Design of Non-Binary Quasi-Cyclic LDPC Codes by Absorbing Set Removal

Behzad Amiri
Electrical Eng. Department
University of California, Los Angeles
Los Angeles, USA
Email: amiri@ucla.edu

Jorge Arturo Flores Castro
Instituto Nacional de Astrofísica
Óptica y Electrónica (INAOE)
Puebla, México
Email: arturo.flores@inaoep.mx

Lara Dolecek
Electrical Eng. Department
University of California, Los Angeles
Los Angeles, USA
Email: dolecek@ee.ucla.edu

***Abstract*—Non-binary quasi-cyclic (NB-QC) codes are a class of graph-based codes with high performance and implementation-friendly structure. In this paper, we introduce a new method for designing NB-QC codes with improved performance in the low error-rate region. Specifically, we propose a construction which reduces the number of non-binary absorbing sets, which are known to cause errors when decoding non-binary LDPC codes. Our construction is based on a careful selection of the code design parameters. Simulation results demonstrate the superior performance of codes designed according to our technique compared to existing state-of-the-art NB-QC codes.**

I. Introduction

Non-binary quasi-cyclic (NB-QC) LDPC codes are an important class of non-binary codes defined on graphs. NB-QC codes offer superior error-correcting performance and implementation friendly structure. They are well suited for emerging data storage applications requiring very low error rates. Design of NB-QC codes is an active area of research; some recent constructions include [1], [2] and [3].

Under message-passing decoding, certain non-codewords compete with the codewords to be the output of the decoder. The existence of these non-codeword objects significantly undermines the performance of (non-binary) graph-based codes. It was shown in [4] by Poulliat, Fosorrier and Declercq that short cycles in the Tanner graph of a non-binary LDPC (NB-LDPC) code, which satisfy certain weight conditions result in non-codeword errors. Inspired by the (early version of the) results in [4], Peng and Chen constructed NB-QC codes by utilizing a cycle elimination algorithm which attempts to remove all cycles shorter than a certain length [5]. Due to the fact that not all cycles in the Tanner graph may be problematic, Bazarsky, Presman, and Litsyn in [6] restrict the cancellation of cycles only to those that violate a given set of ACE spectrum constraints [7].

We showed in our recent work that decoding errors in the output of NB-LDPC can be combinatorially described by certain substructures in the Tanner graph of these codes. We refer to these substructures as *non-binary absorbing sets*. In this paper, by applying insights from [8], we present an approach towards constructing high-performance NB-QC codes. In particular, we eliminate a collection of non-binary absorbing sets (NB ASs) by optimizing the construction parameters that specify lifting and labeling operations. This is achieved by canceling only a single cycle in each of the ASs of interest. Therefore, compared to cycle-only approaches, which try to cancel all cycles that violate certain conditions as in [5], [6], our approach is capable of canceling more ASs of various sizes. Simulation results show the effectiveness of our code design approach compared to state-of-the-art NB-QC codes: the performance improvement is more than one order of magnitude for the codes over GF(4).

This paper is organized as follows. In Section II, we review the construction of NB-QC codes using lifting along with the definition of NB-ASs. In Section III, we first analyze how the conditions in the definition of non-binary absorbing sets map to the choice of parameters in the design steps for NB-QC codes. We then present our method to design NB-QC codes with a reduced number of problematic absorbing sets. Section IV includes our simulation results which demonstrate the superior performance of our codes in the error floor region. Section V concludes the paper.

II. Preliminaries and Background

In this section, the construction of NB-QC codes using the lifting approach is presented. We also revisit the definition of NB-ASs. We introduce the notation necessary for the discussion in our work.

A. Construction of NB-QC codes using lifting

Assume that the following parameters are given:

- n: the number of columns in the binary base matrix,
- m: the number of rows in the binary base matrix,
- z: the lifting factor,
- d_v: the weight of each column in the binary base matrix,
- d_c: the weight of each row in the binary base matrix.

The construction of a (d_v, d_c)-regular NB-QC code over GF(q) using lifting involves the following steps [6]:

1) **Choosing the protograph**: The construction starts with the choice of an $m \times n$ binary parity-check matrix H with column weight d_v and row weight d_c. The parity-check matrix H can be equivalently represented by a bipartite graph $G = (V, C, F)$ called the Tanner graph, with the usual notation of V being the set of variable nodes $v_i, i \in \{1, \ldots, n\}$, C being the set of check nodes $c_j, j \in \{1, \ldots, m\}$, and the set

F describing the edges between the nodes in V and C. The Tanner graph G is called the protograph of the NB-QC code.

2) **Lifting the protograph**: The lifted matrix $\hat{H}$ is constructed by replacing each entry in matrix H with a $z \times z$ matrix. The zero entries in H are replaced by $z \times z$ zero matrices. Each non-zero entry corresponding to edge e in the protograph is replaced by a (z, d_e) circular permutation matrix (CPM). Here, (z, d_e) CPM refers to the $z \times z$ binary matrix obtained by circularly shifting the rows of the identity matrix by d_e places. Throughout the paper, d_e is called the lifting parameter associated with the edge e. The corresponding Tanner graph $\hat{G}$ of matrix $\hat{H}$ is called the binary lifted graph.

3) **Edge weight assignment**: In this step, non-binary weights are assigned to the edges of the binary lifted graph. Let α be a primitive element of GF(q). We choose a parameter λ such that $(q-1)|\lambda z$. We also select a parameter $\rho_e \in \{0, 1, \ldots, q-2\}$ for each edge e in the protograph G. Then, the value of the non-zero element in the kth row of the (z, d_e) CPM is replaced by $\alpha^{\rho_e + (k-1)\times\lambda}$. Throughout the paper, we refer to ρ_e as the labeling parameter associated with the edge e.

Note that the resulting NB-QC code has the length of $nz \log_2 q$ bits and the design rate of $\frac{n-m}{n}$. The construction introduced in [1] is a special case of the NB-QC construction using lifting when $z = q-1$, $\lambda = 1$ and $d_e = \rho_e$ for any edge e. The code designs in [2] and [9] are also special cases of the NB-QC construction using lifting.

B. Non-binary absorbing sets

In this subsection, we review the definition of NB-ASs, that are the non-codewords known to compete with the codewords to be the output of a NB-LDPC decoder [8].

Consider $\mathcal{V}$, with $|\mathcal{V}| = a$, as a subset of variable nodes in the Tanner graph corresponding to a parity-check matrix $\tilde{H}$ over GF(q). We form the $\ell \times a$ matrix A consisting of the columns in $\tilde{H}$ that correspond to the variable nodes in $\mathcal{V}$. The rows of matrix A correspond to the ℓ check nodes connected to the variable nodes in $\mathcal{V}$.

Definition 1. *([8]) The subset $\mathcal{V}$ of variable nodes is an (a, b) absorbing set $A_{a,b}$ over GF(q) if there exists an $(\ell - b) \times a$ submatrix B, with elements $b_{j,i}, 1 \le j \le \ell - b, 1 \le i \le a$, in matrix A that satisfies the following conditions:*

1)
$$\exists \boldsymbol{x} \in N(B) \text{ s.t. } x_i \neq 0 \text{ for } \forall i \in \{1, \ldots, a\} \text{and } \nexists i, \boldsymbol{d}_i \boldsymbol{x}^T = 0, \quad (1)$$

where $N(B)$ is the null-space of matrix B and $\boldsymbol{d}_i, 1 \le i \le b$ is the ith row of matrix D, where D is formed by excluding the matrix B from A.

2)
$$\forall i \in \{1, 2, \ldots, a\} : \left(\sum_{j=1}^{\ell-b} S(b_{j,i}) \right) > \left(\sum_{j=1}^{b} S(d_{j,i}) \right), \quad (2)$$

where $d_{j,i}$, $1 \le j \le b, 1 \le i \le a$ are the elements of the matrix D and the function S is
$$S(y) = \begin{cases} 1 \text{ when } y > 0, \\ 0 \text{ when } y \le 0. \end{cases} \quad (3)$$

Note that a binary AS is a special case of the above definition when the field size q is 2.

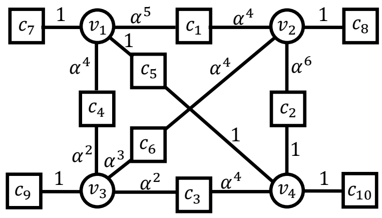

Fig. 1. A non-binary $(4, 4)$ AS over GF(8) based on the primitive polynomial $p(x) = x^3 + x + 1$ whose root is α. Circles indicate variable nodes, and squares indicate check nodes.

Example 1. *Consider the set of variable nodes $\mathcal{V} = \{v_1, v_2, v_3, v_4\}$ and their $\ell = 10$ connected check nodes, as shown in Figure 1. The 10×4 matrix A corresponding to the structure in Figure 1 is formed as*
$$A = \begin{bmatrix} \alpha^5 & 0 & 0 & \alpha^4 & 1 & 0 & 1 & 0 & 0 & 0 \\ \alpha^4 & \alpha^6 & 0 & 0 & 0 & \alpha^4 & 0 & 1 & 0 & 0 \\ 0 & 0 & \alpha^2 & \alpha^2 & 0 & \alpha^3 & 0 & 0 & 1 & 0 \\ 0 & 1 & \alpha^4 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}^T . \quad (4)$$
Consider that the first 6 rows of matrix A form the 6×4 submatrix B. For the input $\boldsymbol{x} = (1, \alpha, \alpha^2, 1)$ (mapped to (v_1, v_2, v_3, v_4) in the figure), the first condition in the Definition 1 is satisfied since $\boldsymbol{x} \in N(B)$ and the 4 check nodes, c_7, c_8, c_9, c_{10}, are not satisfied. Furthermore, for the given input $\boldsymbol{x}$, each variable node in $\mathcal{V}$ is connected to three satisfied check nodes and one unsatisfied check node. Therefore, the second condition in Definition 1 is also satisfied. As a result, for the the given input $\boldsymbol{x}$, the structure in Figure 1 is a $(4, 4)$ NB-AS over GF(8).

An AS $\mathcal{V}$ over GF(q) is called *elementary* if all neighboring satisfied check nodes have degree two and all neighboring unsatisfied check nodes have degree one with respect to $\mathcal{V}$, [8]. Elementary ASs are typically the most detrimental. In the case of elementary ASs over GF(2), i.e., binary elementary ASs, satisfied check nodes have even degrees and unsatisfied check nodes have odd degrees. As a result, a binary AS is elementary if and only if all its check nodes have degrees less than or equal to two.

As shown in [8], in the case of non-binary elementary ASs, the conditions in Definition 1 can be simplified. Let C_p be an arbitrary cycle involving p distinct variable nodes and p distinct neighboring check nodes in the Tanner graph of a given code. We write C_p as the oriented traversal $c_1 - v_1 - c_2 - v_2 - \cdots - c_p - v_p - c_1$, where v and c denote the spanned variable and check nodes, respectively. Let w_{2k-1}, $k \in \{1, \ldots, p\}$ denote the non-binary weight on the $c_k - v_k$ edge, and let w_{2k}, $k \in \{1, \ldots, p\}$ denote the weight on the $v_k - c_{k+1}$ edge.

Lemma 1. *([8]) A subset of variable nodes $\mathcal{V}$ is an elementary AS over GF(q) if and only if:*

1) *(Topological condition) For the induced subgraph corresponding to $\mathcal{V}$ and its neighboring check nodes, unlabeling of all edges (converting all edge weights to one) results in a binary elementary AS.*
2) *(Weight condition) For every cycle C_p (as described above), the weights of the edges w_i, $i \in \{1, 2, \ldots, 2p\}$,*

satisfy the following equation:

$$\prod_{k=1}^{p} w_{2k-1} = \prod_{k=1}^{p} w_{2k} \mod q. \tag{5}$$

In order to avoid non-binary elementary ASs, it is necessary to violate at least one of the two conditions in Lemma 1.

Example 2. *Observe that the configuration in Figure 1 can be interpreted as a* $(4,4)$ *non-binary elementary AS over GF(*8*). First, after removing the edge weights, the resulting subgraph is a binary AS with all its check nodes having degree one or two. Therefore, the unlabeled subgraph (the subgraph without the edge weights) is a binary elementary AS. Second, the edge weights in each of the six cycles in the configuration satisfy Equation* (5)*. As an example, the edge weights in the cycle spanning variable nodes* v_1*,* v_2*, and* v_3 *satisfy:*

$$\alpha^5 \cdot \alpha^4 \cdot \alpha^2 \equiv \alpha^4 \cdot \alpha^3 \cdot \alpha^4 \mod q, \tag{6}$$

where α *specifies the primitive element of GF(*8*).*

III. Design of NB-QC Codes Based on Absorbing Set Elimination

In this section, we first analyze the necessary conditions for the existence of a non-binary elementary AS in the Tanner graph of a NB-QC code. We investigate how the topological and weight conditions (introduced in Lemma 1) map to certain equations which include the design parameters of NB-QC codes. This analysis enables us to propose an algorithm to design NB-QC codes with good error floor performance. In our proposed algorithm, we design NB-QC codes with reduced number of problematic elementary ASs by violating either the topological or the weight conditions.

A. Topological and weight conditions in NB-QC codes

According to condition 1 in Lemma 1, an unlabeled non-binary elementary AS is a binary elementary AS. Since each binary AS is formed by a collection of cycles in the Tanner graph, we first focus our analysis on a single cycle in the protograph G defined in Section II. The following lemma identifies the relationship between a cycle in protograph G and its corresponding cycle(s) in the binary lifted graph $\hat{G}$.

Lemma 2. *([6]) Consider a cycle* C_p *involving* p *distinct variable nodes in the protograph with edges* $e_1, e_2, \ldots, e_{2p}$*. After lifting,* C_p *results in* z *cycles of the same length in the binary lifted graph if and only if the lifting parameters* d_{e_i}*,* $i \in \{1, \ldots, 2p\}$ *associated with the edges involved in* C_p *satisfy the following condition:*

$$\sum_{i=1}^{p} d_{e_{2i-1}} = \sum_{i=1}^{p} d_{e_{2i}} \mod z. \tag{7}$$

Otherwise, C_p *results in one or more cycles of larger lengths.*

Corollary 1. *Consider an* (a,b) *binary AS* $A_{a,b}$ *in the protograph* G*. After lifting* G *by the factor* z *to produce* $\hat{G}$*,* $A_{a,b}$ *results in* z *ASs of the same size in* $\hat{G}$ *if for every cycle* C_p *of length* $2p$ *in* $A_{a,b}$*, the lifting parameters* d_{e_i}*,* $i \in \{1, 2, \ldots, 2p\}$ *associated with edges of* C_p *satisfy* (7)*.*

Based on Corollary 1, we can prevent the existence of absorbing sets in $\hat{G}$ by ensuring that for at least one cycle C_p of AS $A_{a,b}$ in G, the lifting parameters d_{e_i} do not satisfy (7).

The weight condition in Lemma 1 implies that a non-binary elementary AS not only satisfies the topological condition, i.e., the unlabeled subgraph is a binary elementary AS, but also the edge weights in all of its cycles satisfy (5).

We study how the weight condition of non-binary elementary ASs maps to the NB-QC code construction. To analyze the weight condition, we first consider a single cycle.

Lemma 3. *([5]) Consider that a cycle* C_p *in protograph* G *with edges* $\{e_1, e_2, \ldots, e_{2p}\}$ *results in* z *cycles of the same length in the binary lifted graph* $\hat{G}$*. The* z *copies of* C_p *satisfy the weight condition in* (5) *if the labeling parameters* ρ_{e_i}*,* $i \in \{1, \ldots, 2p\}$ *associated with the edges involved in* C_p *satisfy the following condition:*

$$\sum_{i=1}^{p} \rho_{e_{2i-1}} = \sum_{i=1}^{p} \rho_{e_{2i}} \mod q-1. \tag{8}$$

Based on Lemma 1, the edge weights of all the cycles in a non-binary elementary AS satisfy the weight condition in (5). Note that an elementary AS typically consists of more than one cycle. Therefore, Lemma 3 implies the following corollary.

Corollary 2. *Consider that an elementary AS* $A_{a,b}$ *in the protograph* G *results in* z *binary elementary absorbing sets of the same size in* $\hat{G}$*. After edge weight assignment, the* z *copies of* $A_{a,b}$ *result in* z *non-binary elementary absorbing sets if for each cycle in* $A_{a,b}$*, the labeling parameters satisfy* (8)*.*

The above corollary offers an approach to avoid non-binary elementary ASs in the edge weight assignment step of the NB-QC code design. For a binary elementary AS present in the binary lifted graph $\hat{G}$, the labeling parameters should be chosen such that the weight condition is not satisfied for at least one cycle in the elementary absorbing set.

B. An algorithm to avoid absorbing sets in NB-QC codes

In this section, we propose an algorithm to design NB-QC codes with an improved error floor performance. The main idea is to avoid non-binary elementary ASs in the Tanner graph of the designed NB-QC code. Based on the discussion in Section III-A, an elementary AS $A_{a,b}$ in G results in z non-binary elementary ASs if the lifting and the labeling parameters associated with the edges in $A_{a,b}$ satisfy (7) and (8). In design approach, by informed selection of the lifting and the labeling parameters, we ensure that for each binary AS in G, at least one cycle does not satisfy either (7) or (8).

The inputs to the algorithm are: 1) A binary protograph G which determines the design rate, column weight and row weight of the code; 2) The finite field size, q, of the resulting NB-QC code; 3) The lifting factor, z.

The method is stated in Algorithm 1. We first construct a random NB-QC code with random assignment of the lifting and labeling parameters to all the edges in the protograph G. Based on the parameters of G, we first choose the set W of pairwise parameters (i.e., $W =$

$\{(a_1,b_1),(a_2,b_2),\ldots,(a_k,b_k)\}$ corresponding to the k elementary ASs which we wish to avoid[1].

We then find the smallest AS (a,b) in W and form the set U which includes all binary (a,b) ASs in protograph G. For each AS in U, we determine if all its cycles satisfy (7). If they do and if it is possible, we change the lifting parameters of the edges to ensure that at least one cycle in the AS does not satisfy (7). For each AS which is not avoided by the choice of lifting parameters, we determine if all the cycles satisfy (8) or not. If yes, the labeling parameter associated with an edge will be changed to ensure that at least one cycle of the AS does not satisfy (8). The new labeling parameter is chosen such that the previously canceled NB-ASs remain canceled. This process continues until either all ASs are canceled or no more ASs can be canceled.

Note that step 6 in Algorithm 1, where we find all (a,b) absorbing sets in the given photograph G, dominates the computational complexity of our proposed algorithm. Although it is proven that it is NP-complete to exhaustively find small error-prone substructures (stopping sets and trapping sets) in LDPC codes [10], several papers, such as [10] and [11], have proposed algorithms to reduce the computational complexity.

Remark 1. *The work in [8] presents an approach to cancel NB-ASs in unstructured NB-LDPC codes by changing the edge weights in the given Tanner graph. In the unstructured case, the value of each edge weight can be changed to any arbitrary non-zero value from GF(q). In contrast, for the NB-QC codes, the edge weights must satisfy certain conditions to ensure that the quasi-cyclic property is preserved. As described in Section II-A, to satisfy the quasi-cyclic construction over GF(q), the value of each non-zero element in the kth row of the* (z,d_e) *CPM should be equal to* $\alpha^{\rho_e+(k-1)\times\lambda}$*, where parameter* λ *is chosen such that* $(q-1)|\lambda z$ *and parameter* $\rho_e\in\{0,1,\ldots,q-2\}$ *is selected for each edge* e *in the protograph G. The approach presented here preserves the NB-QC structure by carefully modifying the lifting and labeling parameters during the design process.*

Note that our algorithm can be used to design both regular and irregular NB-QC codes. In the case of irregular NB-QC codes, the protograph G is an irregular Tanner graph.

IV. Simulation Results

In this section, we present the results of our simulations for different NB-QC codes[2]. We report bit error rate (BER) figures to compare the performance of our designed codes with other state-of-the-art NB-QC codes. We also present the error profiles of the decoder for different code constructions which explain the superior performance of our designed codes. The following is the list of the code constructions that we consider:

1) **Random construction**: For the given protograph, we randomly assign the lifting and labeling parameters to each edge.

[1]Note that the parameters of G, such as the column weight and girth determine the ASs available in G. For example, $(4,4)$ ASs are possible only when $d_v=4$. For other choices of d_v, $(4,4)$ ASs do not exist in G.

[2]We have performed additional simulations and have observed similar results for other choices of code parameters (blocklength, code rate, and column weight).

Algorithm 1 Design of NB-QC codes with reduced number of non-binary elementary ASs.

```
1:  Inputs: Protograph G, field size q and lifting factor z.
2:  Randomly assign a d_e ∈ {0,1,...,z−1} and a ρ_e ∈ {0,1,...,q−2} to each edge e in G.
3:  Choose W, the set of all ASs to be canceled .
4:  Let C = ∅ be the set of ASs which can not be eliminated in the lifting process.
5:  for ∀(a,b) AS ∈ W do
6:      Find U, the set of all ASs of size (a,b) in G.
7:      for ∀S ∈ U do
8:          Let F_S be the set of all the cycles in S.
9:          If at least one of the cycles in F_S does not satisfy (7), go to the next AS S in U.
10:         Let E_S be the list of all the edges involved in F_S.
11:         Find an edge e in E_S such that there exist a d'_e ≠ d_e which guarantees that at least one of the cycles in F_S does not satisfy (7). If the value exists, replace d_e with d'_e and go to 7, else C ← C ∪ S.
12:     end for
13: end for
14: for ∀ absorbing set S ∈ C do
15:     Let F_S be the set of cycles of S.
16:     If at least one of the cycles in F_S does not satisfy (8), go to next absorbing set S in C.
17:     Let E_S be the list of all the edges involved in F_S.
18:     Find edge e in E_S such that there exist a ρ'_e ≠ ρ_e which guarantees that at least one of the cycles in F_S does not satisfy (8). If the value exists, replace ρ_e with ρ'_e.
19: end for
```

2) **ACE construction**: We compare our results with the method recently introduced in [6]. The algorithm in [6] has two steps. First, for each cycle in the given protograph G, the algorithm finds the ACE value, which is defined as $\sum_{v_i} d_{v_i}-2$, where d_{v_i} is the degree of the node v_i, and the summation is over all the variable nodes of the cycle. Then, it searches for cycles in G which their associated ACE value is greater than a bound which is given as an input to the algorithm. Then the algorithm attempts to eliminate them in the lifted graph $\hat{G}$, by properly choosing the lifting parameters. In the second step, the cycles in $\hat{G}$ that violate a non-binary ACE constraint are found. Then, the algorithm attempts to cancel these cycles by carefully choosing the labeling parameters.

3) **Absorbing set (AS) construction**: Based on our proposed approach stated in Algorithm 1, we first identify a list of problematic ASs to cancel. The algorithm attempts to cancel these ASs by informed selection of the lifting and labeling parameters. Note that unlike the ACE approach, we are able to cancel more problematic ASs by canceling one cycle per AS in the Tanner graph of the code.

Figure 2 shows the simulation results for the three different constructions over field sizes $q=4,8,16$, design rate $R=0.69$, column weight $d_v=4$ and row weight $d_c=13$, transmitted over a binary-input additive white Gaussian noise

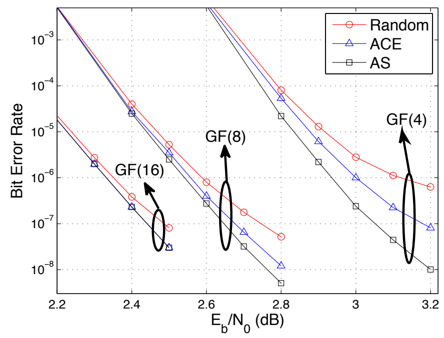

Fig. 2. Performance comparison for NB-QC codes, with blocklength $N = 1014$ for codes over GF(4), $N = 3549$ for codes over GF(8) and $N = 2028$ for codes over GF(16), rate $R = 0.69$, and row weight $d_v = 4$.

TABLE I
ERROR PROFILE FOR THE PERFORMANCE CURVES SHOWN IN FIGURE 2.

Error Type	$(6,4)$	$(6,6)$	$(7,4)$	$(8,2)$	$(8,4)$	$(9,4)$	$(10,4)$	other
GF(4), $N = 1014$ bits, SNR= 3.2 dB, $d_v = 4$								
Random	88	14	10	33	5	8	19	15
ACE	26	8	10	4	0	4	8	21
AS	0	0	0	0	0	0	0	23
GF(8), $N = 3549$ bits, SNR= 2.8 dB, $d_v = 4$								
Random	32	9	11	18	2	6	9	13
ACE	0	0	0	0	0	4	9	21
AS	0	0	0	0	0	0	0	29

(AWGN) channel. The set of curves over GF(4) has the following parameters: block length $N = 1014$ bits and lifting factor $z = 3$. The figure also includes the set of curves for GF(8) which have the following parameters: $N = 3549$ bits and lifting factor $z = 7$. For GF(16) set of curves, $N = 2028$ bits and lifting factor $z = 3$. Note that the ACE spectrum for the three ACE codes in Figure 2 is equal to $(\hat{\tau}_2^{(b)}, \hat{\tau}_4^{(b)}, \hat{\tau}_6^{(b)}, \hat{\tau}_8^{(b)}, \hat{\tau}_{10}^{(b)}) = (\infty, \infty, 6, 8, 10)$. Figure 2 shows that both ACE and AS approaches significantly improve the performance of the NB-QC code compared to the random approach. The AS approach achieves a better performance compared to the ACE approach since it focuses on canceling only one cycle per AS, whereas the ACE approach cancels all the cycles that violate the ACE constraints. Therefore, the AS approach is capable of removing more ASs. The performance comparison for different values of q and z reveals that the performance improvement for both ACE and AS approaches is more pronounced for smaller values of q and z, since for larger values of q and z, there are fewer number of non-binary absorbing sets to begin with (see [8] for more details). Table I includes the error profiles for GF(4) and GF(8) curves in Figure 2 at SNR= 3.2dB and SNR= 2.8dB, respectively. The table confirms that the AS approach cancels more problematic absorbing sets from the NB-QC code. The errors listed as 'other' in the table include 'oscillating' errors, in which the decoder oscillates between different errors in its last few iterations, and 'non-absorbing set' errors, in which the decoder converges to an error which is not an AS.

Figure 3 presents the simulation results for codes constructed by the three different approaches over field sizes $q = 4, 8$, design rate $R = 0.54$, column weight $d_v = 5$ and row weight $d_c = 11$. The constructed codes over GF(4) and GF(8) have $N = 726$ and $N = 2541$ bits, respectively. Note that the ACE spectrum for the three ACE codes in Figure 2

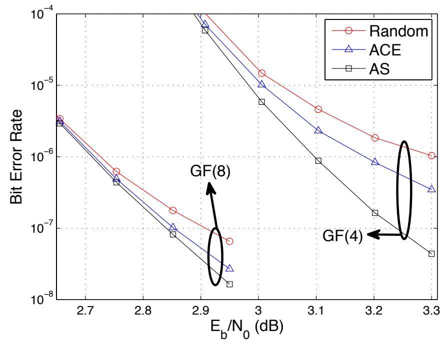

Fig. 3. Performance comparison for NB-QC codes, blocklength $N = 726$ for codes over GF(4) and $N = 2541$ for codes over GF(8), $R = 0.54$, $d_v = 5$.

TABLE II
ERROR PROFILE FOR THE PERFORMANCE CURVES SHOWN IN FIGURE 3.

Error Type	$(4,8)$	$(5,9)$	$(6,8)$	$(6,10)$	$(7,9)$	$(8,6)$	$(8,8)$	$(8,10)$	other
GF(4), $N = 726$ bits, SNR= 3.3 dB, $d_v = 5$									
Random	49	12	19	6	8	2	4	6	17
ACE	13	3	5	2	2	0	5	5	25
AS	0	0	0	0	0	0	0	0	31

is equal to $(\hat{\tau}_2^{(b)}, \hat{\tau}_4^{(b)}, \hat{\tau}_6^{(b)}, \hat{\tau}_8^{(b)}, \hat{\tau}_{10}^{(b)}) = (\infty, \infty, 9, 12, 15)$. Similar to Figure 2, both the ACE approach and the AS approach have significantly better performance in the error floor region compared to the random construction of an NB-QC code. Table II confirms the superior performance of the AS approach: there exist fewer ASs in the error profile of the code constructed by the AS approach.

V. CONCLUSION

In this paper, we first investigated the necessary conditions for the existence of a non-binary elementary AS in NB-QC codes. We then proposed an approach to choose the lifting and labeling parameters in the design of NB-QC codes to avoid problematic non-binary ASs. Our simulation results confirmed the effectiveness of our code design algorithm; our codes demonstrate superior error floor performance compared to some state-of-the-art constructions.

ACKNOWLEDGEMENT

This work was supported in part by UC-MEXUS grant and NSF-CCF grant 1162501.

REFERENCES

[1] L. Zeng *et al.*, "Constructions of nonbinary quasi-cyclic LDPC codes: a finite field approach," *IEEE Trans. Commun.*, Apr. 2008.
[2] J. Huang *et al.*, "Large-girth nonbinary QC-codes of various lengths," *IEEE Trans. Commun.*, Dec. 2010.
[3] B. Zhou *et al.*, "High performance non-binary quasi-cyclic LDPC codes on Euclidean geometries," *IEEE Trans. Commun.*, May 2009.
[4] C. Poulliat *et al.*, "Design of regular $(2, d_c)-$LDPC codes over GF(q) using their binary images," *IEEE Trans. Commun.*, Oct. 2008.
[5] R.-H. Peng and R. R. Chen, "Design of nonbinary quasi-cyclic LDPC cycle codes," in *Proc. IEEE ITW*, Tahoe City, CA Sep. 2007 .
[6] A. Bazarsky *et al.*, "Design of non-binary quasi-cyclic LDPC codes by ACE optimization," in *Proc. IEEE ITW*, Sevilla, Spain, Sep. 2013.
[7] D. Vukobratovic and V. Senk, "Generalized ACE constrained progressive edge-growth LDPC code design," *IEEE Commun. Lett.*, Jan. 2008.
[8] B. Amiri *et al.*, "Analysis and enumeration of absorbing sets for non-binary graph-based codes," *IEEE Trans. Commun.*, Feb. 2014.
[9] L. Zhang *et al.*, "Quasi-cyclic LDPC codes: An algebraic construction, rank analysis, and codes on Latin squares," *IEEE Trans. Commun.*, 2010.
[10] C-C. Wang *et al.*, "Finding all small error-prone substructures in LDPC codes," *IEEE Trans. Inform. Theory*, Apr. 2009.
[11] M. Karimi and A. H. Banihashemi, "An efficient algorithm for finding dominant trapping sets of LDPC codes," *IEEE Trans. Inform. Theory*, Nov. 2012.