

Toward an Algebraic Theory for Turbo Codes

R. Michael Tanner
Department of Computer Science
University of California, Santa Cruz
Santa Cruz, CA 95064
E-mail: tanner@cse.ucsc.edu

Abstract: *Instead of a random interleaver, an algebraic interleaver interconnecting two simple convolutional codes with feedback encoders forms an algebraic quasi-cyclic code. An (n_C, k) convolutional code in tailbiting form becomes a quasi-cyclic (QC) linear code of length Ln_C for some $L = Mn$. Interleavers with a period M connect two such codes to produce a QC turbo code. The QC turbo codes check equations are transformed and analyzed on an extension field of $GF(2)$ with an n th root of unity. Analysis of the QC turbo code gives insight into the minimum distance of the standard turbo code derived from it.*

Keywords: quasi-cyclic, tailbiting, transform, algebraic turbocode

1 Introduction

Turbo codes [2] [6] are typically defined using two or more simple convolutional codes, each with a feedback encoder, combined with a random interleaver. Theoretical analyses of turbo code performance have relied on the properties of random interleavers, which permit probabilistic analysis of the random ensemble of codes. For example, uniform random interleavers [1] can be coupled with analysis of input-output weight distributions for the convolutional codes, yielding bounds on the probability of error for turbo codes in an random ensemble [3]. Asymptotic analysis of the effectiveness of iterative decoding also exploits the independence assumptions that hold for asymptotically large block sizes with random interleavers [9] [4].

From a practical point of view, random interleavers are not attractive for several reasons. First, the entropy of the interleaver description is necessarily large, which requires memory to store the interleaver connections. Second, there is no a priori guarantee that a particular interleaver will be free of identifiable weaknesses for purposes of iterative decoding. In particular, two positions that are close in the initial ordering may also be close after permutation by the interleaver. This creates a short cycle that will cause an iterative decoder to recirculate statistically dependent information after a short number of cycles. Blatant weaknesses of this type can be

discovered by a spread test [5]. Third, there is no assurance that a given permutation will not produce some number of unacceptably low weight words in the turbo code.

Reducing the memory required alone motivates the use of algebraic interleavers which have a short mathematical description [6][pp. 53 -61]. In addition, an algebraically defined interleaver for a particular block length can be tested for the spread it ensures, and it also gives the turbo code some structure that may simplify analysis or testing. The purpose of this paper is to build a meaningful bridge between the rich algebraic understandings that have led to design of large minimum distance block codes, and the problem of interleaver design and convolutional code selection for turbo codes. Eventually it should be possible to design algebraic turbo codes with strong minimum distance and better performance at high signal to noise ratios than are found using random interleavers.

In [13] algebraic designs are presented for the *repeat and accumulate* (RA) codes of Divsalar, Jin, and McEliece [3]. A periodic algebraic interleaver combined with a tailbiting rate one accumulator code creates a quasi-cyclic (QC) code. The derived RA code has performance comparable or superior to random interleaver codes of the same rate and block length. Simulation results for one code dropped beneath the theoretical error floor for the ensemble of random interleaver codes of closely comparable rate and length.

This paper develops an analogous algebraic framework for higher memory order parallel concatenated systems. First a binary convolution code is converted into a QC code by tailbiting after Ln_C bits are transmitted, where $L = Mn$ is deliberately chosen to be a composite and n is odd. For the conversion the convolutional code is defined by its syndrome former, which naturally gives rise to a code constraint (Tanner) graph for the QC code. The QC code consists of n_C cycles of length L . Each of these cycles can be decomposed into M cycles of length n . A parity check matrix for this QC code is then transformed to a system of equations over an extension field of $GF(2)$ containing a primitive n th root of unity. The tailbiting code is put in parallel concatenation with a

quasi-cyclic permutation interleaver, a periodic block interleaver with period M . With the periodicity constraint on the interleaver, the parallel turbo code is still a QC code consisting of cycles of size n . Finally, the tailbiting circle of the QC turbo code can be severed to form a standard turbo code. After minor rate adjustment to ensure a zero state at the point of separation, the properties of the final turbo code can be tightly tied to those of the QC turbo code. In particular, the free distance of the modified turbo code is at least as large as the minimum distance of the concatenated QC code.

2 Tailbiting binary convolutional codes and syndrome formers

Convolutional codes that have proven to be effective for the creation of turbo codes are typically described as the output sequences of a feedback systematic convolutional encoder with a relatively small memory. Feedback in the encoder realizes a mapping from the information sequence to the parity checks that avoids trivial low weight words when the encoder is one component of a turbo code construction. When the same information is input to two or more systematic encoders, with interleaving, obviously the redundant copies of the information sequence can be suppressed in the composite turbo code word.

Following the notation of McEliece [8], in general an (n_C, k) convolutional code can be defined by a generator matrix $G(D)$ over $F(D)$, the rational subfield of the field of all one-sided formal Laurent series with coefficient over a symbol field F . With turbo code applications in mind, we will restrict attention to systematic generators $G(D)$ which contain a $k \times k$ identity matrix. For convenience, we implicitly assume that the degree m of the code, the sum of the Forney indices, is small (typically less than six), and the number of states required for the encoder, 2^m , is correspondingly small.

The code \mathcal{C} generated by $G(D)$ can also be defined by its dual code. If \mathcal{C} is an (n_C, k, m) code, then its dual, \mathcal{C}^\perp is an $(n_C, n_C - k, m)$ code with a generator matrix $H(D)$ [8](p.1108-9). The parity check matrix, also called the syndrome former matrix, $H(D)$ is an $(n_C - k) \times n_C$ matrix over $F(D)$ of rank $n_C - k$ whose rows are orthogonal to the codewords in \mathcal{C} . When $G(D)$ is a systematic generator, it is particularly easy to find a generator for the dual. If $G(D) = [I_k, A]$, $H(D) = [-A^T, I_{n_C - k}]$ will suffice, I_j being the $j \times j$ identity matrix. Any convolutional code has a *polynomial* generator matrix (PGM), and we will assume that $H(D)$ is a PGM. For consistency, we will adopt the convention of keeping the information position of the systematic code on the left.

The convolutional parity check matrix $H(D)$ chosen induces an infinite code constraint (Tanner) graph

[10] [14] [15]. Each of the n_C encoded sequences output from the convolutional encoder can be represented as an infinite sequence of bit nodes. In the systematic form, k of them are information sequences. At each time t , a row of $H(D)$ imposes a parity check equation on a subset of the code sequence nodes that must be satisfied, and each of the $n_C - k$ rows constitutes an infinite sequence of parity check equations operating on the n_C encoded sequences.

For the convolutional codes of interest for turbo codes, these concepts can be easily depicted. Figure 1 shows a systematic rate one-half feedback convolutional encoder with generator $G(D) = [1, h(D)/g(D)]$, with $h(D) = 1 + D^2 + D^3$, $g(D) = 1 + D + D^3$. The denominator $g(D)$ polynomial determines the feedback. Encoded sequences are $X(D) = [x_1(D), x_2(D)] = u(D)G(D) = [u(D), u(D)h(D)/g(D)]$. The syndrome former PGM is simply $H(D) = [h(D), g(D)]$, which gives a sequence of parity check equation nodes as shown in the code graph. Observe that the check nodes realize the equations $x_1(D)h(D) + x_2(D)g(D) = 0$. In the standard convolutional application, the starting state of the encoder is taken to be zero, and so the edges of parity check nodes that would extend to bit nodes for $t < 0$ are shown as dashed lines. The equations must be solved by the non-zero bits at nodes for $t \geq 0$. From the parity check equation $H(D)X(D)^T = 0$, it is obvious that $X(D) = [g(D), h(D)]$ is a weight six solution to the equations. This solution is shown with the six non-zero bit nodes lightly shaded. This 8-state code will be used a illustrative example throughout this paper.

A convolutional code with syndrome former $H(D)$ can be turned into a tailbiting convolutional code or equivalently, a quasi-cyclic block code of length Ln_C by taking the defining parity equation $H(D)X(D)^T = 0$ to be on the ring of polynomials *modulo* $(D^L - 1)$ [11]. This has the effect of turning the n_C infinite sequences of nodes into n_C finite sets or circles of L nodes in which the parity equations are applied with cyclic wrap around. In the graph representation, if the nodes in each sequence are indexed by t , a connections to a node with index $t \geq L$ is made instead to the node of the same sequence with index $t \bmod L$. To emphasize that the code is being viewed as a quasi-cyclic code, we will change the indeterminate from D to the conventional x of block coding theory. The convolutional equations $H(D)X(D)^T = 0$ become the quasi-cyclic code equations $H(x)X(x)^T = 0 \bmod x^L - 1 = 0$. It is immediately obvious that if $X(D)$ is a codeword in the convolutional code, $X(x)$ is a word in the quasi-cyclic code, because the equations $H(D)X(D)^T = 0$ will still be satisfied when they are reduced *modulo* $D^L - 1 = 0$. $X(x)$ is the image of the convolutional word obtained by adding for each sequence all the values at positions $t = i + jL$, $0 \leq i \leq (L - 1), j \geq 0$.

From another perspective, the QC code contains all sequences constituting the encoder response to input sequences $U(D)$ that are periodic with period L with encoder state sequences $S(D)$ that are similarly periodic. For a systematic feedback encoder, there may exist non-zero periodic solutions with a periodic input information sequence that is zero. Consequently the code rate of the QC code may be higher than that of the convolutional code restricted to start and end in the zero state. It is also easy to see that that if $X(x)$ is a non-zero polynomial QC codeword that starts and ends in the zero state, then $X(x)|_{x=D}$ is a convolutional word.

Analysis of the QC code gives insight into the convolutional code. If

$X(x) = X(D)|_{D=x} \bmod (x^L - 1)$ is non-zero, its weight must be smaller than the weight of $X(D)$.

Proposition: The free distance, d_{free} , of the convolutional code satisfies

$$d_{free} \geq \min(d_{min}, \min(wt(X(D)))) \quad (1)$$

where d_{min} is the minimum distance of the QC code and the minimum in the argument is taken over all non-zero $X(D)$ such that $X(D)|_{D=x} = 0 \bmod (x^L - 1)$.

Proof: If $X(x)$ is non-zero, it is a word in the QC code, and each position in $X(x)$ is the sum of mutually exclusive distinct positions of $X(D)$. $X(x)$ is a word in the QC code, and its weight must be greater than d_{min} . \diamond

The matrix $H(D)$ is also the template for the parity check equations for the QC code. Let $H(D) = [h_{ij}(D)]$ be an $n_C - k \times n_C$ polynomial generator matrix for \mathcal{C}^\perp so that $H(D)X(D)^T = H(D)[x_1(D), x_2(D), \dots, x_{n_C}(D)]^T = 0$ defines the convolutional code. In the QC code obtained by reduction $\bmod (D^L - 1)$, each sequence of the convolutional code is replaced by a vector

$\mathbf{x}_i = [x_{i,1}, x_{i,2}, \dots, x_{i,L-1}]$ with associated polynomial $x_i(x) = \sum_{l=1}^{L-1} x_{i,l}x^l$. The QC equations are $H(x)X(x)^T = H(x)[x_1(x), x_2(x), \dots, x_{n_C}(x)]^T = 0$. The binary parity check matrix for the QC code can be interpreted as a block matrix of circulants. Let \mathbf{h}_{ij} be an $L \times L$ binary circulant whose *first column* is the vector of length L defined by the polynomial $h_{ij}(x)$. (We will assume that the degree of $h_{ij}(D)$ is less than L and therefore it is not necessary to say $h_{ij}(x) \bmod (x^L - 1)$.) Let \mathbf{H} be the $(n_C - k) \times n_C$ block matrix of $L \times L$ binary circulants, $\mathbf{H} = [\mathbf{h}_{ij}]$. The QC code equations can then be written as $\mathbf{H}[\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{n_C}]^T = 0$.

Figure 2 shows the QC parity check matrix and the code graph for the code of Fig. 1 with $L=7$. The information positions are on the left, the redundant check bits on the right.

3 Transforms and minimum distance of the QC tailbiting code

The potential theoretical advantage of converting the convolutional code into a tailbiting QC code is that it may be possible to employ algebraic arguments to bound the minimum distance of the QC code. Fourier transform methods are applicable to any set of code equations that can be described by circulant matrices over the symbol field of the code. As developed in [12], the rows of a Fourier transform matrix are eigenvectors of a circulant, and a set of binary equations expressed as $L \times L$ circulant matrices can be transformed to a set of equations over an extension field containing an L th root of unity.

Delving into the details of possible algebraic arguments is not the purpose of this paper. We will illustrate the style of analysis with a quick proof that the free distance of the code of Fig. 1 is six.

The QC code formed by tailbiting with $L = 7$ has a double circulant parity check matrix \mathbf{H} specified by $H(x) = [h_0(x), h_1(x)] = [1 + x^2 + x^3, 1 + x + x^3]$. Let α be a primitive seventh root of unity on $GF(8)$, $\alpha^3 + \alpha + 1 = 0$ its minimum polynomial. The i th row of the Fourier transform matrix is $\mathbf{f}_i = [1 \ \alpha^i \ \alpha^{2i} \ \dots \ \alpha^{6i}]$. Multiplying \mathbf{H} from the left,

$$\mathbf{f}_i \mathbf{H} = [h_0(\alpha^i), h_1(\alpha^i)] \otimes \mathbf{f}_i. \quad (2)$$

The eigenvalues evaluate to $[h_0(1), h_1(1)] = [1, 1]$, $[h_0(\alpha), h_1(\alpha)] = [\alpha^4, 0]$, and $[h_0(\alpha^6), h_1(\alpha^6)] = [0, \alpha]$, and the other values are determined by conjugacy constraints. The rank of the equations is seven. From these equations one can conclude that if $[\mathbf{x}_0 \ \mathbf{x}_1]$ is a word in the QC code, and both halves are non-zero, both have a spectral zero that ensures that each has weight at least three. If only one half is non-zero, it must also have $\alpha^0 = 1$ as a root, implying its weight is four. This is the key algebraic argument for this code. The minimum distance of the QC code can be bounded using proof techniques that are adapted from cyclic coding theory.

Applying (1), this establishes that a convolutional word either reduces to the zero word in the QC code or its weight is at least four. To complete the argument for the convolutional code, we observe that the convolutional codeword must be non-zero in both information and parity sequences, so a convolutional word that has a reduced QC image with weight zero or four must have weight six or greater. If $X(D)|_{D=x} = 0 \bmod (x^7 - 1)$, $U(D) = a(D)(D^7 - 1)$, and $U(D)G(D) = [U(D), a(D)(D^7 - 1)/(1 + D + D^3)] = [U(D), a(D)(1 + D + D^2 + D^4)]$. Then $wt(U(D)) < 4$ implies $a(D) = (D^{7j} - 1)/(D^7 - 1)$ for some $j \geq 1$, so the weight of $X(D)$ is a least six.

This simple code exhibits important phenomena encountered in mapping a convolutional code to a

QC. While a convolutional codeword must be non-zero in both the information and the parity sequences, this QC code has non-zero words that are restricted to one or the other. In terms of the general analysis of [12], the equations in an eigenspace may allow solutions that are restricted to a subset of the output sequences. But conversely, there may not exist a basis for the QC code space of vectors that are non-zero in just the “information” sequences. The rate of the QC code must be at least as large as that of the convolutional code, because the rank of the QC parity check matrix is

$$\text{rank}(\mathbf{H}) = \sum_{i=0}^{(N-1)} \text{rank}(H(D)|_{D=\alpha^i}) \quad (3)$$

and $\text{rank}(H(\alpha^i)) \leq \text{rank}(H(D))$ for each i . While the information set may not be the same, its relative size is at least as large.

4 Decomposing the QC code into smaller blocks

Our goal is the formulation of the equations for the QC tailbiting code with length Ln_C so that a periodic interleaver between two such codes creates an “algebraic QC turbo code,” a turbo code configuration for two tailbiting convolutional codes that satisfies algebraic parity check equations. Describing interleavers that are non-trivial requires the QC to be decomposed into smaller pieces, breaking each large circulant into a number of shorter circulants.

The QC code obtained from the convolutional code by tailbiting after L transmissions has a parity check matrix \mathbf{H} consisting of $L \times L$ circulants. We purposefully choose $L = Mn$ to be a composite, and in this case, each of the $L \times L$ circulants can be broken down into an $M \times M$ block matrix of $n \times n$ circulants. Apply a permutation matrix to the rows and columns of \mathbf{H} that takes the successive $(i + jM)$ th row and column positions, $0 \leq j \leq (M - 1)$, of one of the circulants to form multiple $n \times n$ matrices. The permutation required is that of a classical block interleaver [6] (pp. 38-9) with M columns and n rows, which is a “raster scan” reordering. When \mathbf{A} is an $L \times L$ circulant, this permutation of positions produces \mathbf{A}_n , which is an $M \times M$ block matrix of $n \times n$ circulants. Applying the permutation simultaneously to each of the n_C transmitted sequences, the parity check matrix \mathbf{H} is permuted into what we will define to be \mathbf{H}_n . \mathbf{H}_n is the parity check matrix an equivalent QC with Mn_C circles of n bits.

An alternative viewpoint on this recasting of the QC tailbiting code is that the original convolutional code is undergoing *blocking* in blocks of size M [8]. The convolutional code with generator $G(D)$ is broken into size M blocks to produce an (Mn_C, Mk, m)

convolutional code, and then tailbiting occurs as described above after n transmissions.

For the convolutional codes of particular interest for the construction of turbo codes, the effect of the blocking permutation can be characterized reasonably simply. Suppose \mathbf{A} is an $L \times L$ circulant is defined by the polynomial $a(x)$ in its first column, with $a(x) = 1 + a_1x^1 + \dots + a_dx^d$ of degree d . \mathbf{A}_n , which is an $M \times M$ block matrix of $n \times n$ circulants can be specified by giving the polynomials of the first columns of each of its circulants: $\mathbf{A}_n(y) = [a_{ij}(y)], 0 \leq i, j \leq (M - 1)$, where the indeterminate y is temporarily used to emphasize that the $a_{ij}(y)$ are in the ring of polynomial *mod* $(y^n - 1)$. The notation can be simplified by making the assumption that $d < M$, meaning that the size of a block is larger than the span of the polynomial. In this case, temporarily let $\mathbf{B} = [b_{ij}]$ be the $M \times M$ binary circulant whose first column is $a(x)$. Then $a_{ij}(y) = b_{ij}$ for $0 \leq j \leq i \leq (n - 1)$, $a_{ij}(y) = yb_{ij}$ for $j > i$. The QC parity check matrix \mathbf{H} consists of $L \times L$ circulants \mathbf{h}_{ij} with first column polynomials $h_{ij}(x)$, and the blocking permutation is performed on every $L \times L$ circulant, $\mathbf{A} = \mathbf{h}_{ij}$, $a(x) = h_{ij}(x)$ for $0 \leq i \leq (n_C - k), 0 \leq j \leq n_C$.

Fig. 3 illustrates this blocking operation for the code of Fig.1 in the case that $L = 21$, $M = 7$ and $n = 3$. Notice that \mathbf{H}_n looks like the parity check matrix \mathbf{H} with $L = 7$ of Fig. 2 with every non-zero replaced by a 3×3 circulants. Each non-zero entry below the diagonal is replaced by a 3×3 identity; each non-zero above the diagonal is replaced by a 3×3 identity matrix with columns cyclically shifted by one position.

5 Parallel quasi-cyclic turbo codes

The purpose of putting a convolutional code into this blocked quasi-cyclic form is to enable the introduction of interleavers to form a QC turbo code. In this section the code equations are developed for two codes interconnected by a periodic interleaver.

As mentioned above, there is the possibility that the $Lk = Mn_C$ information inputs to the convolutional codes do not form an information set for the tailbiting QC code. Nonetheless, a parallel concatenated code can be defined by identifying “input” $U(x)$ bits of one encoder with a permuted $U'(x)$ set of bits in the other, and the set of solutions to the equations thereby imposed can be found. Although it is a slight abuse of nomenclature, these leftmost Mn_C positions will still be referred to as the $U(x)$ positions.

Let the QC parity check matrix be $\mathbf{H} = [\mathbf{A} \ \mathbf{B}]$ with \mathbf{A} the matrix in the Mn_C $U(x)$ positions. If an arbitrary permutation Π is introduced between the two codes, the parity check matrix for the turbo code

can be expressed as

$$\mathbf{H}_T = \begin{bmatrix} \mathbf{A} & \mathbf{B} & \mathbf{0} \\ \mathbf{A}\mathbf{\Pi} & \mathbf{0} & \mathbf{B} \end{bmatrix}, \quad (4)$$

where $\mathbf{\Pi}$ is the $Mnk \times Mnk$ permutation matrix of $\mathbf{\Pi}$. Although \mathbf{A} and \mathbf{B} have circulant structure, the introduction of an arbitrary interleaver can destroy it entirely.

Restricting the form of the interleaver can preserve a quasi-cyclic invariance of the constituent tailbiting code in the turbo code composed. Suppose that a tailbiting QC code has parity check matrix $\mathbf{H}_n = [\mathbf{A} \ \mathbf{B}]$, \mathbf{A} and \mathbf{B} being $M(n_C - k) \times Mk$ and $M(n_C - k) \times M(n_C - k)$ block matrices, respectively. Let $\mathbf{A} = [\mathbf{a}_{ij}]$, $0 \leq i \leq (M(n_C - k) - 1)$, $0 \leq j \leq (Mk - 1)$, each \mathbf{a}_{ij} an $n \times n$ circulant. $U(x)$ bits can be indexed by 2-tuples $[j, l]$, $0 \leq j \leq (Mk - 1)$, $0 \leq l \leq (n - 1)$, according to the circulant block j and the position l within the circulant block in which it appears.

Definition (Quasi-Cyclic Permutation): A permutation $\mathbf{\Pi}$ acting on the $U(x)$ bits of a QC code will be called *quasi-cyclic* if $\mathbf{\Pi}([j, l]) = [\sigma(j), (l + s_j) \bmod n]$ for some arbitrary permutation σ acting on the integers Z_{Mk} and a right cyclic shift s_j for the j block.

A quasi-cyclic $\mathbf{\Pi}$ maps all the bits from one circulant block to a possibly different circulant block, and simultaneously right cyclically shifts the bits in the block. Consequently, if \mathbf{A} is a block matrix of circulants, $\mathbf{A}\mathbf{\Pi}$ is also, and the parity check matrix (4) defines a quasi-cyclic code.

The formal notation may obscure an otherwise simple concept. Viewed in the normal order of transmission for both codes, the permutation creates a connection between an input bit in one encoder and a time shifted information bit in the other encoder. To preserve the quasi-cyclic property, the connections must be invariant under a simultaneous shift of M positions in both $U(x)$ streams, with cyclic wrap-around *mod* L .

Fig. 4 illustrates one quasi-cyclic permutation and resulting quasi-cyclic turbo code for the encoders of Fig. 1, now with $L = 21$. The periodic permutation shown has a straightforward description. Let the integers Z_{21} index the inputs to one copy of the convolution code. The permutation $\mathbf{\Pi}$ is given by $\pi(i) = 4i \bmod 21$. Since four is relatively prime to 21, the permutation is well defined. Also $\pi(i + 7) = (4i + 28) \bmod 21 = \pi(i) + 7$, and the permutation is invariant under shifts of seven position. With $M = 7$ and $n = 3$, it is a QC permutation, as needed.

6 Permutations and Transformed Equations

The equations for a QC turbo code can be written as in Eq. (4) with $\mathbf{\Pi} = \mathbf{\Pi}_{QC}$ a QC permutation matrix. As above, let $\mathbf{A} = [\mathbf{a}_{ij}]$, with the $n \times n$ circulant having first column polynomial $a_{ij}(x)$. Let $\mathbf{A}' = \mathbf{A}\mathbf{\Pi}_{QC} = [\mathbf{a}'_{ij}]$. By the definition of the QC permutation, $a'_{ij}(x) = x^{s_j} a_{i\sigma(j)}(x)$. The permutation σ reorders the columns of the block matrix, and right shift of the bits in the block by s_j positions is tantamount to a left shift of the rows of the circulant, which multiplies the first column polynomial by x^{s_j} .

With this explicit expression for the first column polynomials of the circulants of \mathbf{A}' , the matrix of polynomials for \mathbf{H}_{QCT} , $H_{QCT}(x)$ specifying all its circulants is known. The equations can be transformed by multiplying every circulant by a Fourier transform matrix formed using a primitive n root of unity, α . A codeword \mathbf{X} must satisfy

$$\begin{aligned} [H_{QCT}(\alpha^i) \otimes \mathbf{f}_i] \mathbf{X}^T &= \\ [H_{QCT}(\alpha^i) \otimes [1 \ \alpha^i \ \alpha^{2i} \ \dots \ \alpha^{(n-1)i}]] \mathbf{X}^T &= 0 \end{aligned}$$

for $0 \leq i \leq (n - 1)$.

Algebraic analyses of the solutions space and bounds on minimum distance can exploit this general quasi-cyclic form, building on understandings derived from cyclic coding theory. Some of these are developed in [12]. At this time, we do not have any general methods guaranteed to give strong distance bounds in all cases, but the decomposition of the codespace into subspaces can simplify the search for the possible existence of very low weight words in the QC turbo code. If nothing more, the restricted QC form significantly simplifies a computational search for weaknesses.

For the example of Fig. 4, polynomial notation, the parity check matrix for one tailbiting QC convolutional code is $\mathbf{H}_{QC}(x) = [\mathbf{A} \ \mathbf{B}] =$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & x & x & 0 & 1 & 0 & 0 & 0 & x & 0 & x \\ 0 & 1 & 0 & 0 & 0 & x & x & 1 & 1 & 0 & 0 & 0 & x & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & x & 0 & 1 & 1 & 0 & 0 & 0 & x \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}. \quad (5)$$

In the QC turbo code with $\mathbf{\Pi}_{QC}$ as given

$$\mathbf{A}\mathbf{\Pi}_{QC} = \begin{bmatrix} 1 & x & 0 & x^2 & 0 & 0 & 0 \\ 0 & 0 & x & x^2 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & x^2 & 1 & 0 \\ 1 & 0 & x & 0 & 0 & 0 & 1 \\ 0 & 1 & x & 0 & x^2 & 0 & 0 \\ 0 & 0 & 0 & x & x^2 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & x^2 & 1 \end{bmatrix}. \quad (6)$$

The circulants are 3×3 for this example, and the code equations can be transformed using a third root

of unity, ω , on $GF(4)$. There are only three rows to the Fourier transform matrix, and $\mathbf{H}_{QCT}(x)$ must be evaluated at $x = 1, \omega$, and ω^2 . The dual of the $GF(4)$ code that is the row space of $\mathbf{H}_{QCT}(\omega)$ is a $[21,7,9]$ code over $GF(4)$. Codewords in this subspace have weight two or zero in any set of three bits of a circulant, and the minimum distance of nine implies the existence of QC turbo code words of weight 18. The dual of the $GF(2)$ code that is the row space of $\mathbf{H}_{QCT}(1)$ is a $[21,10,4]$ binary code. The minimum weight codewords have all three bits either one or all three zero in each circulant, and the minimum distance of four implies the existence of weight 12 words in the QC turbo code. In fact, there are weight four solutions confined to the parity bit positions, indicating the presence of periodic solutions with the $U(x)$ bits zero. Any word in the QC turbo code must be the linear sum of words from the two eigenspace, and this implies that the minimum distance of the QC turbo code must be at least nine. Testing the code exhaustively using MAGMA [7], the QC turbo code is a $[63,24,9]$.

7 Severing the Quasi-Cyclic Turbo Code

A QC turbo code can be decoded with any of the standard algorithms using a soft-decision decoder for each constituent convolutional code, with the permutation network iteratively passing extrinsic probabilities between the two decoders. This requires that the information from the beginning of the transmission be accessible to the decoder at the same time as information from the end. Although decoding information for the entire transmission is stored through multiple iterations, this is an unusual requirement for the typical turbo code system.

When this poses a difficulty, it can be circumvented with a minor loss of rate. All that is necessary is to restrict the space of QC codewords to include only those for which the encoder starts and ends in the zero state. Suppose each encoder of the two encoders is of degree m and has m memory states. To parity check equations of the tailbiting convolutional code, one can add m linear equations that force the first encoder to be in the zero state at time $t = 0$, which is the same as $t = L$ for the tailbiting code. Another m linear equations will similarly force the second encoder to be in the zero state at $t = 0$. Thus with a loss of at most $2m$ information bits, the zero state condition can be imposed at both the beginning and the end. For decoding this slightly smaller subcode, the standard turbo code algorithms can be used without modification, and the loss of rate is no greater than for normal termination of the two convolutional codes. Since the words are in a subcode of the QC turbo code, the minimum weight word in this QC subcode is the minimum distance of the

zero start and end state turbo code, so no damage is done to any algebraic arguments bounding the minimum distance of the QC code. Finding the basis of solutions respecting the zero state constraint requires some additional effort and may introduce a constraint on the presumed input sequence, a potential nuisance for the encoding algorithm.

8 Conclusion

Parallel concatenated turbo codes incorporating an interleaver defined by a random permutation have achieved remarkably good performance. But random permutations have numerous drawbacks that mean that in many practical applications, a truly random permutation is not employed. In this paper we extend the line of inquiry into algebraic structure for turbo codes that was begun in [13]. In that paper, for the special class of serial concatenated repeat-accumulate codes, we proposed periodic, quasi-cyclic designs that in a couple of simulation studies performed as well or better than carefully qualified “random” permutations. Here the algebraic framework has been adapted to parallel concatenated codes, and it is shown that QC periodic permutations give rise to QC turbo codes, whose equations can be studied on an extension field of $GF(2)$. This alone creates a bridge to the rich algebraic theory of cyclic and quasi-cyclic codes, and simplifies the search for very low weight words in a proposed permutation interleaver design. It also gives some indication of how the choice of interleaver and the choice of convolutional code interact in creating the parity check equations for the turbo code.

Important open questions remain. The aspiration of this work is to gain enough algebraic understanding to facilitate the design of turbo codes with effective interleavers for iterative decoding and to establish meaningful bounds on the minimum distance of the composite turbo code. We conjecture that the class of quasi-cyclic permutations that will create QC turbo codes is rich, rich enough to contain codes that will perform as well or better than random interleavers. A major goal of ongoing research is strengthen the algebraic minimum distance bounding techniques so as to give a general methodology for establishing a good bound, if not a tight bound, on the minimum distance of a quasi-cyclic code.

REFERENCES

- [1] S. Benedetto and G. Montorsi, “Unveiling Turbo Codes: Some results on parallel concatenated coding schemes,” *IEEE Trans. on Information Theory*, vol. 42, no. 2, pp. 409-428, March 1996.
- [2] C. Berrou, A. Glavieux, and P. Thitimajshima, “Near Shannon limit error-correcting coding

- and decoding: turbo codes,” *Proc. 1993 IEEE International Conference on Communications*, Geneva, Switzerland, pp. 1064-1070, May 1993.
- [3] D. Divsalar, H. Jin, and R. J. McEliece, “Coding theorems for ”turbo-like codes,” *Proceedings 36th Allerton Conf. Commun., Control, and Computing*, pp. 201-210, September 1998.
- [4] D. Divsalar, S. Dolinar, and F. Pollara, “Iterative turbo decoder analysis based on gaussian density evolution,” preprint, pp. 1-34, May 2000.
- [5] S. Dolinar and D. Divsalar, “Weight distributions for turbo codes using random and nonrandom permutations,” TDA Progress Report 42-121, JPL, August, 1995.
- [6] C. Heegard and S. B. Wicker, *Turbo Codes*, Kluwer Academic Publishers, 1999.
- [7] MAGMA software for mathematics from the University of Sydney Computational Algebra Group (John Cannon et al.), Sydney, Australia, 1999.
- [8] R. J. McEliece, “The algebraic theory of convolutional codes,” Chapt. 12 in *The Handbook of Coding Theory*, V. S. Pless and W. C. Huffman, Ed., North-Holland, 1998.
- [9] T. Richardson and R. Urbanke, “The capacity of low-density parity check codes under message-passing decoding,” *IEEE Trans. on Information Theory*, submitted.
- [10] R. M. Tanner, “A recursive approach to low complexity codes,” *IEEE Trans. on Information Theory*, vol. IT-27, pp. 533-547, Sept. 1981.
- [11] R. M. Tanner, “Convolutional codes from quasi-cyclic codes: A link between the theories of block and convolutional codes,” Tech. Rpt. UCSC-CRL-87-21, Univ. of Calif., Santa Cruz, Nov. 1987.
- [12] R. M. Tanner, “A transform theory for a class of group-invariant codes,” *IEEE Trans. on Information Theory*, vol. 34, no. 4, pp. 752-775, July 1988.
- [13] R. M. Tanner, “On quasi-cyclic repeat-accumulate codes,” *Proceedings 37th Allerton Conf. Commun., Control, and Computing*, pp. 249-259, September 1999.
- [14] N. Wiberg, *Codes and Decoding on General Graphs*, Linköping Studies in Science and Technology, No. 440, 1996.
- [15] N. Wiberg, H.-A. Loeliger, R. Koetter, “Codes and Iterative Decoding on General Graphs,” *European Trans. on Telecomm.*, vol. 6, no. 5, pp. 513-526, Sept. 1995.