entropy

MDPI

Article

Systematic Encoding and Shortening of PAC Codes

Erdal Arıkan

Elecrical-Electronics Engineering Department, Bilkent University, Ankara 06800, Turkey;
arikan@ee.bilkent.edu.tr; Tel.: +(90)-312-290-1347

Received: 28 October 2020; Accepted: 13 November 2020; Published: 15 November 2020

Abstract: Polarization adjusted convolutional (PAC) codes are a class of codes that combine channel polarization with convolutional coding. PAC codes are of interest for their high performance. This paper presents a systematic encoding and shortening method for PAC codes. Systematic encoding is important for lowering the bit-error rate (BER) of PAC codes. Shortening is important for adjusting the block length of PAC codes. It is shown that systematic encoding and shortening of PAC codes can be carried out in a unified framework.

Keywords: PAC codes; polar codes; systematic encoding; code shortening

1. Introduction

PAC codes are a class of linear block codes designed to improve the performance of polar codes by combining channel polarization with convolutional coding [1]. It has been shown that PAC codes can perform better than polar codes [1], in some instances performing close to the theoretical limits for finite-length codes.

Given the potential of PAC codes for applications requiring extreme reliability at short block-lengths, it is of interest to investigate various aspects of PAC codes that may be important in practice. In this paper, we study systematic encoding and shortening of PAC codes. Systematic encoding is of interest mainly because it provides a better bit error rate (BER) performance compared to non-systematic encoding. Code shortening is important as a means of providing flexibility is choosing the code length. The BER advantage of systematic coding is illustrated in Figure 1 for a PAC code of length $N = 128$ and rate $R = 1/2$ on an additive Gaussian noise channel with binary modulation. A better BER performance is important in concatenation schemes where an outer code corrects the bit errors left over by an inner PAC code.

In Section 2, we give a definition of PAC codes and their non-systematic encoding. In Section 3, we develop a method for systematic encoding of PAC codes. In Section 4, we indicate how the systematic encoding method of Section 3 can be used for shortening PAC codes.

Throughout, we restrict attention to PAC codes over the binary field $\mathbb{F}_2 = \{0, 1\}$. All algebraic operations are over vector spaces over $\mathbb{F}_2$. $\mathbb{F}_2^N$ will denote row vectors of length N over $\mathbb{F}_2$ and $\mathbb{F}_2^{N\times M}$ will denote matrices with N rows and M columns. For any $\mathbf{v} = (v_1, \ldots, v_N) \in \mathbb{F}_2^N$ and $\mathcal{A} \subset \{1, 2, \ldots, N\}$, let $\mathbf{v}_{\mathcal{A}}$ denote the subvector $(v_i : i \in \mathcal{A})$. For any $\mathbf{G} \in \mathbb{F}_2^{N\times M}$, $\mathcal{A} \subset \{1, 2, \ldots, N\}$, and $\mathcal{B} \subset \{1, 2, \ldots, M\}$, let $\mathbf{G}_{\mathcal{A},\mathcal{B}}$ denote the matrix obtained after deleting the rows of $\mathbf{G}$ not in $\mathcal{A}$ and columns of $\mathbf{G}$ not in $\mathcal{B}$. The notation $\mathbf{0}$ denotes a vector or matrix all of whose elements are 0 and $\mathbf{I}$ denotes an identity matrix.

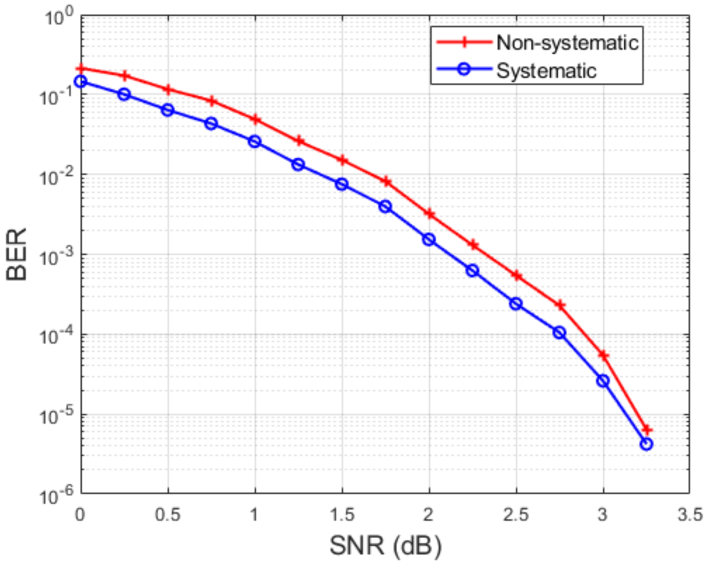

Figure 1. BER comparison for systematic and non-systematic PAC codes.

2. PAC Codes

A PAC code over $\mathbb{F}_2$ is a linear block code parametrized by $(N, K, \mathcal{A}, \mathbf{f}, \mathbf{g})$ where N is a code block length, K is a code dimension, $\mathcal{A}$ is a data index set, $\mathbf{f} \in \mathbb{F}_2^{N-K}$ is a frozen word, and $\mathbf{g} = (g_0, g_1, \ldots, g_m) \in \mathbb{F}_2^{m+1}$ is a convolution impulse response with $g_0 = 1$, $g_m = 1$, with g_i subject to design for $0 < i < m$. The data index set $\mathcal{A}$ is a subset of $\{1, 2, \ldots, N\}$ with size $|\mathcal{A}| = K$. The parameter $(1 + m)$ will be called the *span* of the impulse response $\mathbf{g}$. The span of any impulse response $\mathbf{g}$ that we consider here will be bounded by the block length N. Sometimes, when the span cannot or need not be shown explicitly, we will write $\mathbf{g} = (g_0, g_1, \ldots, g_{N-1})$ to denote an impulse response, with the understanding that $g_i = 0$ for i greater than or equal to the span of $\mathbf{g}$.

An encoder for a PAC code encodes data words $\mathbf{d} \in \mathbb{F}_2^K$ into codewords $\mathbf{x} \in \mathbb{F}_2^N$ by computing a convolution followed by a polar transform. In the convolution step, a convolution input word $\mathbf{v} \in \mathbb{F}_2^N$ is prepared by setting $\mathbf{v}_{\mathcal{A}} = \mathbf{d}$ and $\mathbf{v}_{\mathcal{A}^c} = \mathbf{f}$, and a convolution $\mathbf{u} = \mathbf{v} * \mathbf{g}$ is applied to $\mathbf{v}$ to obtain a polar transform input word $\mathbf{u} \in \mathbb{F}_2^N$. ($\mathcal{A}^c$ denotes the complement of $\mathcal{A}$ in $\{1, 2, \ldots, N\}$.) In the polar transform step, the codeword $\mathbf{x} \in \mathbb{F}_2^N$ is obtained by computing $\mathbf{x} = \mathbf{u}\mathbf{L}$, where $\mathbf{L} = \mathbf{F}^{\otimes n}$ is the polar transform matrix, defined as the nth Kronecker power of a kernel matrix $\mathbf{F} = \left[\begin{smallmatrix} 1 & 0 \\ 1 & 1 \end{smallmatrix}\right]$.

The convolution step $\mathbf{u} = \mathbf{v} * \mathbf{g}$ involves the computation

$$u_i = \sum_{j=0}^{m} v_{i-j} g_j, \quad \text{for } i = 1, 2, \cdots, N, \tag{1}$$

where v_{i-j} is interpreted as 0 if $i - j \leq 0$. In the following analysis, we will represent the convolution alternatively as a linear transformation $\mathbf{u} = \mathbf{v}\mathbf{T}$ where $\mathbf{T} \in \mathbb{F}_2^{N \times N}$ is an upper-triangular Toeplitz matrix of the form

$$\mathbf{T} = \begin{bmatrix} g_0 & g_1 & g_2 & \cdots & g_m & 0 & \cdots & 0 \\ 0 & g_0 & g_1 & g_2 & \cdots & g_m & \cdots & \vdots \\ 0 & 0 & g_0 & g_1 & \ddots & \cdots & g_m & \vdots \\ \vdots & 0 & \ddots & \ddots & \ddots & \ddots & \cdots & \vdots \\ \vdots & \cdots & \ddots & \ddots & \ddots & \ddots & 0 & \vdots \\ \vdots & \cdots & \cdots & \ddots & 0 & g_0 & g_1 & g_2 \\ \vdots & \cdots & \cdots & \cdots & 0 & 0 & g_0 & g_1 \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 & 0 & g_0 \end{bmatrix}. \tag{2}$$

The first row of $\mathbf{T}$ is determined by $\mathbf{g}$ and the rows that follow are shifted versions of the first row. Please note that if $m = 0$ then $\mathbf{T}$ becomes the identity matrix and PAC codes contain polar codes as a special case. To exclude this possibility, PAC codes are often defined with the condition that $m \geq 1$. However, for purposes of the present paper, there is no need to place such a restriction on m.

The encoding operation for PAC codes can be defined more compactly by defining a generator matrix $\mathbf{G} = \mathbf{TL}$. Then, the encoder implements the mapping $\mathbf{x} = \mathbf{vG}$ after preparing the vector $\mathbf{v}$ in the same way as above. A direct implementation of the transform $\mathbf{x} = \mathbf{vG}$, without exploiting the structure in $\mathbf{G}$, has complexity $\mathcal{O}(N^2)$, while the two-step encoder described above has complexity $\mathcal{O}(mN)$ for the convolution operation and $\mathcal{O}(N \log N)$ for the polar transform. Since PAC codes typically have $m \ll N$, the complexity of implementing $\mathbf{x} = \mathbf{vG}$ using the triangular factorization $\mathbf{G} = \mathbf{TL}$ results in significant cost savings. Below, as we develop a systematic PAC encoder, we will exploit this triangular factorization for reducing complexity.

3. Systematic Encoding

The above encoder for a PAC code is non-systematic in the sense that the data word $\mathbf{d}$ does not appear transparently as part of the codeword $\mathbf{x}$. The goal in this paper is to give a systematic encoding method so that there is a subset of coordinates $\mathcal{A}$ such that $\mathbf{x}_{\mathcal{A}} = \mathbf{d}$.

We will consider instances of the systematic encoding problem for PAC codes that are characterized by a collection of parameters $(\mathbf{T}, \mathbf{L}, \mathcal{A}, \mathcal{B}, \mathbf{f}, \mathbf{d})$ where $\mathbf{T} \in \mathbb{F}_2^{N \times N}$ is an invertible upper-triangular Toeplitz matrix, $\mathbf{L} \in \mathbb{F}_2^{N \times N}$ is the polar transform matrix (which is an invertible lower-triangular matrix), $\mathcal{A}$ and $\mathcal{B}$ are subsets of $\{1, 2, \ldots, N\}$ with sizes K and $N - K$, respectively, $\mathbf{f} \in \mathbb{F}_2^{N-K}$ is a fixed vector, and $\mathbf{d} \in \mathbb{F}_2^K$ is a data word. Given such an instance, a systematic encoder seeks a solution to the set of equations

$$\mathbf{x} = \mathbf{vTL}, \quad \mathbf{v}_{\mathcal{B}} = \mathbf{f}, \quad \mathbf{x}_{\mathcal{A}} = \mathbf{d}. \tag{3}$$

More specifically, a systematic PAC encoder seeks to determine the missing part $\mathbf{x}_{\mathcal{A}^c}$ of the codeword $\mathbf{x}$ subject to the conditions (3). To analyze this problem, rewrite $\mathbf{x} = \mathbf{vTL}$ in terms of $\mathbf{G} = \mathbf{TL}$ as

$$\mathbf{x}_{\mathcal{A}} = \mathbf{v}_{\mathcal{B}} \mathbf{G}_{\mathcal{B},\mathcal{A}} + \mathbf{v}_{\mathcal{B}^c} \mathbf{G}_{\mathcal{B}^c,\mathcal{A}}, \quad \mathbf{x}_{\mathcal{A}^c} = \mathbf{v}_{\mathcal{B}} \mathbf{G}_{\mathcal{B},\mathcal{A}^c} + \mathbf{v}_{\mathcal{B}^c} \mathbf{G}_{\mathcal{B}^c,\mathcal{A}^c} \tag{4}$$

where $\mathcal{A}^c$ and $\mathcal{B}^c$ denote the complements of $\mathcal{A}$ and $\mathcal{B}$ in $\{1, 2, \ldots, N\}$, respectively. Substituting $\mathbf{x}_{\mathcal{A}} = \mathbf{d}$ and $\mathbf{v}_{\mathcal{B}} = \mathbf{f}$ into (4), and solving for $\mathbf{x}_{\mathcal{A}^c}$, we obtain a formal solution as

$$\mathbf{x}_{\mathcal{A}^c} = \mathbf{d} \left(\mathbf{G}_{\mathcal{B}^c,\mathcal{A}} \right)^{-1} \mathbf{G}_{\mathcal{B}^c,\mathcal{A}^c} + \mathbf{f} \left[\mathbf{G}_{\mathcal{B},\mathcal{A}^c} - \mathbf{G}_{\mathcal{B},\mathcal{A}} \left(\mathbf{G}_{\mathcal{B}^c,\mathcal{A}} \right)^{-1} \mathbf{G}_{\mathcal{B}^c,\mathcal{A}^c} \right], \tag{5}$$

which is valid if and only if the matrix $\mathbf{G}_{\mathcal{B}^c,\mathcal{A}}$ is invertible. (Please note that $\mathbf{G}_{\mathcal{B}^c,\mathcal{A}}$ is a square matrix since the size of $\mathcal{B}^c$ equals the size of $\mathcal{A}$ by definition.) One way to ensure that $\mathbf{G}_{\mathcal{B}^c,\mathcal{A}}$ is invertible is to choose $\mathcal{A}$ and $\mathcal{B}$ as complementary sets so that $\mathbf{G}_{\mathcal{B}^c,\mathcal{A}}$ becomes a principal submatrix $\mathbf{G}_{\mathcal{A},\mathcal{A}}$ of $\mathbf{G}$.

(Since $\mathbf{G}$ is the product of two invertible matrices, it is invertible; hence, all its principal submatrices are invertible.) We summarize this result as follows.

Proposition 1. *The systematic encoding problem* (3) *for PAC codes has a solution whenever* $\mathcal{B}^c = \mathcal{A}$, *and the solution is given by*

$$\mathbf{x}_{\mathcal{A}^c} = \mathbf{d}\left(\mathbf{G}_{\mathcal{A},\mathcal{A}}\right)^{-1}\mathbf{G}_{\mathcal{A},\mathcal{A}^c} + \mathbf{f}\left[\mathbf{G}_{\mathcal{A}^c,\mathcal{A}^c} - \mathbf{G}_{\mathcal{A}^c,\mathcal{A}}\left(\mathbf{G}_{\mathcal{A},\mathcal{A}}\right)^{-1}\mathbf{G}_{\mathcal{A},\mathcal{A}^c}\right]. \tag{6}$$

Thus, in principle, we have already provided a solution to the systematic encoding problem for any PAC code. However, the complexity of solving the systematic encoding problem by computing $\mathbf{x}_{\mathcal{A}^c}$ using (6) involves $\mathcal{O}((N-K)^2)$ arithmetic operations (additions and multiplications in $\mathbb{F}_2$), which may be prohibitively complex for many applications.

In the rest of this section, we develop a low-complexity systematic encoder for PAC codes under the assumption that the data index set $\mathcal{A}$ is chosen so that $\mathbf{L}_{\mathcal{A}^c,\mathcal{A}} = \mathbf{0}$ is satisfied. This condition is not as restrictive as it may appear since it is satisfied by the preferred choices for the data index set $\mathcal{A}$, such as when $\mathcal{A}$ is chosen according to a polar coding design rule or a Reed-Muller design rule [1].

For clarity, we restate the systematic encoding problem considered in the rest of this section as follows. Given a data word $\mathbf{d} \in \mathbb{F}_2^K$ and a data index set $\mathcal{A}$ for which $\mathbf{L}_{\mathcal{A}^c,\mathcal{A}} = \mathbf{0}$, find a codeword $\mathbf{x} \in \mathbb{F}_2^N$ so that

$$\mathbf{x} = \mathbf{v}\mathbf{T}\mathbf{L}, \quad \mathbf{v}_{\mathcal{A}^c} = \mathbf{f}, \quad \mathbf{x}_{\mathcal{A}} = \mathbf{d}. \tag{7}$$

Proposition 2. *The systematic encoding problem* (7) *can be solved by a method consisting of the following three steps. (i) Generate an auxiliary word* $\mathbf{c} \in \mathbb{F}_2^K$ *by computing* $\mathbf{c} = \mathbf{d}\left(\mathbf{L}_{\mathcal{A},\mathcal{A}}\right)^{-1}$. *(ii) Compute a convolution input-output pair* $(\mathbf{v}, \mathbf{u})$ *so that*

$$\mathbf{u} = \mathbf{v}\mathbf{T}, \quad \mathbf{u}_{\mathcal{A}} = \mathbf{c}, \quad \mathbf{v}_{\mathcal{A}^c} = \mathbf{f}. \tag{8}$$

(iii) Obtain the systematic codeword by computing the polar transform $\mathbf{x} = \mathbf{u}\mathbf{L}$.

Proof. The second and third steps ensure that $\mathbf{x} = \mathbf{v}\mathbf{T}\mathbf{L}$, with $\mathbf{v}_{\mathcal{A}^c} = \mathbf{f}$. Therefore, $\mathbf{x}$ is a codeword in the PAC code. Moreover, we have

$$\mathbf{x}_{\mathcal{A}} = \mathbf{u}_{\mathcal{A}}\mathbf{L}_{\mathcal{A},\mathcal{A}} + \mathbf{u}_{\mathcal{A}^c}\mathbf{L}_{\mathcal{A}^c,\mathcal{A}} = \mathbf{c}\mathbf{L}_{\mathcal{A},\mathcal{A}} = \mathbf{d},$$

since $\mathbf{L}_{\mathcal{A}^c,\mathcal{A}} = \mathbf{0}$, $\mathbf{u}_{\mathcal{A}} = \mathbf{c}$, and $\mathbf{c} = \mathbf{d}\left(\mathbf{L}_{\mathcal{A},\mathcal{A}}\right)^{-1}$. Thus, $\mathbf{x}_{\mathcal{A}} = \mathbf{d}$ is also satisfied, confirming that the encoding method is systematic. □

The above systematic encoding method calculates $\mathbf{v}_{\mathcal{A}}$ although systematic encoding does not explicitly call for the calculation of $\mathbf{v}_{\mathcal{A}}$. On the other hand, the calculation of $\mathbf{v}_{\mathcal{A}}$ proves (implicitly) that a solution to the systematic encoding problem exists.

Next, we examine the complexity of each step of the systematic encoding method of Proposition 2.

Proposition 3. *The first and third steps of the method in Proposition 2 each have complexity* $\mathcal{O}(N \log N)$.

Proof. The third step $\mathbf{x} = \mathbf{u}\mathbf{L} = \mathbf{u}\mathbf{F}^{\otimes n}$ is a polar transform operation, which is known to have complexity $\mathcal{O}(N \log N)$ [2] thanks to the recursive structure of the polar transform. As for the first step, a direct computation of $\mathbf{c} = \mathbf{d}(\mathbf{L}_{\mathcal{A},\mathcal{A}})^{-1}$ (without exploiting the special structure of the polar transform) has complexity $\mathcal{O}(K^2)$. A better method is to embed the calculation $\mathbf{c} = \mathbf{d}(\mathbf{L}_{\mathcal{A},\mathcal{A}})^{-1}$ in a polar transform operation, as in systematic encoding of polar codes [3–5]. To that end, we recall that the inverse of the polar transform $\mathbf{L} = \mathbf{F}^{\otimes n}$ is itself, *i.e.*, $\mathbf{L}^{-1} = \mathbf{L}$. This, combined with the condition that $\mathbf{L}_{\mathcal{A}^c,\mathcal{A}} = \mathbf{0}$, implies that $\left(\mathbf{L}_{\mathcal{A},\mathcal{A}}\right)^{-1} = \mathbf{L}_{\mathcal{A},\mathcal{A}}$. To see this last point, note that for any two matrices $\mathbf{A} \in \mathbb{F}_2^{N\times N}$ and $\mathbf{B} \in \mathbb{F}_2^{N\times N}$,

$$(\mathbf{A}\mathbf{B})_{\mathcal{A},\mathcal{A}} = \mathbf{A}_{\mathcal{A},\mathcal{A}}\mathbf{B}_{\mathcal{A},\mathcal{A}} + \mathbf{A}_{\mathcal{A},\mathcal{A}^c}\mathbf{B}_{\mathcal{A}^c,\mathcal{A}},$$

and let $\mathbf{A} = \mathbf{L}$ and $\mathbf{B} = \mathbf{L}^{-1} = \mathbf{L}$. Therefore, we have $\mathbf{c} = \mathbf{d}(\mathbf{L}_{\mathcal{A},\mathcal{A}})^{-1} = \mathbf{d}\mathbf{L}_{\mathcal{A},\mathcal{A}}$. Now, prepare a vector $\mathbf{x}' \in \mathbb{F}_2^N$ by setting $\mathbf{x}'_{\mathcal{A}} = \mathbf{d}$ and $\mathbf{x}'_{\mathcal{A}^c} = \mathbf{0}$, apply a polar transform $\mathbf{u}' = \mathbf{x}'\mathbf{L}$, and extract $\mathbf{c}$ from $\mathbf{u}'$ by setting $\mathbf{c} = \mathbf{u}'_{\mathcal{A}}$. This yields the desired result since

$$\mathbf{u}'_{\mathcal{A}} = \mathbf{x}'_{\mathcal{A}}\mathbf{L}_{\mathcal{A},\mathcal{A}} + \mathbf{x}'_{\mathcal{A}^c}\mathbf{L}_{\mathcal{A}^c,\mathcal{A}} = \mathbf{d}\mathbf{L}_{\mathcal{A},\mathcal{A}}.$$

□

Proposition 4. *The system of equations* (8) *in the second step of Proposition* 2 *can be solved by a sequential method of complexity* $\mathcal{O}(mN)$ *for a PAC code with a convolution impulse response* $\mathbf{g} = (g_0, g_1, \ldots, g_m)$ *(where* $g_0 \neq 0$ *by definition of PAC codes).*

Proof. To develop a sequential method that solves (8), we begin by rewriting the convolution Equation (1) as follows

$$u_i = g_0 v_i + g_1 v_{i-1} + \cdots + g_m v_{i-m} = v_i + s_i, \quad i = 1, 2, \ldots, N \tag{9}$$

where we used $g_0 = 1$ and have defined $s_i = g_1 v_{i-1} + \cdots + g_m v_{i-m}$ as an ith *feed-forward variable*. Please also note that in (9), we have used the convention that $v_j = 0$ for $j < 1$.

Observe that, for each $1 \leq i \leq N$, either $i \in \mathcal{A}$ or $i \in \mathcal{A}^c$. In the former case, we obtain u_i from the constraint $\mathbf{u}_{\mathcal{A}} = \mathbf{c}$; in the latter case, we obtain v_i from $\mathbf{v}_{\mathcal{A}^c} = \mathbf{f}$. Given the value of one of the elements of the pair (v_i, u_i), the other can be found from the relation $u_i = v_i + s_i$. Also, observe that s_i depends only on the knowledge of $(v_1, v_2, \ldots, v_{i-1})$. These observations suggest a sequential method for carrying out the second step of Proposition 2. The sequential method begins with $i = 1$ with $s_1 = 0$. Either $1 \in \mathcal{A}$ and $(v_1, u_1) = (c_1, c_1)$ where c_1 is the first element of the auxiliary word $\mathbf{c}$; or $1 \in \mathcal{A}^c$ and $(v_1, u_1) = (f_1, f_1)$ where f_1 is the first element of the frozen word $\mathbf{f}$. In either case, we can compute s_2 before proceeding to the next step of the sequential method. In general, the ith step of the sequential method begins with s_i available from the $(i-1)$th step and one determines the missing element of the pair (v_i, u_i) using the relation $u_i = v_i + s_i$. Thus, this method solves the system of equations (8). The method also provides a proof of existence and uniqueness of the solution.

The complexity of the sequential method given above is dominated by the complexity of calculating the feed-forward variables $(s_1, s_2, \ldots, s_N)$. From the definition of s_i, it is clear that s_i can be calculated using at most m multiplications and $m - 1$ additions in $\mathbb{F}_2$. Thus, the overall complexity is $\mathcal{O}(mN)$. □

Remark 1. *An inspection of the above proof will show that the sequential method of Proposition* 4 *can be used to solve the system of equations* (8) *for any IUT matrix* $\mathbf{T}$*; the Toeplitz property is not essential.*

The complexity $\mathcal{O}(mN)$ of the sequential method of Proposition 4 corresponds to a significant savings if $m \ll N$. If $m \ll N$ is not true, it may be worth working with the inverse of $\mathbf{T}$. To discuss this, we first cite a well-known result, see e.g., [6].

Proposition 5. *The class of all N-by-N IUT Toeplitz matrices form a group under matrix multiplication. Let* $\mathbf{T} \in \mathbb{F}_2^{N \times N}$ *be an IUT Toeplitz matrix with its first row given by* $\mathbf{g} = (g_0, g_1, \ldots, g_{N-1}) \in \mathbb{F}_2^N$*. (If* $\mathbf{g}$ *has span* $m+1$*, then* $g_i = 0$ *for* $m < i \leq N-1$*.) Then,* $\mathbf{T}^{-1} \in \mathbb{F}_2^{N \times N}$ *is an IUT Toeplitz matrix with first row given by* $\mathbf{h} = (h_0, h_1, \ldots, h_{N-1}) \in \mathbb{F}_2^N$ *where* $h_0 = (1/g_0)$ *and* $h_k = -\frac{1}{g_0}\sum_{i=1}^{k} g_{k-i} h_i$ *for* $k = 1, 2, \ldots, N-1$.

Proposition 5 allows us to recast the convolution problem (8) in an inverted form: Compute a convolution input-output pair $(\mathbf{v}, \mathbf{u})$ so that

$$\mathbf{v} = \mathbf{u}\mathbf{T}^{-1}, \quad \mathbf{v}_{\mathcal{A}^c} = \mathbf{f}, \quad \mathbf{u}_{\mathcal{A}} = \mathbf{c}. \tag{10}$$

The inverted problem (10) has the same form as the original problem (8) with the roles of $\mathbf{v}$ and $\mathbf{u}$ reversed. Therefore, it can be solved using the same sequential method described above. There may be an advantage in solving the inverted problem if the span of the first row of $\mathbf{T}^{-1}$ is shorter than that of $\mathbf{T}$. For example, let $\mathbf{T} \in \mathbb{F}_2^{16\times 16}$ be an IUT Toeplitz matrix with first row $\mathbf{g} = (1,1,1,1,0,0,0,0,1,1,0,1,1,0,1,0)$, with a span of 15. The inverse $\mathbf{T}^{-1} \in \mathbb{F}_2^{16\times 16}$ is the IUT Toeplitz matrix with first row $\mathbf{h} = (1,1,0,0,1,1,0,0,0,0,1,0,0,0,0,0)$, which has a span of 11.

We end this section by noting that for hardware implementations of the convolution operation in PAC encoding (both for systematic and non-systematic cases), one can use shift-register circuits that are commonly used in encoding algebraic codes. In particular, the convolution operation $\mathbf{u} = \mathbf{v} * \mathbf{g}$ (or, equivalently the transform $\mathbf{u} = \mathbf{v}\mathbf{T}$) can be implemented as shown in Figure 2. A version of the same circuit, with the left-most stage eliminated, generates the feed-forward variable s_i at point A' when v_{i-1} is provided as input at point A.

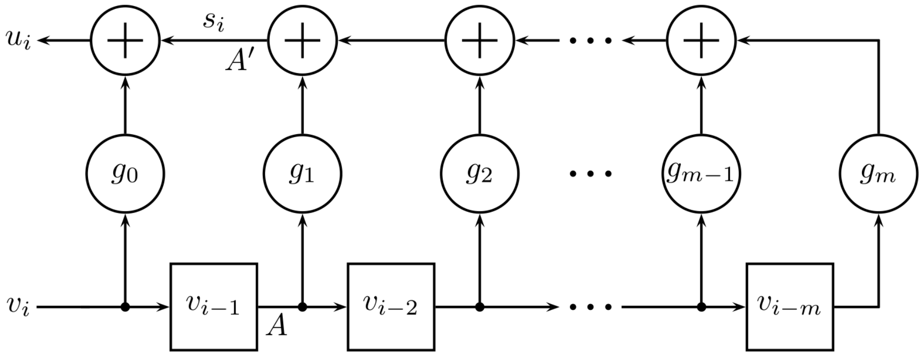

Figure 2. Convolution circuit.

4. Shortening of PAC Codes

PAC codes have native lengths that are powers of two, $N = 2^n$ for some $n \geq 1$. In many applications, it is necessary to adjust the code length to some desired value other than 2^n. One method for adjusting code length is code shortening in which a portion $\mathbf{x}_{\mathcal{C}}$ of the codeword $\mathbf{x}$ is constrained to a predetermined value, say zero, and is not transmitted, effectively reducing the code length from N to $N - |\mathcal{C}|$. A common method of code shortening for polar codes is to choose the set $\mathcal{C}$ so that $\mathbf{L}_{\mathcal{C}^c,\mathcal{C}} = \mathbf{0}$ [7,8]. The systematic encoding method for PAC codes presented above can be used to implement such a shortening method.

Suppose we desire shortening of a PAC code in connection with non-systematic encoding. We partition the index set $\{1,2,\ldots,N\}$ into three disjoint sets: a data index set $\mathcal{A}$, a frozen index set $\mathcal{B}$, and a shortening index set $\mathcal{C}$ subject to the condition $\mathbf{L}_{\mathcal{A}\cup\mathcal{B},\mathcal{C}} = \mathbf{0}$. Then, we apply the systematic encoding method presented above to the problem

$$\mathbf{x} = \mathbf{v}\mathbf{T}\mathbf{L}, \quad (\mathbf{v}_{\mathcal{A}}, \mathbf{v}_{\mathcal{B}}) = (\mathbf{d}, \mathbf{f}), \quad \mathbf{x}_{\mathcal{C}} = \mathbf{0}. \tag{11}$$

In other words, the data word $\mathbf{d}$ is treated as if it is part of the frozen part of the convolution input word $\mathbf{v}$, and the part $\mathbf{x}_{\mathcal{C}}$ of the codeword is treated as if it is the data part of the codeword in a systematic PAC code.

If on the other hand, we desire to shorten a systematic PAC code, then the index set $\{1,2,\ldots,N\}$ is partitioned into a data index set $\mathcal{A}$, a frozen index set $\mathcal{B}$, and a shortening index set $\mathcal{C}$ subject to the condition $\mathbf{L}_{\mathcal{B},\mathcal{A}\cup\mathcal{C}} = \mathbf{0}$, and we apply the above systematic encoding method to the problem

$$\mathbf{x} = \mathbf{v}\mathbf{T}\mathbf{L}, \quad \mathbf{v}_{\mathcal{B}} = \mathbf{f}, \quad (\mathbf{x}_{\mathcal{A}}, \mathbf{x}_{\mathcal{C}}) = (\mathbf{d}, \mathbf{0}). \tag{12}$$

In other words, we treat $(\mathbf{x}_{\mathcal{C}}, \mathbf{x}_{\mathcal{A}})$ as if all of it is data in a systematic PAC code.

Funding: This research received no external funding.

Acknowledgments: The author acknowledges helpful comments by reviewers.

Conflicts of Interest: The author declares no conflicts of interest.

References

1. Arıkan, E. From sequential decoding to channel polarization and back again. *arXiv* **2019**, arXiv: 1908.09594.
2. Arıkan, E. Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels. *IEEE Trans. Inf. Theory* **2009**, *55*, 3051–3073. [CrossRef]
3. Arıkan, E. Systematic polar coding. *IEEE Commun. Lett.* **2011**, *8*, 860–862. [CrossRef]
4. Sarkis, G.; Tal, I.; Giard, P.; Vardy, A.; Thibeault, C.; Gross, W.J. Flexible and low-complexity encoding and decoding of systematic polar codes. *IEEE Trans. Commun.* **2016**, *64*, 2732–2745. [CrossRef]
5. Vangala, H.; Hong, Y.; Viterbo, E. Efficient algorithms for systematic polar encoding. *IEEE Commun. Lett.* **2016**, *20*, 17–20. [CrossRef]
6. Commenges, D.; Monsion, M. Fast inversion of triangular Toeplitz matrices. *IEEE Trans. Autom. Control* **1984**, *3*, 250–251. [CrossRef]
7. Wang, R.; Liu, R. A novel puncturing scheme for polar codes. *IEEE Commun. Lett.* **2014**, *18*, 2081–2084. [CrossRef]
8. Bioglio, V.; Gabry, F.; Land, I. Low-complexity puncturing and shortening of polar codes. In Proceedings of the 2017 IEEE Wireless Communications and Networking Conference Workshops (WCNCW), San Francisco, CA, USA, 19–22 March 2017; pp. 1–6.

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.