Quasi-Cyclic LDPC Codes with Parity-Check Matrices of Column Weight Two for Erasure Correction

Xin Xiao and Bane Vasić
University of Arizona
Tucson, AZ 85721, USA
Email: {7xinxiao7, vasic}@email.arizona.edu

Juane Li
Micron Technology Inc.
Milpitas, CA 95035, USA
Email: jueli@ucdavis.edu

Shu Lin and Khaled Abdel-Ghaffar
University of California
Davis, CA 95616, USA
Email: {shulin, ghaffar}@ucdavis.edu

***Abstract*—In his pioneering work on LDPC codes, Gallager dismissed codes with parity-check matrices of weight two after proving that their minimum Hamming distances grow at most logarithmically with their code lengths. Nevertheless, many coding applications do not demand large minimum Hamming distances such as correcting bursts of erasures. In this paper, we revisit LDPC codes with parity-check matrices of weight two, in particular, matrices composed of two rows of circulant permutation matrices (CPMs). The quasi-cyclic structure of the codes naturally defines a sectionalized structure for codewords. We consider channels causing erasures confined to a number of sections and show that codes with parity-check matrices composed of two rows of CPMs are optimal compared to codes with three or more rows.**

I. Introduction

With the rediscovery of low-density parity-check (LDPC) codes by the turn of the century, researchers have recognized that LDPC codes have good erasure correcting capability in addition to their superior performance over AWGN channels. A simple "peeling" algorithm that can be applied to a sparse parity-check matrix of the code to correct erasures was proposed early on [1]. The algorithm may not correct all erasures that can be corrected by an optimal maximum-likelihood (ML) decoder. However, for long LDPC codes, it is very difficult to determine the capability of an ML decoder to correct erasures let alone implement such a decoder. Motivated by potential applications of LDPC codes in storage systems and communication over fading channels, researchers investigated the capability of LDPC codes to correct erasure bursts, and in particular one long burst of erasures [2]– [6].

In this paper, we consider the case in which a codeword is partitioned into *sections* of equal length. Each section may correspond, for example, to a part of a large file that is stored at a node in a distributed storage system. That part of the file may be subject to losses. In coding-theoretic context, the file can be viewed as a sequence composed of sections, each corresponding to a part of the file. Losses in part of a file stored at a node can then be modeled as a "phased burst" of erasures in which all the erasures are confined to a section.

We consider binary quasi-cyclic (QC) LDPC codes, with parity-check matrices which are $m \times n$ arrays of circulant permutation matrices (CPMs) of size $t \times t$. These are the most widely known, studied, and used QC-LDPC codes. A codeword $\mathbf{v}$ can be written as $\mathbf{v} = (\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_{n-1})$, where $\mathbf{v}_j$, $0 \leq j < n$, is a sequence of t bits that forms a section. We assume that such a codeword is transmitted over a channel that causes multiple phased bursts of erasures. Notice that no QC-LDPC code with a parity-check matrix composed of CPMs can correct two "solid" phased bursts, each having t erasures, i.e., all bits in a section are erased. Therefore, the best we can hope for is to correct pairs of mutually "semi-solid" phased bursts of erasures in which all the bits in two sections are erased except for one bit. We show that a QC-LDPC code with parity-check matrix of column weight two, i.e., composed of just two row blocks of CPMs, if properly designed, can correct any pair of such phased bursts using the peeling algorithm. Codes with such parity-check matrices have the highest possible dimension among all codes with this correction capability. Ge and Xia call a parity-check matrix composed of two rows of CPMs *ultra sparse* [7]. In our investigation, we determine the dimensions and minimum Hamming distances of all codes with such parity-check matrices as well as the girths of the Tanner graphs representing these matrices.

This paper is organized as follows. The notation for burst erasures, QC-LDPC codes, and their parity-check matrices with some basic results are presented in Section II. The main contributions are in Section III which covers QC-LDPC codes with parity-check matrices of weight two. The paper is concluded in Section IV. For smooth reading, all proofs are relegated to appendices.

II. Preliminaries

A. Correcting Bursts of Erasures

We consider binary transmission over an erasure channel in which the value of a transmitted bit is either received correctly or erased. The decoder knows exactly the set of indices, $\mathcal{J}$, of the erased bits. To be able to recover the values of the erased bits, a binary linear code is used. An (N, K) binary linear code, C, is the K-dimensional null space of an $M \times N$ binary matrix, $\mathbf{H}$, for some integer $M \geq N - K$. This matrix is a *parity-check matrix* for the code, the rank of

which is $\mathrm{rank}(\mathbf{H}) = N - K$, which we call the *redundancy* of the code. For any codeword $\mathbf{v}$ in C, we have $\mathbf{v}\mathbf{H}^{\mathsf{T}} = \mathbf{0}$ where computations are over GF(2) and T denotes transpose. Suppose $\mathbf{v}$ is transmitted over the channel and e erasures occur in the bits indexed by $\mathcal{J}$. Then, an ML decoder [8], [9] can recover the erased bits if and only if the code does not have any nonzero codeword in which the indices of all bits of value 1 are confined to $\mathcal{J}$. This is equivalent to the condition that the rank of the submatrix, $\mathbf{H}_{\mathcal{J}}$, of $\mathbf{H}$ composed of the columns indexed by $\mathcal{J}$, is $|\mathcal{J}|$. In this case, we say that the erasures are *recoverable* by the ML decoder. By considering the values of the erased bits to be unknowns in the codeword $\mathbf{v}$, these unknowns can be determined from $\mathbf{v}\mathbf{H}^{\mathsf{T}} = \mathbf{0}$ which is a system of M parity equations. A necessary condition for this to be possible is that $N-K \geq e$. Codes meeting this bound with equality are said to be *optimal* for the erasures specified by $\mathcal{J}$. Although a code in general has many parity-check matrices, its ability to correct erasures does not depend on the choice of $\mathbf{H}$ to solve for the unknowns in the equation $\mathbf{v}\mathbf{H}^{\mathsf{T}} = \mathbf{0}$. However, if e is large, say in the hundreds, then solving this system of equations may be computationally intensive.

In 2001, Luby *et al.* [1] came up with a simple decoding algorithm to correct erasures. The algorithm is applied to a particular parity-check matrix of the code and its success depends on this matrix. Although the algorithm may not be able to recover all erasures recoverable by an ML decoder, it is quite simple as it allows for the recovery of the erased bits one by one. Basically, if there is a parity equation that contains, i.e., checks, only one unknown erasure, then the erased value can be determined from that equation by an XOR operation and the number of unknowns is then reduced by one. Next, if another parity equation is found that contains only one of the remaining unknowns, then that unknown can be determined and the number of unknowns is further reduced by one. This may continue until all erasures are recovered or until no equation is found that contains only one unknown erasure in which case decoding fails and the remaining erased positions form a *stopping set* [8]. Although there is no universal term to identify this algorithm in the coding literature, some call it figuratively the *peeling algorithm* [10], a term which we will adopt. The peeling algorithm was initially developed for randomly constructed low-density parity-check (LDPC) codes and applied to their sparse parity-check matrices. The randomness makes it hard to develop erasure decoding algorithms that exploit the structure of the codes. On the other hand, the sparseness helps in having parity equations involving a small number of terms for which the peeling algorithm is most effective.

The peeling algorithm is best understood in terms of the *Tanner graph*, $\mathcal{G}$, representing the parity-check matrix $\mathbf{H} = [h_{I,J}]_{0\leq I<M, 0\leq J<N}$ [11]– [13]. This is a bipartite graph in which the set of vertices is partitioned into a set of *variable nodes* indexed by the columns of $\mathbf{H}$ and a set of *check nodes* indexed by the rows of $\mathbf{H}$. Edges connect only variable nodes to check nodes. In particular, there is an edge connecting the variable node corresponding to the J-th column to the check node corresponding to the I-th row if and only if $h_{I,J} = 1$. Since the code is the null space of $\mathbf{H}$, if the variable nodes assume the bit values of a codeword, then the sum over GF(2) of the values of the variable nodes adjacent to each check node is zero. The peeling algorithm looks for a check node which is adjacent to only one erased variable node and determines its value as the sum over GF(2) of the values of all other variable nodes adjacent to the check node. The number of erasures is then reduced by one and the process is repeated until all erased bits are recovered, in which case decoding is successful, or there is no check node that checks exactly one erased variable node, in which case decoding fails as the remaining variable nodes form a stopping set. The success of the peeling algorithm depends on the parity-check matrix used or its associated Tanner graph. We say that a parity-check matrix is *peeling-decodable* if every recoverable set of erasures by an ML decoder can also be recovered by the peeling algorithm.

Constructions of peeling-decodable parity-check matrices for an (N, K) linear code are presented in [14]– [16] where the number of rows of the constructed matrices is exponential in $N - K$. For such matrices, the peeling algorithm may cease to be appealing if $N - K$ is large. As a motivation of our investigation of codes with parity-check matrices of column weight two we give the following result, the proof of which is presented in Appendix A.

Theorem 1. *Let $\mathbf{H}$ be a parity-check matrix of a linear code in which each column has weight at most two. Then, $\mathbf{H}$ is peeling-decodable.*

B. QC-LDPC Codes and Their Parity-Check Matrices

Throughout this paper, we use $(x)_t$ for an integer x and a positive integer t to denote the least nonnegative integer congruent to x modulo t, i.e., $(x)_t = x - \lfloor x/t \rfloor t$. All indices of vectors and of rows and columns of matrices are numbered starting with 0.

By an $m \times n$ array $\mathbf{H} = [\mathbf{H}_{i,j}]_{0\leq i<m, 0\leq j<n}$ of $t \times t$ matrices $\mathbf{H}_{i,j}$ we mean the $mt \times nt$ matrix in which the (I, J) entry in $\mathbf{H}$, $0 \leq I < mt$, $0 \leq J < nt$, is the (i', j') entry in $\mathbf{H}_{i,j}$ where $i' = (I)_t$, $j' = (J)_t$, $i = \lfloor I/t \rfloor$, and $j = \lfloor J/t \rfloor$. In general, we use (I, J), $0 \leq I < mt, 0 \leq J < nt$, to denote indices of entries in the $mt \times nt$ matrix $\mathbf{H}$, (i', j'), $0 \leq i', j' < t$, to denote indices of entries in a $t \times t$ submatrix, and (i, j), $0 \leq i < m, 0 \leq j < n$, to denote the indices of the submatrix within the array $\mathbf{H}$. For $0 \leq i < m$, the $t \times nt$ submatrix $[\mathbf{H}_{i,0}, \mathbf{H}_{i,1}, \mathbf{H}_{i,1}, \ldots, \mathbf{H}_{i,n-1}]$ of $\mathbf{H}$ is called the i-th *row block* and for $0 \leq j < n$, the $mt \times t$ submatrix $[\mathbf{H}_{0,j}^{\mathsf{T}}, \mathbf{H}_{1,j}^{\mathsf{T}}, \ldots, \mathbf{H}_{m-1,j}^{\mathsf{T}}]^{\mathsf{T}}$ is called the j-th *column block*. For $0 \leq i < m$ and $0 \leq i' < t$, a row in $\mathbf{H}$ is indexed by $(i; i')$ if it is the i'-th row in the i-th row block. Thus, a row in $\mathbf{H}$ can be indexed by I for some I, $0 \leq I < mt$, or by the pair $(i; i')$, $0 \leq i < m$, $0 \leq i' < t$, where $i' = (I)_t$, $i = \lfloor I/t \rfloor$, and $I = it + i'$. Similarly, a column in $\mathbf{H}$ can be indexed by J for some J, $0 \leq J < nt$, or by the pair $(j; j')$, $0 \leq j < n$, $0 \leq j' < t$, where $j' = (J)_t$, $j = \lfloor J/t \rfloor$, and $J = jt + j'$, indicating the j'-th column in the j-th column block.

A *circulant* is a square matrix in which every row other than the top row is the cyclic shift of the row above it by one position to the right. It follows that the top row is also the cyclic shift of the bottom row. Hence, a circulant is completely characterized by its top row. A binary $t \times t$ matrix is called a *circulant permutation matrix* (CPM) if its top row has weight one. A CPM in which the single 1 in its top row is in position p, $0 \leq p < t$, is denoted by $\mathbf{CPM}_t(p)$. Notice that all the entries in $\mathbf{CPM}_t(p)$ are zeros except those in positions $(i', (i'+p)_t)$ for $0 \leq i' < t$, i.e., positions $((j'-p)_t, j')$ for $0 \leq j' < t$. Suppose that $\mathbf{H}$ is an $m \times n$ array of $t \times t$ CPM's, i.e., $\mathbf{H} = [\mathbf{CPM}_t(p_{i,j})]_{0 \leq i < m, 0 \leq j < n}$. To capture the parameters of $\mathbf{H}$ we denote it by $\mathbf{H}_{m,n,t}$. The matrix $\mathbf{H}_{m,n,t}$ is specified by the mn numbers $p_{i,j}$, $0 \leq p_{i,j} < t$, $0 \leq i < m, 0 \leq j < n$.

A code is *quasi-cyclic* (QC) [12], [13] if it is the null space of an array of circulants of equal size. In particular, if $\mathbf{H}_{m,n,t} = [\mathbf{CPM}_t(p_{i,j})]_{0 \leq i < m, 0 \leq j < n}$, then it is a parity-check matrix of a QC code, $C_{m,n,t}$, of length nt and dimension $nt - \mathrm{rank}(\mathbf{H}_{m,n,t})$. Assuming that t is not small, then $\mathbf{H}_{m,n,t}$ is sparse and the code is a QC-LDPC code.

The composition of the parity-check matrix $\mathbf{H}_{m,n,t}$ as an array of circulants, naturally defines a *sectionalized* structure for codewords. A binary sequence $\mathbf{v} = (v_0, v_1, \ldots, v_{nt-1})$ composed of nt bits can be written as $\mathbf{v} = (\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_{n-1})$, where $\mathbf{v}_i = (v_{it}, v_{it+1}, \ldots, v_{it+t-1})$, $0 \leq i < n$. Erasures affecting only one section of the transmitted codeword form a *phased burst*. Thus, a phased burst may contain up to t erasures. If the number of erasures is t, then we say that the phased burst is *solid*. We say that two phased bursts affecting two sections are *mutually semi-solid* if the total number of erasures is $2t-1$, i.e., one phased burst is solid and the other contains $t-1$ erasures.

Let $e(r)$ be the maximum number of guaranteed correctable erasures by an ML decoder if the channel causes erasures confined to any $r \leq n$ sections. Clearly, $e(1) = t$ as the columns in any column block are linearly independent. We also have $e(n) = d - 1$, where d is the minimum Hamming distance of the code, as $e(n)$ is the maximum number of guaranteed correctable erasures that occur anywhere in a codeword.

By circularly shifting the columns in each column block and the rows in each row block of $\mathbf{H}_{m,n,t} = [\mathbf{CPM}_t(p_{i,j})]_{0 \leq i < m, 0 \leq j < n}$, we can put $\mathbf{H}_{m,n,t}$ in a form of an $m \times n$ array of CPMs in which the 0-th row block and the 0-th column block consist only of $t \times t$ identity matrices $\mathbf{CPM}_t(0)$. Such a form is called *canonical*. These shifting operations do not change the rank of the matrix $\mathbf{H}_{m,n,t}$ and, being confined to columns in the same column block, do not change the capability of $C_{m,n,t}$ to correct phased bursts using ML decoding or the peeling algorithm. Therefore, from now on, we only consider matrices $\mathbf{H}_{m,n,t}$ in canonical form.

Since the sum over GF(2) of the columns in any column block is the all-one vector, the sum over GF(2) of the columns in any two column blocks is the all-zero vector. We conclude that the columns in any two column blocks are linearly dependent. This implies that $e(r) \leq 2t-1$ for all $r \geq 2$. For $m = 1$, $\mathbf{H}_{1,n,t}$ is just a row of CPMs and $e(2) = 1$. Therefore, to have $e(2) > 1$, m should be at least two. We will see that the upper bound $2t-1$ on the number of erasures that can be corrected in a pair of phased bursts can be attained for $m = 2$. Since the code rate may decrease by increasing m, it is interesting to consider the case $m = 2$ which is treated in the next section.

III. QC Codes with Parity-Check Matrices Composed of Two Rows of CPMs

With $m = 2$, we consider a parity-check matrix, $\mathbf{H}_{2,n,t}$, in the canonical form

$$\begin{bmatrix} \mathbf{CPM}_t(0) & \mathbf{CPM}_t(0) & \cdots & \mathbf{CPM}_t(0) \\ \mathbf{CPM}_t(p_0) & \mathbf{CPM}_t(p_1) & \cdots & \mathbf{CPM}_t(p_{n-1}) \end{bmatrix}, \quad (1)$$

where $n \geq 2$ and $p_0 = 0$. For convenience, we call the two row blocks in $\mathbf{H}_{2,n,t}$ the top row block and the bottom row block. Then $\mathbf{H}_{2,n,t}$ is a parity-check matrix of a QC-LDPC code, $C_{2,n,t}$, of length nt and dimension $nt - \mathrm{rank}(\mathbf{H}_{2,n,t})$. The rank of $\mathbf{H}_{2,n,t}$, which equals the redundancy of $C_{2,n,t}$, is given in the following theorem. The proof is presented in Appendix B.

Theorem 2. *The rank of the matrix $\mathbf{H}_{2,n,t}$ in (1) equals* $\mathrm{rank}(\mathbf{H}_{2,n,t}) = 2t - \mathrm{GCD}(p_1, \ldots, p_{n-1}, t)$.

The following theorem, the proof of which is given in Appendix C, gives the phased burst erasure correcting capabilities, $e(r)$, of the code $C_{2,n,t}$. From Theorem 1, all these erasures are also correctable by the peeling algorithm.

Theorem 3. *For the code $C_{2,n,t}$ with the parity-check matrix $\mathbf{H}_{2,n,t}$ in (1), we have $e(1) = t$ and*

$$e(2) = \frac{2t}{\max\limits_{0 \leq j_0 < j_1 < n} \mathrm{GCD}(p_{j_1} - p_{j_0}, t)} - 1.$$

Furthermore, for $3 \leq r \leq n$, $e(r) = 1$ if p_j for $0 \leq j < n$, are not distinct. If they are distinct, then $e(r) = 3$ if $(p_{j_1} - p_{j_0})_t = (p_{j_2} - p_{j_3})_t$ for distinct pairs (j_0, j_1) and (j_3, j_2) such that $0 \leq j_0, j_1, j_2, j_3 < n$, $j_0 \neq j_1$, and $j_2 \neq j_3$ and, in case $r = 3$, not all j_0, j_1, j_2, j_3 are distinct. Finally, if p_j are distinct for $0 \leq j < n$ and there are no distinct pairs (j_0, j_1) and (j_3, j_2) for which $(p_{j_1} - p_{j_0})_t = (p_{j_2} - p_{j_3})_t$ such that $0 \leq j_0, j_1, j_2, j_3 < n$, $j_0 \neq j_1$, and $j_2 \neq j_3$ or such pairs exist in case $r = 3$ but for all of them j_0, j_1, j_2, j_3 are not distinct, then $e(r) = 5$.

Since the minimum Hamming distance of the code is $d = e(n) + 1$, it follows that d is either 2, 4, or 6. Gallager [17, Theorem 2.5] has shown that the minimum Hamming distances of codes, with parity-check matrices in which each column has weight two, grow at most logarithmically with the code length. Theorem 3 gives a much more pessimistic result in case the codes are quasi-cyclic. However, Theorem 1 implies that $\mathbf{H}_{2,n,t}$ is optimal for peeling. In particular, all erasures recoverable by an ML decoder, and not only those limited in number by minimum Hamming distance, are also

correctable by the peeling algorithm. We also notice from the proofs in Appendix C that the girth of the Tanner graph representing $\mathbf{H}_{2,n,t}$ is twice the minimum Hamming distance, i.e., it is 4, 8, or 12. The fact that the girth of a Tanner graph associated with a parity-check matrix composed of two row blocks of circulants is divisible by 4 was observed by Fossorier [19, Corollary 2.1] who has also shown that the girth is at most 12 [19, Corollary 2.5]. Chen, Bai, and Wang [18] gave necessary and sufficient conditions for the girth to be equal to 12.

As mentioned earlier, a linear (N, K) code is optimal for some erasures if these erasures are correctable by the code and the redundancy, $N - K$, equals the number of erasures. From Theorem 3, it is clear that $C_{2,n,t}$ has poor erasure correcting capability unless the erasures are confined to at most two sections. If t is a prime and p_j are distinct for $0 \leq j < n$, then Theorem 3 implies that $e(2) = 2t - 1$, which equals the redundancy as given in Theorem 2. In this case $C_{2,n,t}$ is optimal for any two mutually semi-solid phased bursts of erasures. No other code of the same length that can correct the same erasures has higher dimension.

Example 1. Consider the parity-check matrix $\mathbf{H}_{2,n,t}$ in (1) in which $t \geq n \geq 3$, and $p_j = j$ for $0 \leq j < n$. From Theorem 2, we have

$$\mathrm{rank}(\mathbf{H}_{2,n,t}) = 2t - \mathrm{GCD}(1, 2, \ldots, n-1, t) = 2t - 1.$$

Notice that p_j are distinct for $0 \leq j < n$ but $(p_{j_1} - p_{j_0})_t$ are not distinct for $0 \leq j_0 \neq j_1 < n$. Indeed, $(p_1 - p_0)_t = (p_2 - p_1)_t$ as both equal 1. Hence, from Theorem 3, we have $e(1) = t$,

$$e(2) = \frac{2t}{\max\limits_{0 \leq j_0 < j_1 < n} \mathrm{GCD}(j_0 - j_1, t)} - 1 = \frac{2t}{t_n} - 1,$$

and $e(r) = 3$ for $3 \leq r \leq n$, where t_n is the largest factor of t less than n. The code has minimum Hamming distance of four. If t is a prime, then $t_n = 1$ and the code can correct any two mutually semi-solid phased bursts of erasures and, in this case, it is optimal for these erasures. ∎

Example 2. Consider the parity-check matrix $\mathbf{H}_{2,n,t}$ in (1) in which $t = 2^\tau - 1$, $\tau \geq n \geq 3$, and $p_j = 2^j - 1$ for $0 \leq j < n$. From Theorem 2, we have

$$\mathrm{rank}(\mathbf{H}_{2,n,t}) = 2t - \mathrm{GCD}(1, 3, \ldots, 2^{n-1} - 1, t) = 2t - 1.$$

For $0 \leq j_0 \leq j_1 < n$,

$$\begin{aligned} \mathrm{GCD}(p_{j_1} - p_{j_0}, t) &= \mathrm{GCD}((2^{j_1} - 1) - (2^{j_0} - 1), t) \\ &= \mathrm{GCD}(2^{j_1 - j_0} - 1, 2^\tau - 1) \\ &= 2^{\mathrm{GCD}(j_1 - j_0, \tau)} - 1, \end{aligned}$$

where we used the well-known fact that $\mathrm{GCD}(x^a - 1, x^b - 1) = x^{\mathrm{GCD}(a,b)} - 1$ for nonnegative integers a and b. Notice that not only p_j for $0 \leq j < n$ are distinct, but also $(p_{j_1} - p_{j_0})_t$ are distinct for $0 \leq j_0 \neq j_1 < n$. Indeed, suppose that $(p_{j_1} - p_{j_0})_t = (p_{j_2} - p_{j_3})_t$ for distinct pairs (j_0, j_1) and (j_3, j_2) such that $0 \leq j_0, j_1, j_2, j_3 < n$, $j_0 \neq j_1$, and $j_2 \neq j_3$. Then $2^{j_3} - 2^{j_2} + 2^{j_1} - 2^{j_0}$ is divisible by t. Since $-2^n + 2 \leq 2^{j_3} - 2^{j_2} + 2^{j_1} - 2^{j_0} \leq 2^n - 2$ and $t \geq 2^n - 1$, it follows that $2^{j_3} - 2^{j_2} + 2^{j_1} - 2^{j_0} = 0$. Without loss of generality, assume that $j_3 \geq j_0, j_1, j_2$. Since $2^{j_3} > 2^{j_3 - 1} + 2^{j_3 - 2} + \cdots + 1$, we conclude that $j_2 = j_3$ and $j_1 = j_0$ or $j_0 = j_3$ and $j_1 = j_2$. Both cases contradict the conditions imposed on the two pairs. From Theorem 3, we have $e(1) = t$,

$$e(2) = \frac{2t}{\max\limits_{0 \leq j_0 < j_1 < n} 2^{\mathrm{GCD}(j_1 - j_0, \tau) - 1}} - 1 = \frac{2t}{2^{\tau_n} - 1} - 1,$$

and $e(r) = 5$ for $3 \leq r \leq n$, where τ_n is the largest factor of τ less than n. The code has minimum Hamming distance of six. If τ is a prime, then $\tau_n = 1$ and the code can correct any two mutually semi-solid phased bursts of erasures and, in this case, it is optimal for these erasures. ∎

Recall that a collection of integers $p_0, p_1, \ldots, p_{n-1}$ forms a *t-modular Golomb ruler* [20] if $(p_{j_1} - p_{j_0})_t$ are nonzero and distinct for $0 \leq j_0 \neq j_1 < n$. This means that for every positive integer less than t, there is at most one pair of i and j such that $(p_i - p_j)_t$ equals this integer[1]. Theorem 3 implies that the minimum Hamming distance is at most six with equality if and only if $p_0, p_1, \ldots, p_{n-1}$ form a t-modular Golomb ruler as in Example 2 where the size of the circulants, t, is exponential in the number, n, of circulants in a row block. Although this leads to a simple construction of a matrix satisfying the modular Golomb ruler property, such an exponential growth of t as a function of n may not be desirable. Fortunately, this need not be the case. Let $t_{\min}(n)$ be the minimum value of t such that there are n nonnegative integers $p_0 = 0, p_1, \ldots, p_{n-1}$ less than t that form a t-modular Golomb ruler. This function has been studied extensively, see e.g., [20]. It is stated in [26] that

$$n^2 - n + 1 \leq t_{\min}(n) \leq n^2 + O(n^{36/23}),$$

which shows that quadratic growth in n is sufficient. Constructions of t-modular Golomb rulers are due to Singer [27] ($n = p + 1$, $t = p^2 + p + 1$, p is a prime power), Bose [28] ($n = p$, $t = p^2 - 1$, p is a prime power), and Ruzsa [29] ($n = p - 1$, $t = p(p - 1)$, p is a prime).

Although the code $C_{2,n,t}$ cannot correct two solid phased bursts of $2t$ erasures, if it can correct any two mutually semi-solid phased bursts of erasures, then it is easy to come up with a subcode, C, of $C_{2,n,t}$ that can correct any two solid phased bursts of erasures. Since any vector of weight $2t$ in which all its 1's are confined to two sections is in the null space of $\mathbf{H}_{2,n,t}$, a parity-check matrix, $\mathbf{H}$, of C can be obtained by augmenting $\mathbf{H}_{2,n,t}$ with a matrix that does not have any such vector in its null space. Hence, the sums over GF(2) of the columns in each column block in the augmenting matrix should be distinct. Therefore, the number of rows in the augmenting matrix is at least $\lceil \log_2(n) \rceil$. A possible choice for such matrix with that many rows is to have the $(j; 0)$

[1] In case $(p_i - p_j)_t$ is replaced by $(p_i + p_j)_t$, the sequence is a t-modular Sidon sequence [21] while if the difference sign is kept but "at most" is replaced by "exactly", the modular Golomb ruler is a perfect difference set [20, Section 19.3]. These combinatorial objects and variations thereof were used in numerous papers, e.g., [18], [22]– [25], to construct LDPC codes with Tanner graphs of large girths.

column to be the binary representation of j, $0 \leq j < n$, and all other columns to be all-zero columns. Notice that the augmenting matrix is not composed of CPMs and the code C is not quasi-cyclic. For $0 \leq j < n$, the $(j;0)$ columns in the augmenting matrix are all distinct and hence for any pair of such columns there is a row with a 1 in exactly one of the two columns. Hence, if the channel causes two solid phased bursts of erasures, then there is a parity-check that can be used to recover one of the erased bits. Again, the remaining erasures form two mutually semi-solid phased bursts which are within the correcting capability of $C_{2,n,t}$. In particular, the peeling algorithm applied to the augmented parity-check matrix can correct any two solid phased bursts of erasures although it is not optimal for such erasures. However, subject to the restriction that it is a subcode of $C_{2,n,t}$, its redundancy is minimum among all codes that can correct two solid phased bursts. Without this restriction, if $n \leq 2^t$, one can construct a code which is optimal for any two solid phased bursts of erasures. Indeed, a (possibly shortened or lengthened) Reed-Solomon code of length n and dimension $n-2$ over GF(2^t) in which each symbol is represented by a binary vector of length t is optimal for pairs of solid phased bursts of erasures. The issue with this construction is that if the length of the section, t, is large, e.g., in the hundreds, then decoding is prohibitive as it involves computations over GF(2^t).

IV. Conclusion

In this paper, we investigated the erasure correction capabilities of QC-LDPC codes with parity-check matricies composed of two rows of CPMs. We completely determined the dimensions and minimum Hamming distances of these codes as well as the girth of the Tanner graphs associated with their parity-check matrices. In spite of their poor minimum Hamming distances, these codes have good correcting capabilities for phased erasures confined to two sections of a codeword. In particular, we have shown that there are codes among this class of codes that are optimal for any two mutually semi-solid phased bursts of erasures. These codes may find applications in distributed data storage.

Appendix A
Proof of Theorem 1

Suppose $\mathbf{H}$ is not peeling-decodable. Then, there is a nonempty set, $\mathcal{J}$, of variable nodes in $\mathcal{G}$ that forms a stopping set such that the columns of $\mathbf{H}_{\mathcal{J}}$ are linearly independent. Let $\mathcal{I}$ be the set of check nodes in the subgraph $\mathcal{G}(\mathcal{J})$ of $\mathcal{G}$ induced by $\mathcal{J}$. As every column in $\mathbf{H}_{\mathcal{J}}$ has weight at most two, the number of edges incident on $\mathcal{J}$ is at most $2|\mathcal{J}|$. Since $\mathcal{J}$ forms a stopping set, every check node in $\mathcal{I}$ is adjacent to at least two variable nodes in $\mathcal{J}$. Hence, the number of edges incident on these check nodes is at least $2|\mathcal{I}|$. As the edges in $\mathcal{G}(\mathcal{J})$ incident on $\mathcal{I}$ are the same as those incident on $\mathcal{J}$, we have $2|\mathcal{J}| \geq 2|\mathcal{I}|$. Since the columns of $\mathbf{H}_{\mathcal{J}}$ are linearly independent, we have $|\mathcal{J}| \leq |\mathcal{I}|$. We conclude that $|\mathcal{I}| = |\mathcal{J}|$ and every node in $\mathcal{I}$ or $\mathcal{J}$ is incident on exactly two edges. This is equivalent to saying that every row in $\mathbf{H}_{\mathcal{J}}$ has weight two, contradicting the assumption that the columns indexed by $\mathcal{J}$ are linearly independent. ∎

Appendix B
Proof of Theorem 2

We start with the following lemma which gives the rank of an array of circulants that are not necessarily CPMs. The proof is based on Bézout's identity which states that given polynomials $a_1(x), \ldots, a_n(x)$ over some field with greatest common divisor (GCD) $f(x)$, there exist polynomials $q_1(x), \ldots, q_n(x)$ such that $f(x) = q_1(x)a_1(x) + \cdots + q_n(x)a_n(x)$, see e.g., [30, Corollary 1.37].

Lemma 1. *Let $\mathbf{A}_1, \ldots, \mathbf{A}_{n-1}$ be $t \times t$ circulants over some field and $\mathbf{A} = [\mathbf{A}_1, \ldots, \mathbf{A}_{n-1}]$. For $1 \leq j < n$, let $\mathbf{a}_j = (a_{0,j}, a_{1,j}, \ldots, a_{t-1,j})$ be the top row of $\mathbf{A}_j$ and $a_j(x) = a_{0,j} + a_{1,j}x + \cdots + a_{t-1,j}x^{t-1}$. Then,* $\mathrm{rank}(\mathbf{A}) = t - \deg(f(x))$ *where $f(x) = \mathrm{GCD}(a_1(x), \ldots, a_{n-1}(x), x^t - 1)$.*

Proof: Let $f(x) = \sum_{j=0}^{t-1} f_i x^j$ and define the sequence $\mathbf{f} = (f_0, f_1, \ldots, f_{t-1})$. Then, with $a_n(x) = x^t - 1$, Bézout's identity implies that

$$f(x) \equiv q_1(x)a_1(x) + \cdots + q_{n-1}(x)a_{n-1}(x) \pmod{x^t - 1}$$

for some polynomials $q_1(x), \ldots, q_{n-1}(x)$. Hence, $\mathbf{f}$ is a linear combination of $\mathbf{a}_1, \ldots, \mathbf{a}_{n-1}$ and their cyclic shifts, i.e., it is in the column space of the matrix $\mathbf{A}$. Notice that $\mathbf{f}$ ends with $t - \deg(f(x)) - 1$ zeros. Hence, $\mathbf{f}$ and its $t - \deg(f(x)) - 1$ cyclic shifts are linearly independent. Since $\mathbf{f}$ is in the column space of $\mathbf{A}$ which is a row of circulants, all cyclic shifts of $\mathbf{f}$ are also in the same column space. Thus, $\mathbf{A}$ has rank at least $t - \deg(f(x))$ as it contains that many linearly independent vectors. To show that the rank of $\mathbf{A}$ does not exceed $t - \deg(f(x))$, we argue that every vector $\mathbf{s} = (s_0, s_1, \ldots, s_{t-1})$ in the column space of $\mathbf{A}$ is a linear combination of these $t - \deg(f(x))$ linearly independent vectors. Indeed, let $s(x) = \sum_{i=0}^{t-1} s_i x^i$. Then, as $\mathbf{s}$ is in the column space of $\mathbf{A}$, it is a linear combination of $\mathbf{a}_1, \ldots, \mathbf{a}_{n-1}$ and their $t-1$ cyclic shifts. In particular, for some polynomials $u_1(x), \ldots, u_{n-1}(x)$, we have

$$s(x) \equiv u_1(x)a_1(x) + \cdots + u_{n-1}(x)a_{n-1}(x) \pmod{x^t - 1}.$$

Since $f(x) = \mathrm{GCD}(a_1(x), \ldots, a_{n-1}(x), x^t - 1)$, it follows that $f(x)$ divides $s(x)$, i.e., $s(x) = q(x)f(x)$ for some polynomial $q(x)$ of degree less than $t - \deg(f(x))$. This is equivalent to saying that $\mathbf{s}$ is a linear combination of $\mathbf{f}$ and its $n - \deg(f(x)) - 1$ cyclic shifts. ∎

To complete the proof of Theorem 2, we subtract the top row block of $\mathbf{H}_{2,n,t}$ as given in (1) from the bottom row block to obtain the matrix

$$\mathbf{H}' = \begin{bmatrix} \mathbf{CPM}_t(0) & \mathbf{CPM}_t(0) & \cdots & \mathbf{CPM}_t(0) \\ \mathbf{0} & \mathbf{A}_1 & \cdots & \mathbf{A}_{n-1} \end{bmatrix},$$

where $\mathbf{0}$ is the $t \times t$ all-zero matrix and $\mathbf{A}_j = \mathbf{CPM}_t(p_j) - \mathbf{CPM}_t(0)$. For $1 \leq j < n$, the matrix $\mathbf{A}_j$ is a circulant in which its top row is either the all-zero vector or has exactly two 1's at positions p_j and 0. Since $\mathbf{H}'$ is obtained from $\mathbf{H}_{2,n,t}$ by elementary row operations, they have the same rank.

Furthermore, as $\mathbf{CPM}_t(0)$, being an identity matrix, has rank t, we have

$$\operatorname{rank}(\mathbf{H}_{2,n,t}) = \operatorname{rank}(\mathbf{H}') = t + \operatorname{rank}(\mathbf{A}), \tag{2}$$

where $\mathbf{A} = [\mathbf{A}_1, \ldots, \mathbf{A}_{n-1}]$ is composed of $n-1$ circulants. We invoke Lemma 1 to find the rank of this matrix. For this purpose, let $a_j(x) = x^{p_j} - 1$ for $1 \le j < n$. Then,

$$\begin{aligned} f(x) &= \operatorname{GCD}(a_1(x), \ldots, a_{n-1}(x), x^t - 1) \\ &= \operatorname{GCD}(x^{p_1} - 1, \ldots, x^{p_{n-1}} - 1, x^t - 1) \\ &= x^{\operatorname{GCD}(p_1 \ldots, p_{n-1}, t)} - 1. \end{aligned}$$

The result now follows directly from Lemma 1 and (2). ∎

Appendix C
Proof of Theorem 3

Since the t columns in any column block are linearly independent, $e(1) = t$. As any r column blocks, where $2 \le r \le n$, have linearly dependent columns, $e(r)$ is one less than the minimum number of linearly dependent columns confined to r column blocks. Since each column in $\mathbf{H}_{2,n,t}$ has a single 1 in the top row block and a single 1 in the bottom row block, only an even number of columns in $\mathbf{H}_{2,n,t}$ can sum up to the all-zero vector and, therefore, $e(r)$ is odd for $2 \le r \le n$. The following lemma relates linear dependence of columns in $\mathbf{H}_{2,n,t}$ to cycles in the Tanner graph $\mathcal{G}$ representing $\mathbf{H}_{2,n,t}$.

Lemma 2. *The columns of $\mathbf{H}_{2,n,t}$ indexed by a nonempty set $\mathcal{J}$ of indices are linearly dependent if and only if there is a cycle in the subgraph, $\mathcal{G}(\mathcal{J})$, of $\mathcal{G}$ induced by $\mathcal{J}$.*

Proof: The columns indexed by $\mathcal{J}$ are linearly dependent if and only if there is a nonempty subset $\mathcal{J}_0 \subseteq \mathcal{J}$ of indices of columns that sum over GF(2) to the all-zero vector such that no proper nonempty subset of $\mathcal{J}_0$ has this property. This is the case if and only if every row in the matrix $\mathbf{H}_{\mathcal{J}_0}$, the submatrix of $\mathbf{H}_{2,n,t}$ composed of the columns indexed by $\mathcal{J}_0$, has weight two which holds if and only if the nodes in the subgraph $\mathcal{G}(\mathcal{J}_0)$ form a cycle. ∎

Without loss of generality, we can assume that a cycle in $\mathcal{G}$ starts with the variable node $(j_0; j'_0)$ followed by a check node in the top row block followed by the variable node $(j_1; j'_0)$ followed by a check node in the bottom row block and so on until it reaches a variable node $(j_{e-1}; j'_{e-1})$ followed by a check node in the bottom row block and finally ends at the variable node $(j_e; j'_e) = (j_0; j'_0)$ we started with. Based on this, the cycle can be completely specified by the sequence $(j_0; j'_0), (j_1; j'_1), \ldots, (j_{e-1}; j'_{e-1})$ of variable nodes without listing the check nodes or the ending variable node which is the same as the starting node. The length of the cycle is $2e$. For such a sequence to form a cycle it is necessary that

1) $j_\ell \ne j_{\ell+1}$ for $0 \le \ell < e$ where $j_e = j_0$ as no check node is adjacent to two variable nodes in the same column block;
2) If ℓ is even, then $j'_\ell = j'_{\ell+1}$ for the variables nodes $(j_\ell; j'_\ell)$ and $(j_{\ell+1}; j'_{\ell+1})$ to be adjacent to a check node in the top row block;
3) If ℓ is odd, then $(j'_\ell - p_{j_\ell})_t = (j'_{\ell+1} - p_{j_{\ell+1}})_t$, where $(j_e; j'_e) = (j_0; j'_0)$, for the variables nodes $(j_\ell; j'_\ell)$ and $(j_{\ell+1}; j'_{\ell+1})$ to be adjacent to a check node in the bottom row block.

Combined with the condition that $(j_0; j'_0), (j_1; j'_1), \ldots, (j_{e-1}; j'_{e-1})$ are distinct gives a necessary and sufficient condition for the sequence to form a cycle. If this extra condition is not met, then the sequence represents a closed walk that contains a cycle of length less than $2e$.

To determine $e(2)$, we consider the minimum number of linearly dependent columns confined to the column blocks j_0 and j_1, where $0 \le j_0 \ne j_1 < n$. From Lemma 2, there are e such columns only if there is a sequence $(j_0; j'_0), (j_1; j'_1), \ldots, (j_{e-1}; j'_{e-1})$ of variable nodes satisfying conditions 1), 2), and 3). Then, for even ℓ we have $j_\ell = j_0$ and $j'_\ell = j'_{\ell+1}$ while for odd ℓ we have $j_\ell = j_1$ and $(j'_\ell - p_{j_\ell})_t = (j'_{\ell+1} - p_{j_{\ell+1}})_t$. Summing over $\ell = 0, 1, \ldots, e-1$, we get $\frac{1}{2}(p_{j_1} - p_{j_0})e \equiv 0 \pmod t$. The minimum value of e for this congruency to hold is $2t/\operatorname{GCD}(p_{j_1} - p_{j_0}, t)$. Hence, there is no cycle of length less than $2e$ with $e = 2t/\operatorname{GCD}(p_{j_1} - p_{j_0}, t)$ involving only variable nodes confined to the column blocks j_0 and j_1. For such e, we can find a closed walk of length $2e$. Indeed, let $(j_\ell; j'_\ell) = (j_0; ((p_{j_0} - p_{j_1})\frac{\ell}{2})_t)$ if $\ell = 0, 2, \ldots, e$ and $(j_\ell; j'_\ell) = (j_1; ((p_{j_0} - p_{j_1})\frac{\ell-1}{2})_t)$ if $\ell = 1, 3, \ldots, e-1$. Then, $(j_e; j'_e) = (j_0; j'_0)$ and the three conditions 1), 2), 3) hold. We conclude that the length of a shortest cycle of variable nodes confined to the column blocks j_0 and j_1 is $2e$ and e is the minimum number of linearly dependent columns confined to these column blocks. From this, the expression of $e(2)$ follows.

Next, we consider $e(r)$ in case $3 \le r \le n$. If $p_{j_0} = p_{j_1}$, where $0 \le j_0 \ne j_1 < n$, then the j'-th column, $0 \le j' < t$, in the j_0-th column block is the same as the j'-th column in the j_1-th column block and $e(r) = 1$. In the following, we assume that $p_0, p_1, \ldots, p_{n-1}$ are distinct. Then no two columns in $\mathbf{H}_{2,n,t}$ are identical and $e(r) > 1$ which implies that $e(r) \ge 3$. Suppose $(p_{j_1} - p_{j_0})_t = (p_{j_2} - p_{j_3})_t$ for distinct pairs (j_0, j_1) and (j_3, j_2) such that $0 \le j_0, j_1, j_2, j_3 < n$, $j_0 \ne j_1$, and $j_2 \ne j_3$. Then $j_1 \ne j_2$ otherwise $p_{j_0} = p_{j_3}$ which implies that $j_0 = j_3$ as $p_0, p_1, \ldots, p_{n-1}$ are distinct. Similarly, $j_3 \ne j_0$. The sequence of variable nodes $(j_0; 0), (j_1; 0), (j_2; (p_{j_2} - p_{j_1})_t), (j_3; (p_{j_2} - p_{j_1})_t)$ satisfies conditions 1), 2), and 3) and forms a cycle of length eight. From Lemma 2, the columns of $\mathbf{H}_{2,n,t}$ indexed by these four variable nodes are linearly dependent. Hence, $e(r) \le 3$ if the four variable nodes are confined to r column blocks. Otherwise, if there are no such pairs (j_0, j_1) and (j_3, j_2) confined to r column blocks, then $e(r) > 3$ which implies that $e(r) \ge 5$.

Finally, consider the sequence of the six variable nodes $(j_0; 0), (j_1; 0), (j_2; (p_{j_2} - p_{j_1})_t), (j_0; (p_{j_2} - p_{j_1})_t), (j_1; (p_{j_2} - p_{j_0})_t), (j_2; (p_{j_2} - p_{j_0})_t)$, where $0 \le j_0 < j_1 < j_2 < n$. This sequence satisfies conditions 1), 2), and 3). Hence, we conclude from Lemma 2 that the columns of $\mathbf{H}_{2,n,t}$ indexed by these six variable nodes which are confined to three column blocks are linearly dependent. This proves that $e(r) \le 5$ for all $3 \le r \le n$. ∎

ACKNOWLEDGMENT

The work of B. Vasić is funded in part by the NSF under grants NSF ECCS-1500170 and NSF SaTC-1813401.

REFERENCES

[1] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman "Efficient erasure correcting codes," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 569–584, Feb. 2001.

[2] M. Yang and W. E. Ryan, "Performance of efficiently encodable low-density parity-check codes in noise bursts on the EPR4 channel," *IEEE Trans. Magn.*, vol. 40, no. 2, pp. 507–512, Mar. 2004.

[3] G. Hosoya, H. Yagi, T. Matsushima, and S. Hirasawa, "A modification method for constructing low-density parity-check codes for burst erasures," *ICICE Trans. Fundamentals*, vol. E89-A, no. 10, pp. 2501–2509, Oct. 2006.

[4] Y. Y. Tai, L. Lan, L. Zeng, S. Lin, and K. A. S. Abdel-Ghaffar, "Algebraic construction of quasi-cyclic LDPC codes for the AWGN and erasure channels," *IEEE Trans. Commun.*, vol. 54, pp. 1765–1774, Oct. 2006.

[5] S. J. Johnson, "Burst erasure correcting LDPC codes," *IEEE Trans. Commun.*, vol. 57, no. 3, pp. 641–652, Mar. 2009.

[6] K. Li, A. Kavčić, and M. F. Erden, "Construction of burst-erasure efficient LDPC codes for use with belief propagation decoding," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Cape Town, South Africa, May 23–27, 2010, pp. 1–5

[7] X. Ge and S. -T. Xia, "Structured non-binary LDPC codes with large girth," *Electron. Lett.*, vol. 43, no. 22, pp. 1220–1221, Oct. 2007.

[8] C. Di, D. Proietti, I. E. Telatar, T. J. Richardson, and R. L. Urbanke, "Finite-length analysis of low-density parity-check codes on the binary erasure channel," *IEEE Trans. Inf. Theory*, vol. 48, no. 6, pp. 1570–1579, Jun. 2002.

[9] H. Pishro-Nik and F. Fekri, "On decoding of low-density parity-check codes over the binary erasure channel," *IEEE Trans. Inf. Theory*, vol. 50, no. 3, pp. 439–454, Mar. 2004.

[10] V. Savin, "LDPC decoders," in *Channel Coding: Theory, Algorithms, and Applications*, D. Declercq, M. Fossorier, and E. Biglieri Eds., Oxford, UK: Academic Press, 2014.

[11] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inf. Theory*, vol. 27, no. 5, pp. 533 – 547, Sep. 1981.

[12] S. Lin and D. J. Costello, Jr., *Error Control Coding: Fundamentals and Applications*, 2nd ed. Upper Saddle River, NJ: Prentice Hall, 2004.

[13] W. E. Ryan and S. Lin, *Channel Codes: Classical and Modern*. New York, NY: Cambridge University Press, 2009.

[14] H. D. L. Hollmann and L. M. G. M. Tolhuizen, "Generic erasure correcting sets: bounds and constructions," *J. Combin. Theory, Ser. A*, vol. 113, no. 8, pp. 1746–1759, Nov. 2006.

[15] H. D. L. Hollmann and L. M. G. M. Tolhuizen, "On parity check collections for iterative erasure decoding that correct all correctable erasure patterns of a given size," *IEEE Trans. Inf. Theory*, vol. 53, no. 2, pp. 823–828, Feb. 2007.

[16] J. H. Weber and K. A. S. Abdel-Ghaffar, "Results on parity-check matrices with optimal stopping and/or dead-end set enumerators," *IEEE Trans. Inf. Theory*, vol. 54, no. 3, pp. 1368–1374, Mar. 2008.

[17] R. G. Gallager, *Low Density Parity Check Codes*. Cambridge, MA: MIT Press, 1963.

[18] C. Chen, B. Bai, and X. Wang. "Construction of nonbinary quasi-cyclic LDPC cycle codes based on Singer perfect difference set," *IEEE Commun. Lett.*, vol. 14, no. 2, pp. 181–183, Feb. 2010.

[19] M. P. C. Fossorier, "Quasi-cyclic low-density parity-check codes from circulant permutation matrices," *IEEE Trans. Inf. Theory*, vol. 50, no. 8, pp. 1788–1793, Aug. 2004.

[20] C. J. Colbourn and J. H. Dinitz, *Handbook of Combinatorial Designs*, 2nd ed. CRC Press: Boca Raton, FL, 2007.

[21] K. O'Bryant, "A complete annotated bibliography of work related to Sidon sequences," *Electron. J. Combin.*, vol. DS11, pp. 1–39, Jul. 2004.

[22] B. Vasic and O. Milenkovic, "Combinatorial constructions of low-density parity-check codes for iterative decoding," *IEEE Trans. Inf. Theory*, vol. 50, no. 6, pp. 1156–1176, Jun. 2004.

[23] M. Esmaeili and M. Javedankherad, "4-cycle free LDPC codes based on difference sets," *IEEE Trans. Commun.*, vol. 60, no. 12, pp. 3579–3586, Dec. 2012.

[24] G. Zhang, R. Sun, and X. Wang "New quasi-cyclic LDPC codes with girth at least eight based on Sidon sequences," in *Proc. Int. Symp. Turbo Codes and Iterative Information Processing (ISTC)*, Gothenburg, Sweden, Aug. 27–31, 2012, pp. 31–35.

[25] H. Park, S. Hong, J.-S. No, and D. -J. Shin, "Construction of high-rate regular quasi-cyclic LDPC codes based on cyclic difference families," *IEEE Trans. Commun.*, vol. 61, no. 8, pp. 3108–3113, Aug. 2013.

[26] R. L. Grahams and N. J. A. Sloane, "On additive bases and harmonious graphs," *SIAM J. Alg. Disc. Meth.*, vol. 1, no. 4, Dec. 1980.

[27] J. Singer, "A theorem in finite projective geometry and some applications to number theory," *Trans. Amer. Math. Soc.*, vol. 43, no. 3, pp. 377–385, May 1938.

[28] R. C. Bose, "An affine analogue of Singer's theorem," *J. Indian Math. Soc.*, vol. 6, pp. 1–15, 1942.

[29] I. Z. Ruzsa, "Solving a linear equation in a set of integers I," *Acta Arith.*, vol. 65, no. 3, 259–282, 1993.

[30] P. A. Fuhrmann, *A Polynomial Approach to Linear Algebra*, 2nd ed. New York, NY: Springer, 2012.