Regenerating
MDS	Maximum Distance Separable

NOTATION

M	Message vector matrix
M_i	The i^{th} regenerative matrix
α	Number of storage elements per node
H	Parity check matrix of the quasi-cyclic code
H_{ij}	$(n-k) \times n$ sub-matrix of H.
$[a:b]$	Set of all sequential elements from a to b
$a. * b$	An element-wise product of two vectors a and b(identical to the MATLAB notation)

CHAPTER 1

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 Dimakis *et al.* (2010). In the past few years, there has been very active research in this area, as summarized in Dimakis *et al.* (2011). Important code designs, methods and bounds were first presented and explored in Shah *et al.* (2012). Code constructions using array codes are presented in Tamo *et al.* (2011). The connection to linear algebraic "interference alignment" is particularly interesting, and has been explored further in Cadambe *et al.* (2011). Other algebraic code constructions include Oggier and Datta (2011) and GastoÌĄn *et al.* (2011).

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 Shah *et al.* (2012), the range of n and k for which MSR codes exist with no symbol extension is very limited. The constructions in Tamo *et al.* (2011)Cadambe *et al.* (2011) 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 GastoÌĄn *et al.* (2011); Thangaraj and Sankar (2011). In GastoÌĄn *et al.* (2011), 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 Lally and Fitzpatrick (2001). 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.

CHAPTER 2

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 Lally and Fitzpatrick (2001) and references thereon), a codeword

$c = [c_1\, c_2 \ldots c_n] \in C$ can be thought of as a concatenation of α vectors of length-n $\mathbf{c}_i = [c_{1,i}\, c_{2,i} \ldots c_{n,i}]$ 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 Lally and Fitzpatrick (2001), 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 $h_{ij}(x)$ Lally and Fitzpatrick (2001); Thangaraj and Sankar (2011).

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 Thangaraj and Sankar (2011). 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 3.1 and 3.2, 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.

CHAPTER 3

Results of non-existence for specific parameters

3.1 Non-existence of (7,4) quasi-cyclic MSR code for α=3

For $n = 7$, $k = 4$ and $\alpha = n - k = 3$ (no symbol extension), a linear MSR code is known to exist Shah *et al.* (2012). 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

$$[H(11)]_{4:6} = c_1[H(8)]_{4:6} + c_2[H(9)]_{4:6} + c_3[H(10)]_{4:6}. \tag{3.1}$$

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, (3.1) 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

$$\mathbf{b}_4 = c_1\mathbf{b}_1 + c_2\mathbf{b}_2 + c_3\mathbf{b}_3 \tag{3.2}$$

resulting in

$$\begin{aligned} &c_1H(8) + c_2H(9) + c_3H(10) + H(11) + \\ &\qquad c_1b_{13}H(15) + c_2b_{23}H(16) + c_3b_{33}H(17) + b_{43}H(18) = \mathbf{0}_{6\times 1}. \end{aligned} \tag{3.3}$$

From (3.3), we see that

$$[\mathbf{0}_{1\times 7}|c_1\ c_2\ c_3\ 1\ 0\ 0\ 0|c_1b_{13}\ c_2b_{23}\ c_3b_{33}\ b_{43}\ 0\ 0\ 0] \in C. \tag{3.4}$$

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

$$\mathbf{b}_5 = c_1\mathbf{b}_2 + c_2\mathbf{b}_3 + c_3\mathbf{b}_4. \tag{3.5}$$

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

$$[\mathbf{0}_{1\times 7}|c_1\ c_2\ c_3\ 1\ 0\ 0\ 0|c_1b_{23}\ c_2b_{33}\ c_3b_{43}\ b_{53}\ 0\ 0\ 0] \in C. \tag{3.6}$$

By a similar argument, we further have

$$[\mathbf{0}_{1\times 7}|c_1\ c_2\ c_3\ 1\ 0\ 0\ 0|c_1b_{33}\ c_2b_{43}\ c_3b_{53}\ b_{63}\ 0\ 0\ 0] \in C. \tag{3.7}$$

Adding the codewords in (3.4) and (3.6), and the codewords in (3.6) and (3.7), we get that

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

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 $b_{73} \in \text{GF}(2^m)$ 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

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

which in turn implies that

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

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.

3.2 Non-existence of quasi-cyclic MSR codes for α=n-k and k $\geq$ 4

Linear MSR codes with $\alpha = n-k$ (no symbol extension) do not exist if $n < 2k-2$ Shah *et al.* (2012). 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 3.1 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 rank$(M_i) = 1$, $1 \leq i \leq n - 1$, and 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 rank$(M_n) < \alpha$.

Now, proceeding as in Section 3.1, 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 $1 \leq i \leq n-1$,

$$\mathbf{b}_i = \mathbf{a}_{i,\alpha-1} = H(i + (\alpha - 2)n) + b_{i\alpha}H(i + (\alpha - 1)n), \tag{3.8}$$

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

$$\begin{aligned}\mathbf{c}_j =&[\mathbf{0}_{1\times(\alpha-2)n}|\\ &c_1 \quad c_2 \quad \cdots\ c_{n-k} \quad 1 \quad \mathbf{0}_{k-1}|\\ &c_1 b_{j\alpha}\ c_2 b_{1+j,\alpha}\ \cdots\ c_{n-k} b_{n-k-1+j,\alpha}\ b_{n-k+j,\alpha}\ \mathbf{0}_{k-1}] \in C.\end{aligned} \tag{3.9}$$

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 **??** shows the existence of b_{nj} such that

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

resulting in the contradiction that 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 Thangaraj and Sankar (2011) is the only non-trivial one with no symbol extension.

CHAPTER 4

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 rank$(M_7) = 3$, and $\beta_i =$ 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):

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

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 Thangaraj and Sankar (2011).

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 4.1.

(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 4.1: Codes found by computer search.

The exact code obtained by numerical search for $n = 7, k = 4$, with $\alpha = 9$, such that rank(M_i)=4, for $1 \leq i \leq 6$, and rank(M_7)=9 is presented in Appendix B.

CHAPTER 5

Role of constraints on the parity check matrix

In all the discussions thus far, we have not been able to get a hold on the exact nature of these cyclic codes. That is because for the MSR case with no symbol extension, the proof of contradiction doesn't require any specific structure, other than the plain cyclic nature. We know that the parity check matrices play a very important role in the exact form of the cyclic code. For instance, we know that any parity check matrix H_{ij} of a cyclic code can be characterized by the first row of the matrix, and this is given in polynomial notation as $h_{ij}(x)$. We know from Thangaraj and Sankar (2011) that those elements which are roots of both $h_{ii}(x)$ and $h_{jj}(x)$, are necessarily roots of $h_{ij}(x)$. This only enforces further constraints on the parity check structure. In cases of symbol extension, we realize that the conditions enforced by this quasi-cyclic structure play a fairly involved role in determining the non-existence of codes, for parameters where codes would have existed otherwise. This is observed while trying to prove the non-existence of $(7, 4)$ codes with $\alpha = 6$.

Lemma 5 *There exists no* $(7, 4)$ *code* C*, such that* $\alpha = 6$.

Proof: The proof of this lemma is presented in Appendix A.

The proof of the above lemma ensures that the roots of the polynomials forming the parity check matrices cannot be picked out at random. On the contrary, it is very specific parity check matrices, with particular properties which can be used as valid parity check matrices to form a code that can be regenerated.

CHAPTER 6

Existence of quasi-cyclic codes for random parameters

Most of the results presented up until now are negative results, proving non-existence of codes. But, in case we relax the MSR constraint on (n, k) codes, it is possible to obtain codes for all (n, k). We will prove that it is possible to obtain (n, k) codes such that if $(n-1)$ divides $k(n-k)$, rank(M_i)=$\frac{k(n-k)}{n-1}$, for $1 \leq i \leq n-1$, and rank(M_n) = $n-k$, where $\alpha = n-k$. To prove this, we first prove the following lemma.

Lemma 6 *For $S_i \subset [1:n]$, $1 \leq i \leq n-k$, with $|S_i| = n-k-1$, if the message vector matrix M, and the parity check matrix H are picked such that $k \in S_i, \Rightarrow$ M_i's k^{th} column is zero, this is definitely a code that can be reconstructed.*

Proof: The proof of this lemma is presented in Appendix .

Now, we can use the above lemma to prove the claim made in the beginning of this chapter. If the subsets S_i, $1 \leq i \leq n-k$ are picked such that $1, 2, ..., n-1$ occur an equal number of times in S_1 through S_{n-k}, then that would mean exactly $\frac{(n-k-1)(n-k)}{n-1}$ columns of M_i are zero, for $1 \leq i \leq n-1$. This means we have obtained a construction by which rank(M_i) = $n-k-\frac{(n-k-1)(n-k)}{n-1} = \frac{k(n-k)}{n-1}$, for $1 \leq i \leq n-1$. Since this is a code that can be reconstructed, from Lemma 6, it is implied that $rank(M_n) = n-k$.

The above proof also ensures that even if $n-1$ does not divide $k(n-k)$, picking the sets S_i in the above manner will give us $\sum_{i=0}^{i=n-1} rank(M_i) = k(n-k)$. The exact split of the ranks can be customised by manner of choosing S_i's appropriately.

CHAPTER 7

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 Thangaraj and Sankar (2011), 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.

APPENDIX A

Proof to Lemma 5

Consider an (n, k) quasi-cyclic MSR code for $\alpha = 6$. Let the parity check matrix have the regular lower-triangular sub-structure, containing matrices H_{ij} where $H_{ij} = 0_{3\times 7}$ for $i > j$. We also know that H_{ii} defines $(7, 4)$ codes for $1 \leq i \leq 6$, and that H_{ij} is defined such that the parity check polynomial of H_{ij}, $h_{ij}(x)$ necessarily has all roots which are common roots to the corresponding parity check polynomials $h_{ii}(x)$ and $h_{jj}(x)$.

Since the rank of the regenerative matrices M_i needs to be 2 for $1 \leq i \leq 6$, any 3 columns of M_i are dependent. Since $M_i = MH_i = [H(i)\, H(i+7) \ldots H(i+35)]$, there need to be four independent null-space vectors of M, which can be formed using the column vectors of H_i. This implies that we can obtain code-words of H of the form $[0_{1\times 21}\, c\, c. * \alpha\, c. * \beta]$. Here c is a minimum distance codeword of H_{44} of the form $[c_1\, c_2\, c_3\, 1\, 0\, 0\, 0]$. α is a vector of the form $[\alpha_1\, \alpha_2\, \alpha_3\, \alpha_4\, \alpha_5\, \alpha_6\, 0]$. Similarly, β is a vector of the form $[\beta_1\, \beta_2\, \beta_3\, \beta_4\, \beta_5\, \beta_6\, 0]$. There is also a codeword of the form $[0_{1\times 21}\, c\, c. * \alpha^{'}\, c. * \beta^{'}]$ and of the form $[0_{1\times 21}\, c\, c. * \alpha^{''}\, c. * \beta^{''}]$, where $\alpha^{'}$ refers to the cyclically left shifted version of α and $\alpha^{''}$ refers to the cyclically twice-left shifted version of α. $\beta^{'}$ and $\beta^{''}$ are defined similarly. Due to the rank 2 condition, there also exist codewords of the form $[0_{1\times 14}\, d\, \hat{d}\, (d. * \gamma + \hat{d}. * \alpha)\, (d. * \delta + \hat{d}. * \beta)]$. Similarly there also exist codewords of the form $[0_{1\times 14}\, d\, \hat{d}\, (d. * \gamma^{'} + \hat{d}. * \alpha^{'})\, (d. * \delta^{'} + \hat{d}. * \beta^{'})]$ and $[0_{1\times 14}\ d\ \hat{d}\ (d. * \gamma^{''} + \hat{d}. * \alpha^{''})\ (d. * \delta^{''} + \hat{d}. * \beta^{''})]$, where γ, δ are vectors defined similarly as α, β and $\gamma^{'}, \gamma^{''}, \delta^{'}, \delta^{''}$ are also defined as their corresponding predecessors.

Now, let us define $\alpha_i + \alpha_{i+1} = A_i$,$\beta_i + \beta_{i+1} = B_i$,$\gamma_i + \gamma_{i+1} = G_i$ and $\delta_i + \delta_{i+1} = D_i$. Now, among the obtained codewords, adding the first to the second, the second to the third, the fourth to the fifth and the fifth to the sixth codewords, we obtain four new codewords of H of the following form

$$C_1 = [0_{1\times 28}\ c_1A_1\ c_2A_2\ c_3A_3\ A_4\ 0_{1\times 3}\ c_1B_1\ c_2B_2\ c_3B_3\ B_4\ 0_{1\times 3}]$$

$$C_2 = [0_{1\times 28}\ c_1A_2\ c_2A_3\ c_3A_4\ A_5\ 0_{1\times 3}\ c_1B_2\ c_2B_3\ c_3B_4\ B_5\ 0_{1\times 3}]$$

$$C_3 = [0_{1\times 28}d_1G_1+\hat{d}_1A_1d_2G_2+\hat{d}_2A_2d_3G_3+\hat{d}_3A_3G_4 0_{1\times 3}d_1D_1+\hat{d}_1B_1d_2D_2+\hat{d}_2B_2d_3D_3+\hat{d}_3B_3D_4 0_{1\times 3}]$$

$$C_4 = [0_{1\times 28}d_1G_2+\hat{d}_1A_2d_2G_3+\hat{d}_2A_3d_3G_4+\hat{d}_3A_4G_5 0_{1\times 3}d_1D_2+\hat{d}_1B_2d_2D_3+\hat{d}_2B_3d_3D_4+\hat{d}_3B_4D_5 0_{1\times 3}]$$

(A.1)

The first seven elements of each of these codewords are minimum distance codewords of H_{55}, and so are scalar multiples of one another. Assuming the first seven elements of the C_3 is $k^{'}$ times the first seven elements of C_1, and the first seven elements of the C_4 is $k^{''}$ times the first seven elements of the C_2, we can perform the following operations $D_1 = C_3 + k^{'}C_1$ and $D_2 = C_4 + k^{''}C_2$. The first seven elements of D_1 and D_2 are all zero by definition of $k^{'}, k^{''}$. This gives us the

following equations

$$
\begin{aligned}
G_1 + \frac{(\hat{d_1} + k^{'} c_1)}{d_1} A_1 = 0. \\
G_2 + \frac{(\hat{d_2} + k^{'} c_2)}{d_2} A_2 = G_2 + \frac{(\hat{d_1} + c_1 k^{''})}{d_1} A_2 = 0. \\
G_3 + \frac{(\hat{d_3} + k^{'} c_3)}{d_3} A_3 = G_3 + \frac{(\hat{d_2} + c_2 k^{''})}{d_2} A_3 = 0. \\
G_4 + k^{'} A_4 = G_2 + \frac{(\hat{d_3} + c_3 k^{''})}{d_3} A_4 = 0. \\
G_5 + k^{''} A_5 = 0.
\end{aligned}
\tag{A.2}
$$

Since none of the A_i values can be zero (which would mean all the values are zero, and hence there will exist a value of β_7 to make M_7 not full rank), the above equations necessarily mean that $\frac{(\hat{d_2}+k^{'}c_2)}{d_2} = \frac{(\hat{d_1}+c_1k^{''})}{d_1}$, $\frac{(\hat{d_3}+k^{'}c_3)}{d_3} = \frac{(\hat{d_2}+c_2k^{''})}{d_2}$ and $k^{'} = \frac{(\hat{d_3}+c_3k^{''})}{d_3}$. Let us call $s_1 = \frac{(\hat{d_1}+c_1k^{'})}{d_1}$,$s_2 = \frac{(\hat{d_2}+k^{'}c_2)}{d_2} = \frac{(\hat{d_1}+c_1k^{''})}{d_1}$, $s_3 = \frac{(\hat{d_3}+k^{'}c_3)}{d_3} = \frac{(\hat{d_2}+c_2k^{''})}{d_2}$,$s_4 = k^{'} = \frac{(\hat{d_3}+c_3k^{''})}{d_3}$ and $s_5 = k^{''}$.

Now consider the last seven elements of D_1 and D_2. They form minimum distance codewords of H_{66}. These codewords are given by $D_1 = [d_1(G_1 + s_1A_1)\ d_2(G_2 + s_2A_2)\ d_3(G_3 + s_3A_3)\ G_4 + s_4A_4\ 0\ 0\ 0]$ and $D_2 = [d_1(G_2 + s_2A_2)\, d_2(G_3 + s_3A_3)\, d_3(G_4 + s_4A_4)\, G_5 + s_5A_5\, 0\, 0\, 0]$. Since both of these are minimum distance codewords, they are scalar multiples of each other. This implies that if $[a_1\ a_2\ a_3\ 1\ 0\ 0\ 0]$ is a minimum distance codeword of H_{66}, $1, \frac{d_3}{a_3}, \frac{d_2}{a_2}, \frac{d_1}{a_1}$ form a GP. This means that the common difference between the roots of $h_{33}(x)$ and the roots of $h_{66}(x)$ are identical. We also know that the common difference between the roots of $h_{44}(x)$ and those of $h_{55}(x)$ are also identical. We have also proved

that since M_7 needs to be full rank, the two common differences (that of h_{33}, h_{66} and that of h_{44}, h_{55}) must necessarily be different.

Now, since $[d\,\hat{d}]$ is a codeword of the sub-matrix containing H_{33}, H_{43}, H_{44}, we know that the following two equations hold.

$$\begin{aligned} h_{33}(x)d(x) &= 0. \\ h_{43}(x)d(x) + h_{44}(x)\hat{d}(x) &= 0. \end{aligned} \tag{A.3}$$

The first equation implies that $d(x)$ has roots which are exactly the three non-roots of $h_{33}(x)$ (Since the largest exponent in $d(x)$ is x^3 due to its minimum distance property). By definition of $h_{43}(x)$, it necessarily contains roots which are common to both $h_{33}(x)$ and $h_{44}(x)$. This means the product $h_{43}(x)d(x)$ necessarily divides $h_{44}(x)$. For the second equation to be zero, the term $(\frac{h_{43}(x)d(x)}{h_{44}(x)} + \hat{d}(x)$ must necessarily have roots which are non-roots of $h_{44}(x)$. Since by definition of $d(x)$, the first term of the summation has all roots which are non-roots of both $h_{33}(x)$ and $h_{44}(x)$, $\hat{d}(x)$ must necessarily contain roots which are non-roots of both $h_{33}(x)$ and $h_{44}(x)$. Since $h_{33}(x)$ and $h_{44}(x)$ have necessarily different common differences of roots, they must necessarily have at least one common root. So assume there is more than one common root, then the number of common non-roots are 3 in number, which would make the leading exponent of $\hat{d}(x)$ go up to 3. This is not possible since $\hat{d}(x)$ is a quadratic expression. Therefore, the number of common roots is exactly one, and the number of common non-roots is exactly 2.

Now assume the roots of $h_{33}(x)$ are $\alpha^{i_1}, \alpha^{i_1+\Delta_1}, \alpha^{i_1+2\Delta_1}$, and those of $h_{44}(x)$ are $\alpha^{i_2}, \alpha^{i_2+\Delta_2}, \alpha^{i_2+2\Delta_2}$ (note that one of the six roots is shared), and the sum of

the roots of $\hat{d}(x)$ is S, and the product of the roots is P. From the equations in A, we know that the following determinant must be 0 for a non-zero solution of $\hat{d}$.

$$\begin{bmatrix} \frac{\hat{d_1}}{d1} + \frac{\hat{d_2}}{d2} + \frac{\hat{d_3}}{d3} & \frac{\hat{d_2}}{d2} + \frac{\hat{d_3}}{d3} & \frac{\hat{d_3}}{d3} \\ \\ \frac{c_1}{d_1} & \frac{c_2}{d_2} & \frac{c_3}{d_3} \\ \\ \frac{c_2}{d_2} & \frac{c_3}{d_3} & 1 \end{bmatrix}$$

By definition of roots of $h_{33}(x)$ and $h_{44}(x)$, we have $d_3 = \alpha^{i_1}(1+\alpha^{\Delta_1}+\alpha^2\Delta_1)$, $d_2 = \alpha^{2i_1+\Delta_1}(1 + \alpha^{\Delta_1} + \alpha^2\Delta_1)$ and $d_1 = \alpha^{3i_1+3\Delta_1}$. We also have $c_3 = \alpha^{i_2}(1 + \alpha^{\Delta_2} + \alpha^2\Delta_2)$, $c_2 = \alpha^{2i_2+\Delta_2}(1 + \alpha^{\Delta_2} + \alpha^2\Delta_2)$ and $c_1 = \alpha^{3i_2+3\Delta_2}$.

Then the determinant equation reduces to the following:

$$\begin{aligned} &P(1 + \alpha^{\Delta_1} + \alpha^{2\Delta_1})(1 + \alpha^{\Delta_2} + \alpha^{2\Delta_2})+ \\ &S\alpha^{i_1+2\Delta_1}(\alpha^{i_2-i_1+\Delta_2-\Delta_1}(1 + \alpha^{\Delta_1+\Delta_2})(\alpha^{\Delta_1} + \alpha^{\Delta_2}) + (1 + \alpha^{\Delta_2} + \alpha^{2\Delta_2}))+ \\ &\alpha^{2i_1+3\Delta_1+(i_2-i_1+\Delta_2-\Delta_1)}((1+\alpha^{\Delta_1+\Delta_2})(\alpha^{\Delta_1}+\alpha^{\Delta_2})+\alpha^{i_2-i_1+\Delta_2-\Delta_1}(1+\alpha^{\Delta_2}+\alpha^{2\Delta_2})) = 0 \end{aligned} \tag{A.4}$$

A simple search over all possible $i_1, i_2, \Delta_1, \Delta_2$ will show that this equation has no solutions in $GF(2^3)$. Therefore the only possible solution is $\hat{d} = 0_{1\times 7}$. This would imply that the roots of $h_{33}(x)$ and $h_{44}(x)$ must have the same common difference which gives us a contradiction.

APPENDIX B

Sample non-MSR (7,4) code with rank(M_i)=4 for $1 \leq i \leq 8$, and rank(M_9)=9.

The parity check matrices for the same are as follows.

$$
\begin{aligned}
h_{11}(x) &= \gamma^2 + \gamma^2 x + \gamma^3 x^2 + \gamma x^3 + x^4 \\
h_{22}(x) &= \gamma^4 + x + \gamma^4 x^2 + \gamma^5 x^3 + x^4 \\
h_{33}(x) &= \gamma^6 + \gamma^5 x + \gamma^5 x^2 + \gamma^4 x^3 + x^4 \\
h_{44}(x) &= \gamma + \gamma^3 x + \gamma^6 x^2 + \gamma^6 x^3 + x^4 \\
h_{55}(x) &= \gamma^3 + \gamma x + x^2 + \gamma^3 x^3 + x^4 \\
h_{66}(x) &= \gamma^5 + \gamma^6 x + \gamma x^2 + x^3 + x^4 \\
h_{77}(x) &= 1 + \gamma^4 x + \gamma^2 x^2 + \gamma^4 x^3 + x^4 \\
h_{88}(x) &= \gamma^2 + \gamma^2 x + \gamma^3 x^2 + \gamma x^3 + x^4 \\
h_{99}(x) &= \gamma^4 + \gamma x + \gamma^4 x^2 + \gamma^5 x^3 + x^4
\end{aligned}
\tag{B.1}
$$

The regenerative vectors are given by

$$c_1 = [\gamma^2\gamma^6\gamma^2\gamma^2\gamma^6\gamma\gamma^4\gamma^3\gamma^3\gamma^6 1\gamma^3\gamma\gamma\gamma^2 1\gamma^2 1\gamma^6 1\gamma^3$$
$$\gamma^5\gamma^3\gamma^6 0\gamma^2\gamma^5\gamma 1\gamma\gamma\gamma\gamma^2\gamma 010\gamma^2\gamma^2\gamma^2 0\gamma$$
$$\gamma\gamma^6 0\gamma^2\gamma^6\gamma^4\gamma^4\gamma^2\gamma^4\gamma^4 1\gamma^6\gamma^2\gamma^4\gamma^5\gamma^4\gamma^4 0\gamma^5 01]$$
$$c_2 = [\gamma^3 1\gamma^2\gamma^3\gamma^6\gamma^2\gamma^3\gamma^2\gamma^3\gamma\gamma^4\gamma^4\gamma\gamma^5\gamma^5\gamma^6\gamma^5\gamma^3\gamma^2 11$$
$$\gamma^6 1\gamma 0110\gamma^3\gamma^3\gamma^3\gamma^3\gamma^5\gamma^2\gamma^3\gamma^4\gamma 0\gamma^6\gamma^6\gamma^3\gamma^4$$
$$1\gamma^4\gamma 4\gamma^5\gamma 1\gamma^4\gamma^5\gamma^4\gamma^5\gamma^6\gamma^2\gamma^3\gamma 10\gamma 0\gamma^3\gamma 1]$$
$$c_3 = [\gamma 1\gamma^3\gamma^5\gamma^2\gamma^2\gamma^5\gamma^2 0\gamma^5 1\gamma^2\gamma^4\gamma^6\gamma^3 01\gamma^5\gamma^6\gamma^3\gamma$$
$$1\gamma^6\gamma\gamma^2\gamma^2 0\gamma\gamma\gamma^3\gamma^6 0\gamma\gamma 0\gamma^6 1\gamma^4\gamma^2 00\gamma$$
$$\gamma^6\gamma^5 0\gamma^6\gamma^4 0\gamma^5 1\gamma\gamma^5\gamma^5\gamma\gamma^3\gamma^3\gamma^4\gamma 1\gamma\gamma^6 1\gamma 4]$$
$$c_4 = [\gamma^2\gamma^4\gamma^5\gamma^3\gamma^6\gamma^2\gamma\gamma^5\gamma^4\gamma^3\gamma^6 1\gamma^5\gamma^5\gamma^2\gamma\gamma\gamma^4 1\gamma^4 1$$
$$\gamma^4 0\gamma^3 11\gamma^5\gamma\gamma^3\gamma^2\gamma^3\gamma^4 1\gamma^5\gamma\gamma^4\gamma^2\gamma^6\gamma\gamma^2\gamma^3 1$$
$$\gamma 0\gamma^2\gamma^2\gamma\gamma^3\gamma^2\gamma^6\gamma^6\gamma^4\gamma 6\gamma^5\gamma^2\gamma^6\gamma^3\gamma^3 1\gamma^6 1\gamma^5\gamma^6]$$
$$c_5 = [\gamma^2\gamma\gamma^3\gamma^2\gamma\gamma\gamma^2 0\gamma 1\gamma^6\gamma^4 00\gamma^2\gamma\gamma^6 1\gamma^5 0\gamma$$
$$\gamma 1\gamma 1\gamma\gamma\gamma^5 0\gamma^5 0\gamma^6\gamma^4\gamma^6 0\gamma^5\gamma\gamma^3\gamma^2\gamma^2\gamma^6\gamma^5$$
$$\gamma\gamma^4\gamma^2\gamma^5\gamma^6\gamma^6\gamma^3 1\gamma^5 00\gamma^6\gamma^4\gamma^3\gamma^6\gamma\gamma^3\gamma^6\gamma^2\gamma^2\gamma]$$
$$c_6 = [\gamma^6\gamma^6\gamma^5\gamma^3 1\gamma^4\gamma\gamma^5\gamma\gamma^2\gamma^3\gamma^4 00\gamma^2 0\gamma^3\gamma^3\gamma^2 0\gamma^5$$
$$0\gamma^2\gamma^5 1\gamma\gamma^4\gamma^4 1\gamma^2\gamma^5\gamma^5\gamma^2\gamma^6 111\gamma^6\gamma^4\gamma^3\gamma^2 0$$
$$\gamma^4\gamma^6\gamma^3 0\gamma^6 0\gamma^4\gamma^3\gamma^6\gamma^2 100\gamma^2\gamma^3 10\gamma^6\gamma\gamma^2\gamma^5]$$
$$c_7 = [\gamma^6 1\gamma^6\gamma\gamma\gamma^5\gamma^5\gamma^2 0\gamma\gamma^5\gamma^6\gamma^2\gamma 110\gamma^6\gamma^2\gamma\gamma$$
$$\gamma^3\gamma^3\gamma^4 0\gamma^2\gamma^4\gamma^4\gamma^4\gamma^3\gamma^4\gamma^5\gamma^5\gamma^4\gamma^4\gamma^5 0\gamma\gamma^2\gamma^5\gamma^4 1$$
$$\gamma^2 011\gamma^5\gamma^6\gamma^2\gamma\gamma^4 1\gamma^2 0\gamma^3\gamma 1\gamma^4 10\gamma^4\gamma^5\gamma^5]$$
$$c_8 = [\gamma^5 10\gamma 0\gamma^4 0\gamma^2 01\gamma^2\gamma\gamma^3\gamma^5\gamma^5\gamma^6\gamma^6\gamma^3\gamma^6\gamma^2\gamma^2$$
$$\gamma^4 1\gamma^6\gamma^5\gamma^2\gamma^2\gamma^6\gamma 01\gamma\gamma^6 1\gamma^5 1\gamma^6\gamma^5\gamma^2 1\gamma^5\gamma^2$$
$$\gamma^6\gamma^6 1\gamma^5 0\gamma^6 01\gamma^5\gamma^3 1\gamma^2 0\gamma\gamma^2\gamma^2\gamma^3\gamma^4\gamma^3 1\gamma^4]$$
$$c_9 = [1\gamma^4\gamma^4 01\gamma^2\gamma^4\gamma^2\gamma^3\gamma^5\gamma^5\gamma^5 0\gamma^6 0\gamma\gamma^3\gamma^5\gamma^3 0\gamma^5$$
$$0\gamma^4 11\gamma^4\gamma^2\gamma^5\gamma^5\gamma^5\gamma^3\gamma\gamma\gamma^3\gamma^2 001\gamma\gamma^2\gamma^4\gamma^2$$
$$\gamma^5\gamma^6 11\gamma^4\gamma\gamma^3\gamma\gamma^5\gamma^4\gamma^3 11\gamma^3\gamma^6\gamma^5\gamma^2\gamma^2\gamma^4\gamma^4\gamma^5]$$

(B.2)

APPENDIX C

Proof to Lemma 6

The size of the set S_i is given to be $n-k-1$. This means that there are exactly $n-k-1$ matrices $M_j, 1 \leq j \leq n-1$ with the i^{th} column being zero. This would in turn mean that there are exactly $n-k-1$ null-space vectors of M with the first $(i-1)(n-k)$ elements being all zero, followed by $n-k$ non-zero elements belonging to columns of H_{ii}. Since the minimum distance of H_{ii} is $n-k+1$, it follows that all these $n-k-1$ null-space vectors are independent, and necessarily in their first $n-k$ non-zero elements itself. All the above statements are true for all values of i from 1 to $n-k$. Now, since the number of zero-elements in the top are different for each i, it is also true that all the $(n-k)(n-k-1)$ null-space vectors we have now obtained from columns of our parity check matrix H are independent. Also, since the dimension of the null-space of M is $(n-k)(n-k-1)$, these set of null-space vectors form a basis for the null-space of M.

Let us consider M_n. If M_n is not full-rank, then it is necessary that there exists some combination of the jn^{th} columns of H, $1 \leq j \leq n-k$, which belongs to the null-space of M. Let the smallest such index j be equal to p. This means there exists a null-space vector of M with the first $(p-1)(n-k)$ elements being all zero, and the next $n-k$ elements being the last column of H_{pp}. It is also necessary that this null-space vector can be expressed in terms of the basis defined above. Now, since the first $(p-1)(n-k)$ are all-zero, the null-space vectors having non-zero elements in the first $(p-1)(n-k)$ positions are not relevant. Also the relevant null-space vectors whose first $n-k$ non-zero elements are columns of H_{pp} are

only $n-k-1$ in number, which is lesser than the minimum distance of the code defined by H_{pp}, it is not possible to find such a basis vector representation for that particular null-space vector. This implies that M_n must necessarily be full-rank, thereby proving Lemma 6.

REFERENCES

1. **Cadambe, V. R.**, **C. Huang**, **S. A. Jafar**, and **J. Li** (2011). Optimal repair of MDS codes in distributed storage via subspace interference alignment. *CoRR*, **abs/1106.1250**.

2. **Dimakis, A.**, **P. Godfrey**, **Y. Wu**, **M. Wainwright**, and **K. Ramchandran** (2010). Network coding for distributed storage systems. *Information Theory, IEEE Transactions on*, **56**(9), 4539 –4551. ISSN 0018-9448.

3. **Dimakis, A.**, **K. Ramchandran**, **Y. Wu**, and **C. Suh** (2011). A survey on network codes for distributed storage. *Proceedings of the IEEE*, **99**(3), 476 –489. ISSN 0018-9219.

4. **GastoÌĄn, B.**, **J. Pujol**, and **M. Villanueva**, Quasi-cyclic minimum storage regenerating codes for distributed data compression. *In Data Compression Conference (DCC), 2011*. 2011. ISSN 1068-0314.

5. **Lally, K.** and **P. Fitzpatrick** (2001). Algebraic structure of quasicyclic codes. *Discrete Applied Mathematics*, **111**(12), 157 – 175. ISSN 0166-218X. URL `http://www.sciencedirect.com/science/article/pii/S0166218X00003504`. Coding and Cryptology.

6. **Oggier, F.** and **A. Datta**, Self-repairing codes for distributed storage - a projective geometric construction. *In Information Theory Workshop (ITW), 2011 IEEE*. 2011.

7. **Shah, N.**, **K. Rashmi**, **P. Kumar**, and **K. Ramchandran** (2012). Interference alignment in regenerating codes for distributed storage: Necessity and code constructions. *Information Theory, IEEE Transactions on*, **58**(4), 2134 –2158. ISSN 0018-9448.

8. **Tamo, I.**, **Z. Wang**, and **J. Bruck** (2011). Zigzag codes: MDS array codes with optimal rebuilding. *to appear in IEEE Transactions on Information Theory*, **abs/1112.0371**.

9. **Thangaraj, A.** and **C. Sankar**, Quasicyclic MDS codes for distributed storage with efficient exact repair. *In Information Theory Workshop (ITW), 2011 IEEE*. 2011.

LIST OF PAPERS BASED ON THESIS

1. Vignesh G and Andrew Thangaraj Quasi-Cyclic Regenerating Codes for Distributed Storage: Existence and Near-MSR Examples Accepted for presentation at *ISIT-2013*, Istanbul, Turkey in July 2013