IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 25, 2019, accepted April 16, 2019, date of current version May 6, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2913339

Construction of Multi-State Capacity-Approaching Variable-Length Constrained Sequence Codes With State-Independent Decoding

CONGZHE CAO AND IVAN FAIR, (Member, IEEE)
Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6G 1H9, Canada
Corresponding author: Congzhe Cao (congzhe@ualberta.ca)

This work was supported in part by the Natural Science and Engineering Research Council (NSERC) of Canada, and in part by the Alberta Innovates Technology Futures (AITF).

ABSTRACT We consider the construction of capacity-approaching variable-length constrained sequence codes based on the multi-state encoders that permit state-independent decoding. Based on the finite-state machine description of the constraint, we first select the principal states and establish the minimal sets. By performing partial extensions and normalized geometric Huffman coding, efficient codebooks that enable state-independent decoding are obtained. We then extend this multi-state approach to a construction technique based on the n-step FSMs. We demonstrate the usefulness of this approach by constructing the capacity-approaching variable-length constrained sequence codes with improved efficiency and/or reduced implementation complexity to satisfy a variety of constraints, including the runlength-limited (RLL) constraint, the DC-free constraint, and the DC-free RLL constraint, with an emphasis on their application in visible light communications.

INDEX TERMS Constrained sequence codes, variable-length codes, capacity-approaching codes, multi-state codes, state-independent decoding, visible light communication.

I. INTRODUCTION

Constrained sequence (CS) codes, such as runlength-limited (RLL) codes, DC-free codes and DC-free RLL codes, continue to be studied for application in digital transmission [1]–[4], magnetic and optical recording [5]–[7], non-volatile storage [8]–[11], DNA-based storage systems [12] and visible light communication (VLC) [13]–[18]. Most constrained sequence codes are fixed-length codes, where codebooks consist of source words and codewords of uniform length. However, it has been shown that simple, variable-length codes can achieve a higher code rate than fixed-length codes with lower implementation complexity [19]–[29]. Single-state codes allow codewords to be freely concatenated, whereas in multi-state codes, the encoded sequence is a function of both the source input and the state of the encoder. Well designed multi-state codes have the property that the received codewords can be instantaneously decoded without state information, which is important in limiting error propagation.

The associate editor coordinating the review of this manuscript and approving it for publication was Xueqin Jiang.

In [19]–[21] the authors present construction techniques for synchronous variable-length codes in which the ratio between source word length and codeword length is fixed, and therefore the code rate is the ratio of relatively small integers. Since the capacity of most constraints is irrational, synchronous variable-length codes approach the capacity only with large codebooks. In [24]–[27] the authors consider the design of capacity-approaching variable-length codes in which the constrained sequence encoder contains a single state, and therefore is comprised of codewords that can be freely concatenated. The application of such single-state codes in various constraints is discussed in [28], [29]. In this paper, we consider the use of multiple encoding states during encoding and propose a framework for designing variable-length constrained sequence codes to achieve near-capacity performance based on a multi-state encoder. These codes demonstrate high efficiency and retain the property of state-independent decoding.

The construction technique described in this paper can be applied to a large variety of constraints including the RLL constraint, the DC-free constraint, and the DC-free RLL constraint. We show that it is possible to construct

constrained sequence codes with this technique to achieve higher efficiency and lower implementation complexity than many codes in use in current communication and data storage systems. For some constraints, this multi-state construction technique can also result in codes with higher efficiency and shorter codeword lengths than the single-state technique outlined in [24]–[27]. Similar to those variable-length single-state codes, the codes proposed in this paper can be instantaneously decoded since no codeword is a prefix of another codeword. As noted above, our new codes require no state information during decoding.

The rest of this paper is organized as follows. In Section II we provide a brief background of constrained sequence coding theory, and review the construction technique of single-state variable-length constrained sequence codes in [24]–[27]. We propose our multi-state encoding approach in Section III and extend this approach to a construction technique based on n-step FSMs in Section IV. In Section V we consider a characteristic of n-step FSMs that applies to DC-free codes, and exploit this characteristic to construct simple and high-efficiency multi-state DC-free codes for visible light communications (VLC). In Section VI we provide conclusions. Examples are included throughout the paper.

II. PRELIMINARIES

A. CONSTRAINED SEQUENCE CODING

RLL and DC-free codes are two widely used classes of CS codes. RLL coded sequences are sequences where the number of bits between transitions is bounded. One approach for their construction is the generation of a (d, k) sequence, where d and k denote the minimum and maximum number of logic zeros between consecutive logic ones, followed by differential encoding that encodes a one as a change in value and a zero as no change. This results in minimum and maximum runlengths of $d + 1$ and $k + 1$ respectively. It is also possible to construct RLL codes directly without first generating a (d, k) code. DC-free codes are designed so that the spectral components at low frequency are suppressed to match the characteristics of the physical channel. In the time domain, the running digital sum (RDS) of a DC-free encoded sequence is bounded, where RDS is the ongoing summation of encoded bit weights in the sequence, given that a logic one has weight $+1$ and a logic zero has weight -1 [1]. Following the notation in [1] we use N to denote the maximum number of different RDS values in the DC-free sequence. This implies that at most $N - 1$ consecutive logic ones or logic zeros can exist in the coded sequence. In some systems, DC-free RLL constraints place limits on runlength other than those implied by the RDS bounds [30], [31].

It is well known that a constraint can be described with an FSM that contains states, edges and labels, where labels are the coded sequences resulting from transitions between states. For an FSM with $\mathcal{S}$ states, the matrix of the directed graph underlying the constraint is denoted by an $\mathcal{S} \times \mathcal{S}$ adjacency matrix $D = \{d_{ij}\}$, where d_{ij} is the number of edges transitioning from state i to state j. The transition probability matrix is denoted by an $\mathcal{S} \times \mathcal{S}$ matrix $Q = \{q_{ij}\}$, where q_{ij} is the probability of transitioning from state i to state j. Based on D, the maxentropic transition probabilities and steady-state distribution can be obtained which describes the statistical properties when the maximum amount of information is represented by the FSM [32].

The maximum amount of information that can be carried in a sequence that satisfies the constraint is the ***capacity*** of the constraint C, which is defined as [33]

$$C = \lim_{m \to \infty} \frac{\log_2 \mathcal{U}(m)}{m} \tag{1}$$

where $\mathcal{U}(m)$ is the number of constraint-satisfying sequences of length m. Given the FSM description of the constraint, the capacity can be evaluated as

$$C = \log_2 \lambda_{max}, \tag{2}$$

where λ_{max} is the largest real root of the determinant equation

$$det[D - zI] = 0 \tag{3}$$

where I is an identity matrix. As discussed in [1], maxentropic transition probabilities in an FSM are given by

$$q_{ij} = \lambda_{\max}^{-1} d_{ij} \frac{p_j}{p_i} \tag{4}$$

where $1 \leq i, j \leq \mathcal{S}$ and p is the eigenvector of D associated with the eigenvalue λ_{max}.

B. SINGLE-STATE VARIABLE-LENGTH CODES

In this section, we briefly review the single-state capacity-approaching encoding technique for variable-length constrained sequence codes introduced in [24]–[29]. As discussed in [26], [27], a critical step in construction of these codes is the formation of a minimal set from which codewords can be concatenated to generate constraint-satisfying sequences. A *minimal set* M can be established by enumerating all words that exit and re-enter a specific state in the FSM. A minimal set of an FSM is not unique and can have an infinite number of words [27], [34]. Criteria for choosing a minimal set are discussed in [26], [27]. A *partial extension* of a minimal set M_p is formed by post-fixing all words in the minimal set to some or all words of a previous partial extension, starting from the minimal set. After performing partial extensions over a minimal set, we obtain a set of codewords that are instantaneously decodable because no codeword is the prefix of another.

Normalized geometric Huffman (NGH) coding [34, Section 4.1] [35] is used to assign these codewords to the corresponding source words such that the maximum information density is approached. Starting with the desired maxentropic codeword probabilities obtained from (2)–(4) as the input probabilities, geometric Huffman coding merges the two smallest probabilities q_i and q_j according to the following rule to obtain the merged probability:

$$q_{merged} = \begin{cases} 2\sqrt{q_i q_j} & \text{if } q_i < 4q_j \\ q_i & \text{if } q_i \geq 4q_j. \end{cases} \tag{5}$$

TABLE 1. Codebook of a ($d = 1, k = 3$) code with efficiency of 98.9%.

Source word	Codeword
0	01
10	001
11	0001

The smaller probability is pruned from the Huffman tree when the lower condition is satisfied. As in the well-known Huffman construction technique, this process is repeated until a single value remains, and source words are assigned based on the merging pattern.

Given a one-to-one correspondence between variable-length source words and variable-length codewords, the average code rate $\overline{R}$ is

$$\overline{R} = \frac{\sum_{s_i} 2^{-s_i} s_i}{\sum_{o_i} 2^{-s_i} o_i} \tag{6}$$

where s_i is the length of i-th source word that is mapped to the i-th codeword of length o_i. The efficiency of a variable-length code is defined as $\eta = \overline{R}/C$. After obtaining $\bar{R}$, NGH coding repeats the above process with updated input probabilities and with C replaced by $\bar{R}$ when calculating the maxentropic probabilities in (2)–(4), until $\bar{R}$ converges.

Since different partial extensions result in different codebooks with different η, we establish parameters such as the maximum number of source words in the codebook n_{max}, or maximum codeword length l_{max}, and exhaustively search over all codebooks that satisfy these limits to find the one with the best η.

Example 1: ($(d = 1, k = 3)$ RLL code): the FSM of the $(d = 1, k = 3)$ constraint is shown in Fig. 1. According to [26], [27], we choose state 1 as the specified state upon which to construct the code; its minimal set is established as $M = \{01, 001, 0001\}$. We may choose to directly perform NGH coding over the minimal set to construct the simple codebook shown in Table 1 which has efficiency $\eta = 98.9\%$. By performing extensions with $n_{max} = 11$, we have constructed the code shown in Table 2 with $\eta = 99.25\%$. The partial extension process shown in Fig. 2 results in the codewords in Table 2. Starting from the minimal set, the set of words is updated as $M_p = \{01, 001, 0001\} \rightarrow \{01, 00101, 001001, 0010001, 0001\} \rightarrow \{01, 00101, 001001, 0010001, 000101, 0001001, 00010001\} \rightarrow \{01, 00101, 00100101, 001001001, 001001 0001, 0010001, 000101, 0001001, 00010001\} \rightarrow \{01, 00101, 00100101, 001001001, 0010010001, 0010001, 00010101, 00 0101001, 0001010001, 0001001, 00010001\}$. Note that the encoding and decoding processes are instantaneous with the words in this table. To compare, note that a widely used (1,3) RLL code is the Modified Frequency Modulation (MFM) code with $\eta = 91\%$ [1]. For more examples refer to [24]–[29] and [34, Chapter 6.4].

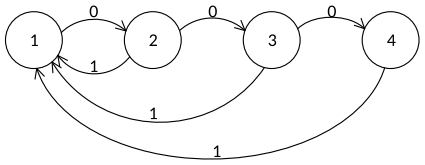


FIGURE 1. FSM of a (1, 3) RLL code.

TABLE 2. Codebook of a ($d = 1, k = 3$) code with efficiency of 99.25%.

Source word	Codeword	Source word	Codeword
0	01	11100	00010101
100	00101	11101	001001001
1010	0010001	11110	000101001
1011	0001001	111110	0010010001
1100	00010001	111111	0001010001
1101	00100101		

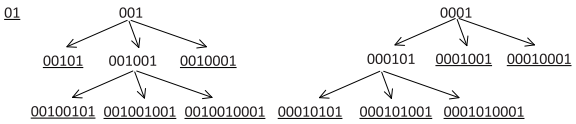


FIGURE 2. Partial extensions over the minimal set {01, 001, 0001}. Underlined codewords are in the final set after partial extensions.

Although codebooks with high η can be constructed with the single-state variable-length construction technique, two potential drawbacks of this approach should be considered. First, the codewords can be long, especially when long words exist in the minimal set, which increases the complexity of the encoding and decoding circuits. This occurs, for instance, with a large value of k in RLL constraints and a large value of N in DC-free constraints. Second, for some types of constraints, a minimal set consisting of a finite number of words does not capture all the constraint-satisfying sequences because of loops that exist in the FSMs. This results in a loss in the achievable code rate, as discussed in [26], [27]. Typical examples are DC-free constraints with $N \geq 4$ and most DC-free RLL constraints.

To overcome these drawbacks, in the next section we extend the single-state encoding technique by proposing a construction technique for variable-length constrained sequence codes that involves multiple states in the codebook.

III. MULTI-STATE ENCODING BASED ON FSM

A. SELECTION OF PRINCIPAL STATES

As outlined in [27], the first step in the design of a single-state variable-length CS code is selection of the specified state used to generate the minimal set. Similar to the single-state technique, the first step of our proposed multi-state technique is to determine which multiple states of the FSM that describes the constraint should be considered when generating the words in the minimal set. We call these states the principal states. Denote the j-th principal state as σ_j, $\sigma_j \in \Psi = \{\sigma_1, \sigma_2, \ldots, \sigma_{|\Psi|}\}$, where Ψ is the set of principal states and $|\Psi|$ is the size of Ψ. We first define how concatenation of words in the minimal set is performed.

Definition 1: (concatenation of words) Denote $W(\sigma_j) = \{w(\sigma_j)_1, w(\sigma_j)_2, \ldots, w(\sigma_j)_{|W(\sigma_j)|}\}$ as the set of words generated by the j-th principal state. Given $W(\sigma_j)$, let the set of next states corresponding to words in $W(\sigma_j)$ be $H(\sigma_j) = \{h(\sigma_j)_1, h(\sigma_j)_2, \ldots, h(\sigma_j)_{|W(\sigma_j)|}\}$ where $h(\sigma_j)_i \in \Psi, 1 \leq i \leq \left|H(\sigma_j)\right|$. The words in the minimal set are $W(\Psi) = \{W(\sigma_1), W(\sigma_2), \ldots, W(\sigma_{|\Psi|})\}$. When considering concatenation of words in $W(\Psi)$, $w(\sigma_j)_i$ is only allowed to be concatenated with words in the set $W(h(\sigma_j)_i)$.

Based on this definition of concatenation, we introduce the definition of principal states and the criterion to select them.

Definition 2: (principal states) Ψ is determined such that *all* constraint-satisfying sequences can be generated through the concatenation of words in $W(\Psi)$. In addition, to ensure that it is possible to instantaneously decode the received sequence, no word in $W(\sigma_i)$ is the prefix of a word in $W(\sigma_j)$ $\forall \sigma_i, \sigma_j \in \Psi$.

From Definition 2 it follows immediately that if a state σ_j has a loop associated with itself in the FSM, then $\sigma_j \in \Psi$ otherwise not all constraint-satisfying sequences can be generated with finite-length codewords due to the loop at σ_j. The principal states should also be selected such that $|W(\sigma_i)| = |W(\sigma_j)|$ $\forall \sigma_i, \sigma_j \in \Psi$, in order that each state will have a codeword associated with each source word in the codebook. Examples of the appropriate selection of principal states follow discussion of establishing the minimal set.

B. MINIMAL SET

Similar to the construction of single-state variable-length codes described in Section II-B, establishment of the minimal set is an essential step in our multi-state variable-length construction technique. After determining the principal states, we establish the minimal set of the constraint based on its underlying FSM. Given $|\Psi|$ principal states, the ***minimal set*** in multi-state encoding is a tabular representation that contains $2|\Psi|$ columns, where $|\Psi|$ of the columns indicate the words generated by the principal states, i.e. $W(\Psi) = \{W(\sigma_1), W(\sigma_2), \ldots, W(\sigma_{|\Psi|})\}$, and the other $|\Psi|$ columns indicate the next states corresponding to each word, i.e. $H(\Psi) = \{H(\sigma_1), H(\sigma_2), \ldots, H(\sigma_{|\Psi|})\}$.

Assignment of words in a minimal set with multiple states will, in general, result in the necessity for state-dependent decoding, which requires knowledge of both the received codeword and the corresponding encoding state in order to correctly determine the corresponding source word. Decoding that can be performed with knowledge of only the received codeword and without tracking the encoder state is called state-independent decoding. To enable state-independent decoding, the following necessary and sufficient condition [19] [31] must be satisfied.

Condition 1 (state-independent decoding): When assigning words from $W(\Psi)$ in the minimal set, the necessary and sufficient condition for state-independent decoding is that for $1 \leq u \leq v \leq y \leq |\Psi|$ and for each

$$W_r \in W(\sigma_u) \cap W(\sigma_y)$$

such that

$$W_r \notin W(\sigma_v)$$

there exists a $W_q \in W(\sigma_v)$ such that there exists no σ_l, $1 \leq l \leq |\Psi|$, for which $W_r, W_q \in W(\sigma_l)$.

As will become evident in the examples below, this condition implies that in the minimal set table, a word does not appear in more than one row. Therefore, enabling state-independent decoding requires satisfying Condition 1, which implies careful design of the encoder. It should be mentioned, however, that it may not be possible for $|W(\sigma_i)|$ and $\left|W(\sigma_j)\right|$ to be equal $\forall i, j$. In such cases we can extend some of the words in $W(\Psi)$ using $H(\Psi)$ with the goal of generating an ***extended minimal set*** with $|W(\sigma_i)| = \left|W(\sigma_j)\right|$ for all i, j. We note that, without adequate care, this concatenation of words in $W(\Psi)$ may result in a situation where one word becomes a prefix of another, meaning that the codewords are not prefix-free and that the decoder would not be able to instantaneously decode the received sequence. In this section we focus on situations where it is possible to have $|W(\sigma_i)| = \left|W(\sigma_j)\right|$ without causing the prefix problem, and in the next section we extend the construction technique to consider situations where the prefix problem arises.

Example 2: ($(d = 1, k = 3)$ code) Consider the FSM previously shown in Fig. 1. We choose states 1 and 3 as the principal states, which we denote σ_1 and σ_2, respectively. With these states, it follows that $W(\sigma_1) = \{01, 00\}, H(\sigma_1) = \{\sigma_1, \sigma_2\}$ and $W(\sigma_2) = \{01, 1\}, H(\sigma_2) = \{\sigma_1, \sigma_1\}$. We note that this selection of states and codewords satisfies Condition 1 and the prefix condition, therefore instantaneous state-independent decoding is viable. The minimal set is given in tabular form in Table 3.

TABLE 3. The minimal set of a two-state ($d = 1, k = 3$) code.

$W(\sigma_1)$	$H(\sigma_1)$	$W(\sigma_2)$	$H(\sigma_2)$
01	σ_1	01	σ_1
00	σ_2	1	σ_1

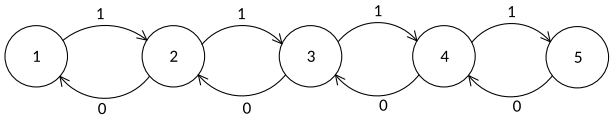


FIGURE 3. FSM of a DC-free code with $N = 5$.

Example 3: (DC-free code with $N = 5$) The FSM of a DC-free sequence with $N = 5$ is shown in Fig. 3. Similar to the previous example, we follow the steps of the construction technique to select states 2 and 4 as the principal states, i.e. $\sigma_1 =$ state 2 and $\sigma_2 =$ state 4. The minimal set of this DC-free code with $N = 5$ is shown in Table 4. As in the example above, this minimal set enables the construction of codes with instantaneous state-independent decoders.

Example 4: (DC-free RLL codes) We also employ the proposed multi-state encoding technique to construct codes that satisfy both DC-free and RLL constraints. We describe the code construction process with an example of a codethat

TABLE 4. The minimal set of a multi-state DC-free code with $N = 5$.

$W(\sigma_1)$	$H(\sigma_1)$	$W(\sigma_2)$	$H(\sigma_2)$
11	σ_2	00	σ_1
10	σ_1	10	σ_2
01	σ_1	01	σ_2

TABLE 5. The minimal set of a DC-free RLL code corresponding to an NRZI encoded ($d = 1, k = 3$) code with $N = 5$.

$W(\sigma_1)$	$H(\sigma_1)$	$W(\sigma_2)$	$H(\sigma_2)$	$W(\sigma_3)$	$H(\sigma_3)$	$W(\sigma_4)$	$H(\sigma_4)$
011	σ_4	011	σ_3	100	σ_2	100	σ_1
11	σ_3	1100	σ_2	00	σ_1	1100	σ_2
		0011	σ_4			0011	σ_4

TABLE 6. The extended minimal set of a DC-free RLL code corresponding to an NRZI encoded ($d = 1, k = 3$) code with $N = 5$.

$W(\sigma_1)$	$H(\sigma_1)$	$W(\sigma_2)$	$H(\sigma_2)$	$W(\sigma_3)$	$H(\sigma_3)$	$W(\sigma_4)$	$H(\sigma_4)$
011	σ_4	011	σ_3	100	σ_2	100	σ_1
1100	σ_1	1100	σ_2	00011	σ_4	1100	σ_2
11100	σ_2	0011	σ_4	0011	σ_3	0011	σ_4

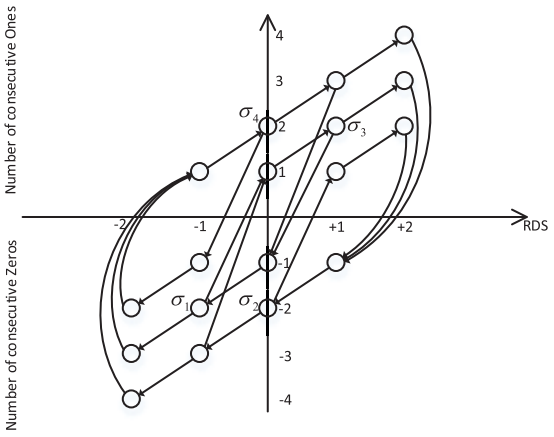


FIGURE 4. FSM of the DC-free RLL constraint corresponding to an NRZI encoded ($d = 1, k = 3$) code with $N = 5$.

limits to $N = 5$ the RDS of the sequence that arises after non-return-to-zero-inverse (NRZI) encoding a sequence that satisfies the $(d = 1, k = 3)$ constraint. We will later present results of other codes with different d, k and N values. The FSM of the $(d = 1, k = 3, N = 5)$ DC-free RLL constraint (after NRZI coding) is shown in Fig. 4, where the coded sequence has runlengths between $d + 1 = 2$ and $k + 1 = 4$, and the RDS is limited to five different values.

We choose four states as the principal states, and refer to them by their locations in the x-y coordinates in Fig. 4, i.e. $\sigma_1 = (-1, -2), \sigma_2 = (0, -2), \sigma_3 = (1, 2), \sigma_4 = (0, 2)$. With this selection of principal states, all loops in the FSM contain at least one principal state, and therefore the minimal set will contain a finite number of words. This ensures that all constraint-satisfying sequences can be generated with concatenation of words in the minimal set. We establish the minimal set shown in Table 5. Note that $|W(\sigma_1)| = |W(\sigma_3)| = 2$, and $|W(\sigma_2)| = |W(\sigma_4)| = 3$. Therefore, we extend some of the words in $W(\sigma_1)$ and $W(\sigma_3)$ by referring to $H(\sigma_1)$ and $H(\sigma_3)$, in order to have the same number of rows in all columns of the minimal set. After extension of the word 11 in $W(\sigma_1)$ and the word 00 in $W(\sigma_3)$, we obtain the extended minimal set as shown in Table 6 where $|W(\sigma_i)| = 3\ \forall i$.

Note that if we use the single-state encoding technique, minimal sets for DC-free constraints with $N \geq 4$ and for most DC-free RLL constraints consist of an infinite number of words, so to be practical, these sets must be truncated to result in sets with a finite number of words. Since valid words are removed from the minimal set, the achievable code rate is reduced, as noted in [26], [27]. However, with the multi-state encoding technique described above, all constraint-satisfying sequences can be generated with the words in the minimal set, hence full capacity can potentially be approached.

C. PARTIAL EXTENSIONS

After obtaining a minimal set or an extended minimal set, we may perform partial extensions to obtain sets of codewords. However, as opposed to the single-state encoding technique where words in a minimal set can be freely concatenated, concatenation as defined in Definition 1 must be performed with multi-state encoding. Therefore, in addition to the words in $W(\Psi)$, we must have knowledge of $H(\sigma_j)_i$ to determine how to extend the word $w(\sigma_j)_i$, $1 \leq j \leq |\Psi|$, $1 \leq i \leq \left|W(\sigma_j)\right|$. A ***partial extension*** in multi-state encoding is the simultaneous extension of words $w(\sigma_j)_i\ \forall j$ for a fixed i, where extension is according to the concatenation of words in Definition 1. Similar to the single-state encoding in Section II-B, a partial extension can be applied to the result of a previous partial extension where the first partial extension starts from the minimal set.

We denote the set of codewords generated through extension of $W(\sigma_j)$ as $\alpha(\sigma_j)$ and the corresponding set of next states as $\beta(\sigma_j)$.[1] The size of each set is denoted as ξ. Note that when a word $w(\sigma_j)_i$ is extended in the table, all the other words in the i-th row that are generated from other principal states are simultaneously extended to ensure $|\alpha(\sigma_1)| = |\alpha(\sigma_2)| = \ldots = \left|\alpha(\sigma_{|\Psi|})\right| = \xi$.

Example 5: ($(d = 1, k = 3, N = 5)$ DC-free RLL code) We perform partial extensions of Table 6 to obtain a set of codewords for the $(d = 1, k = 3, N = 5)$ DC-free RLL constraint. We extend the words $W(\sigma_j)_1, \forall j$ and then extend the words $W(\sigma_j)_2, \forall j$ based on the previous partial extension, by following Definition 1. Our set of codewords is shown in Table 7, where $\xi = 7$.

D. NGH CODING AND CODE RATE EVALUATION

The last step of the encoding technique is to perform NGH coding over the codebook to assign source words to codewords $\alpha(\sigma_1), \alpha(\sigma_2), \ldots, \alpha(\sigma_{|\Psi|})$ in the codebook. To approach capacity, we attempt to approximate the maxentropic probabilities of codewords in the codebook as closely as possible.

[1]When we use the words in the minimal set as the set of codewords directly, for consistency, we still denote $W(\sigma_j)$ as $\alpha(\sigma_j)$ and $H(\sigma_j)$ as $\beta(\sigma_j)$.

TABLE 7. Partial extension of the extended minimal set of a DC-free RLL code corresponding to an NRZI encoded ($d = 1, k = 3$) code with $N = 5$.

$\alpha(\sigma_1)$	$\beta(\sigma_1)$	$\alpha(\sigma_2)$	$\beta(\sigma_2)$	$\alpha(\sigma_3)$	$\beta(\sigma_3)$	$\alpha(\sigma_4)$	$\beta(\sigma_4)$
011100	σ_1	011100	σ_2	100011	σ_3	100011	σ_4
0111100	σ_2	01100011	σ_4	1001100	σ_2	1001100	σ_1
0110011	σ_4	0110011	σ_3	1000011	σ_4	10011100	σ_2
1100011	σ_4	1100011	σ_3	00011100	σ_1	1100011	σ_3
11001100	σ_1	11001100	σ_2	000111100	σ_2	11001100	σ_2
110011100	σ_2	11000011	σ_4	000110011	σ_4	11000011	σ_4
11100	σ_2	0011	σ_4	0011	σ_3	0011	σ_4

TABLE 8. Codebook of a ($d = 1, k = 3$) RLL code with two states and $\eta = 98.91\%$.

Source word	$\alpha(\sigma_1)$	$\beta(\sigma_1)$	$\alpha(\sigma_2)$	$\beta(\sigma_2)$
0	01	σ_1	01	σ_1
1	00	σ_2	1	σ_1

We first obtain the maxentropic transition probabilities of the constraint based on its FSM representation, which is well studied in [1]. Based on the maxentropic transition probabilities, we evaluate the maxentropic probability of each codeword in the final codebook and the steady-state distribution of Ψ. Denote the maxentropic probability of the i-th codeword in α_j as $p(\alpha_j)_i$, the steady-state distribution as $\boldsymbol{\pi} = [\pi(\sigma_1), \pi(\sigma_2), \ldots, \pi(\sigma_{|\Psi|})]$, and the vector of input probabilities to NGH coding as $\boldsymbol{p}_{NGH} = [p_1, p_2, \ldots, p_\xi]$. The desired probability of the i-th source word is then

$$p_i = \sum_{j=1}^{|\Psi|} \pi(\sigma_j) \times p(\alpha_j)_i, \ 1 \le i \le \xi. \tag{7}$$

With this vector of desired input probabilities, NGH coding is performed to generate the corresponding source words.

After constructing the codebook, we must evaluate the average code rate. Assume independent and equiprobable input bits, and let codeword $\alpha(\sigma_j)_i$ be assigned to a source word of length l_i. The probability of occurrence of that codeword when the encoder is in state j is $p(\alpha(\sigma_j)_i) = 2^{-l_i}$. Note that since these probabilities are not in general equal to the maxentropic probabilities, the steady-state probabilities of the principal states are not exactly the probabilities in the steady-state distribution of the FSM. Based on the probability of occurrence of each codeword in the codebook, it is possible to evaluate the steady-state distribution $\tilde{\boldsymbol{\pi}} = [\tilde{\pi}(\sigma_1), \tilde{\pi}(\sigma_2), \ldots, \tilde{\pi}(\sigma_{|\Psi|})]$ of all the principal states Ψ by solving:

$$\tilde{\boldsymbol{\pi}} \boldsymbol{P} = \tilde{\boldsymbol{\pi}} \tag{8}$$

where P is a $|\Psi| \times |\Psi|$ matrix, p_{ji} is the element in j-th row and i-th column and

$$p_{ji} = \sum_{k, \forall h(\sigma_j)_k = \sigma_i} p(\alpha(\sigma_j)_k). \tag{9}$$

TABLE 9. Codebook of a $N = 5$ DC-free code with $\eta = 99.14\%$.

Source word	$\alpha(\sigma_1)$	$\beta(\sigma_1)$	$\alpha(\sigma_2)$	$\beta(\sigma_2)$
00	11	σ_2	00	σ_1
010	0111	σ_2	1000	σ_1
011	0101	σ_1	0101	σ_2
100	0110	σ_1	0110	σ_2
101	1011	σ_2	0100	σ_1
110	1001	σ_1	1001	σ_2
111	1010	σ_1	1010	σ_2

TABLE 10. Codebook of a $N = 5$ DC-free code with ternary source and $\eta = 100\%$.

Source word	$\alpha(\sigma_1)$	$\beta(\sigma_1)$	$\alpha(\sigma_2)$	$\beta(\sigma_2)$
0	11	σ_2	00	σ_1
1	10	σ_1	10	σ_2
2	01	σ_1	01	σ_2

Given the steady-state distribution of the codebook, the average code rate $\bar{R}$ is evaluated as

$$\bar{R} = \frac{\sum_{l_i} l_i \times 2^{-l_i}}{\sum_{l_i} (\sum_{j=1}^{|\Psi|} \tilde{\pi}(\sigma_j) \times o(\sigma_j)_i) \times 2^{-l_i}} \tag{10}$$

where $o(\sigma_j)_i$ is the length of the codeword emitted from state σ_j due to the occurrence of the i-th source word.

Similar to single-state encoding, by performing partial extensions, different codebooks can be generated depending on different concatenations of words in partial extensions. We can establish limits on n_{max} or l_{max}, use an exhaustive search to compare all codebooks that are within these limits, and choose the one that has the highest $\bar{R}$.

Example 6: ($d = 1, k = 3$ RLL code) If we do not perform partial extensions but directly perform NGH coding over the minimal set shown in Table 3, we obtain the simple yet efficient codebook shown in Table 8. Although $|\alpha(\sigma_1)| = |\alpha(\sigma_2)| = 2$, the number of unique codewords in Table 8 is only three. It can be verified that the steady-state distribution for this codebook is $\tilde{\pi} = [\frac{2}{3} \ \frac{1}{3}]$ and that the average code rate is

$$\bar{R} = \frac{1 \times \frac{1}{2} + 1 \times \frac{1}{2}}{\frac{1}{2} \times 2 + \frac{1}{2} \times (\frac{2}{3} \times 2 + \frac{1}{3} \times 1)} = \frac{6}{11}$$

which achieves 98.9% of capacity. Note that this code is as efficient as the single-state code given in Table 1, but that it has shorter codewords and source words. Higher

TABLE 11. Codebook of a DC-free RLL code corresponding to an NRZI encoded ($d = 1, k = 3$) code with $N = 5$, $\eta = 98.09\%$.

Source word	$\alpha(\sigma_1)$	$\beta(\sigma_1)$	$\alpha(\sigma_2)$	$\beta(\sigma_2)$	$\alpha(\sigma_3)$	$\beta(\sigma_3)$	$\alpha(\sigma_4)$	$\beta(\sigma_4)$
0	011	σ_4	011	σ_3	100	σ_2	100	σ_1
10	1100	σ_1	1100	σ_2	00011	σ_4	1100	σ_2
11	11100	σ_2	0011	σ_4	0011	σ_3	0011	σ_4

TABLE 12. Codebook of a DC-free RLL code corresponding to an NRZI encoded ($d = 1, k = 3$) code with $N = 5$, $\eta = 98.47\%$.

Source word	$\alpha(\sigma_1)$	$\beta(\sigma_1)$	$\alpha(\sigma_2)$	$\beta(\sigma_2)$	$\alpha(\sigma_3)$	$\beta(\sigma_3)$	$\alpha(\sigma_4)$	$\beta(\sigma_4)$
000	011100	σ_1	011100	σ_2	100011	σ_3	100011	σ_4
001	0111100	σ_2	01100011	σ_4	1001100	σ_2	1001100	σ_1
010	0110011	σ_4	0110011	σ_3	1000011	σ_4	10011100	σ_2
011	1100011	σ_4	1100011	σ_3	00011100	σ_1	1100011	σ_3
100	11001100	σ_1	11001100	σ_2	000111100	σ_2	11001100	σ_2
101	110011100	σ_2	11000011	σ_4	000110011	σ_4	11000011	σ_4
11	11100	σ_2	0011	σ_4	0011	σ_3	0011	σ_4

TABLE 13. Codebook of a DC-free RLL code with ($d = 2, k = 3, N = 5$), $\eta = 98.62\%$.

Source word	$\alpha(\sigma_1)$	$\beta(\sigma_1)$	$\alpha(\sigma_2)$	$\beta(\sigma_2)$
0	111000	σ_1	000111	σ_2
1	0111	σ_2	1000	σ_1

TABLE 14. Codes constructed that satisfy different DC-free RLL constraints.

Constraint	η	Number of states	Number of source words
$d = 1, k = 3, N = 5$	98.09%	4	3
$d = 2, k = 3, N = 5$	98.62%	2	2
$d = 1, k = 4, N = 6$	97.61%	6	4
$d = 1, k = 5, N = 7$	98.27%	8	5
$d = 2, k = 5, N = 7$	97.66%	6	4

efficiency can be achieved with partial extensions and a larger codebook. To compare, another variable-length coding technique [20] gives a rate 0.5 variable-length ($d = 1, k = 3$) code with efficiency 90.7%, hence demonstrating the effectiveness of our proposed construction technique.

Example 7: (DC-free codes with $N = 5$) Using the minimal set shown in Table 4 as the codebook, we are able to construct a code with efficiency $\eta = 96.46\%$. After performing partial extensions with maximum codeword length $l_{max} = 4$, we obtain the codebook shown in Table 9 which has efficiency $\eta = 99.14\%$. We also note that, as demonstrated in Table 10, with a ternary source it is possible to achieve 100% of capacity simply by using words in the minimal set as the codewords, since the occurrence probability of each codeword is equal to its maxentropic probability.

Example 8: (DC-free RLL codes) Based on the extended minimal set shown in Table 6, if we use the words in the extended minimal set as the codewords, the average code rate is $\bar{R} = 0.4167$ and $\eta = 98.09\%$. The corresponding codebook is shown in Table 11. Based on the partial extension in Table 7, we construct the codebook with $\bar{R} = 0.4183$ and $\eta = 98.47\%$ shown in Table 12. Further improvement of efficiency can be obtained via partial extensions with larger l_{max} and/or n_{max}.

In Table 13 we present a very simple, but highly efficient, multi-state code for the ($d = 2, k = 3, N = 5$) constraint. In Table 14 we list parameters of other DC-free RLL codes that we have constructed. Note that still higher efficiencies can be achieved for these values of d, k and N with larger codebooks.

To compare, a state-of-the-art fixed-length code construction technique for the ($d = 1, k = 3, N = 5$) DC-free RLL constraint gives a rate 0.4 code with $\eta = 94.16\%$, and the codebook has 18 states, 256 source words and hundreds of codewords [31]. As shown in Tables 11 and 14, however, our proposed construction gives a codebook with only 4 states, 3 source words and 6 different codewords, and has efficiency $\eta = 98.09\%$. [31] also proposed a ($d = 1, k = 5, N = 7$) DC-free RLL code with 20 states and 16 source words that results in an efficiency of $\eta = 90.96\%$. Our construction gives a code with 8 states, 5 source words and $\eta = 98.27\%$ demonstrating that our proposed variable-length construction technique can significantly reduce the complexity and improve the efficiency of constrained sequence codes. The codebook is shown in Table 15. Other examples of DC-free RLL codes we constructed are summarized in Table 14.

As is evident in the above examples, all the codes we have presented (and the codes we construct in the rest of this paper) are state-independently decodable. This requires attention during the construction process. For instance, consider the minimal set where $W_r \in W(\sigma_u) \cap W(\sigma_y)$ and $W_r \notin W(\sigma_v)$. This indicates that state σ_v cannot generate W_r because of the constraint described by the FSM. But with an appropriate selection of principal states, state σ_v would generate another word W_s such that $W_s \notin W(\sigma_u)$ and $W_s \notin W(\sigma_y)$. State-independent code design would have W_r and W_s constitute a row, thus satisfying Condition 1. For example in Table 5, $w(\sigma_1)_1$ and $w(\sigma_2)_1$ are 011 while $w(\sigma_3)_1$ and $w(\sigma_4)_1$ are 100, where it is clear that states σ_1 and σ_2 can generate 011 but not 100, and vice versa for states σ_3 and σ_4. Words 100 and 011 constitute the first row, and Condition 1 is satisfied. Given a minimal set that has the state-independent decoding property, it is readily seen that

TABLE 15. Codebook of a DC-free RLL code with ($d = 1, k = 5, N = 7$), $\eta = 98.27\%$.

Source word	$\alpha(\sigma_1)$	$\beta(\sigma_1)$	$\alpha(\sigma_2)$	$\beta(\sigma_2)$	$\alpha(\sigma_3)$	$\beta(\sigma_3)$	$\alpha(\sigma_4)$	$\beta(\sigma_4)$
11	011	σ_2	100	σ_1	011	σ_8	011	σ_6
01	1100	σ_1	0000011	σ_6	1100	σ_3	1100	σ_4
00	0011	σ_8	0011	σ_2	0011	σ_5	111100	σ_3
101	00011	σ_5	00011	σ_8	00011	σ_6	1111100	σ_1
100	000011	σ_6	000011	σ_5	11100	σ_1	11100	σ_7

Source word	$\alpha(\sigma_5)$	$\beta(\sigma_5)$	$\alpha(\sigma_6)$	$\beta(\sigma_6)$	$\alpha(\sigma_7)$	$\beta(\sigma_7)$	$\alpha(\sigma_8)$	$\beta(\sigma_8)$
11	100	σ_8	100	σ_4	011	σ_5	100	σ_3
01	1100	σ_3	1100	σ_8	1100	σ_8	110	σ_1
00	0011	σ_5	0011	σ_6	0011	σ_6	0011	σ_7
101	00011	σ_6	111100	σ_1	111100	σ_1	00011	σ_5
100	11100	σ_1	11100	σ_3	11100	σ_3	000011	σ_6

codebooks constructed through its partial extensions can be state-independently decoded.

IV. MULTI-STATE ENCODING BASED ON *N*-step FSM

As demonstrated above, different principal states may have a different number of words, i.e. there may exist i, j such that $|W(\sigma_i)| \neq |W(\sigma_j)|$. This will cause ***imbalance*** in the number of words in different states in the minimal set, i.e., the number of words in different states is unequal, and hence causes possibly a different number of codewords associated with different states in the codebook after partial extensions. Although as we illustrated in the previous section, it may be possible to establish an extended minimal set where $|W(\sigma_i)| = |W(\sigma_j)|$ by extending some words, this approach does not apply for all constraints. For example, with a DC-free $N = 6$ constraint, if we select states 2, 4 and 6 as principal states, i.e. $\sigma_1 = 2, \sigma_2 = 4, \sigma_3 = 6$, we have $W(\sigma_1) = \{01, 10, 11\}$, $W(\sigma_2) = \{01, 10, 11, 00\}$, and $W(\sigma_3) = \{01, 00\}$. A feasible codebook requires that the number of codewords in $\alpha(\sigma_i)$, $1 \leq i \leq |\xi|$ to be the same. If we choose to concatenate words in $W(\sigma_3)$ to compensate for the imbalance, some words in $W(\sigma_1)$ and $W(\sigma_2)$ become prefixes of words in $W(\sigma_3)$, and after partial extensions, some codewords in $\alpha(\sigma_1)$ and $\alpha(\sigma_2)$ become prefixes of words in $\alpha(\sigma_3)$, eliminating the possibility of state-independent decoding. Alternatively, it is possible to eliminate words from $W(\sigma_1)$ and $W(\sigma_2)$ to force the same number of words in all the principal states, however this will result in rate loss.

In this section we extend our construction technique to include the use of n-step FSMs, and illustrate this extension with DC-free codes because of their importance in recently-developed VLC systems. In the next section we consider a special case of n-step FSMs for DC-free codes that can result in even further enhancement.

A. *n*-STEP FSM

An n-step FSM describes transitions among states where the edge labels represent the concatenation of n successive edges of the initial FSM. For example, when the initial FSM contains edge labels of a single symbol, the labels in the n-step graph have length n. The adjacency matrix of an n-step FSM is D^n and the n-step transition matrix is Q^n. The asymptotic steady-state distribution $\boldsymbol{\pi}$ of a n-step FSM is the same as that of the initial FSM.

B. PRINCIPAL STATES AND MINIMAL SETS

Given the n-step FSM, the general code construction procedure is similar to the one introduced above. The concatenation of words and selection of principal states is the same as that introduced in Section III-A. Should the number of words in the principal states be unequal, we perform concatenation over some of the words in $W(\Psi)$ according to Definition 1 in an attempt to construct an extended minimal set with the same number of words in each principal state. Care must be taken in this step to ensure that the prefix condition is maintained, and similarity in the number of words is improved. For example, it might occur that $w_r \in W(\sigma_1)$ and $w_r \in W(\sigma_2)$, where $|W(\sigma_1)| < |W(\sigma_2)|$. If we concatenate w_r only in $W(\sigma_1)$ to increase the number of words in state σ_1, w_r in $W(\sigma_2)$ will become a prefix of some words in $W(\sigma_1)$. If we concatenate w_r in both $W(\sigma_1)$ and $W(\sigma_2)$, the inequality in number of words might become more pronounced. To address this problem, we must choose an appropriate value of n such that some words can be concatenated without those problems occurring in the new minimal set, which we call the n-step minimal set.

We observe that in n-step FSMs of DC-free codes with N RDS values, state 1 and state N have fewer words than state $\lfloor \frac{N}{2} \rfloor$. Note that with $n = N - 1$, the all-one word of length $N - 1$ is generated by state 1 and the all-zero word of length $N - 1$ is generated by state N. In addition, those two words do not occur in any other states in the minimal set. Therefore, it is possible to concatenate those two words with other words in the minimal set according to Definition 1 to compensate for the imbalance of words without causing the prefix problem to arise.

This observation that the imbalance of words in the n-step minimal set can be reduced also holds for n-step FSMs with $n = N - 2$ where the all-one word is generated by states 1 and 2 and the all-zero word is generated by states $N - 1$ and $N - 2$. It is straightforward to verify that this observation applies when n is in the range

$$n = [\left\lceil \frac{N}{2} \right\rceil, \quad N - 1]. \tag{11}$$

TABLE 16. The minimal set of a 3-step DC-free code with $N = 6$.

$W(\sigma_1)$	$H(\sigma_1)$	$W(\sigma_2)$	$H(\sigma_2)$	$W(\sigma_3)$	$H(\sigma_3)$	$W(\sigma_4)$	$H(\sigma_4)$	$W(\sigma_5)$	$H(\sigma_5)$	$W(\sigma_6)$	$H(\sigma_6)$
101	σ_2	101	σ_3	101	σ_4	101	σ_5	101	σ_6	010	σ_5
110	σ_2	110	σ_3	110	σ_4	110	σ_5	001	σ_4	001	σ_5
111	σ_4	011	σ_3	011	σ_4	011	σ_5	011	σ_6	000	σ_3
		010	σ_1	010	σ_2	010	σ_3	010	σ_4		
		100	σ_1	100	σ_2	100	σ_3	100	σ_4		
		111	σ_5	001	σ_2	001	σ_3	000	σ_2		
				111	σ_6	000	σ_1				

TABLE 17. An extended 3-step minimal set of a DC-free code with $N = 6$, $n = 3$.

$W_c(\alpha_1)$	$H_c(\alpha_1)$	$W_c(\alpha_2)$	$H_c(\alpha_2)$	$W_c(\alpha_3)$	$H_c(\alpha_3)$	$W_c(\alpha_4)$	$H_c(\alpha_4)$	$W_c(\alpha_5)$	$H_c(\alpha_5)$	$W_c(\alpha_6)$	$H_c(\alpha_6)$
101	σ_2	101	σ_3	101	σ_4	101	σ_5	101	σ_6	000010	σ_2
111101	σ_5	010	σ_1	010	σ_2	010	σ_3	010	σ_4	010	σ_5
110	σ_2	110	σ_3	110	σ_4	110	σ_5	000011	σ_3	000011	σ_3
111100	σ_3	111100	σ_4	001	σ_2	001	σ_3	001	σ_4	001	σ_5
111011	σ_5	100	σ_1	100	σ_2	100	σ_3	100	σ_4	000100	σ_2
111110	σ_5	011	σ_3	011	σ_4	011	σ_5	011	σ_6	000001	σ_2
111010	σ_3	111010	σ_4	111010	σ_5	000101	σ_2	000101	σ_3	000101	σ_3
111001	σ_3	111001	σ_4	111001	σ_5	000110	σ_2	000110	σ_3	000110	σ_3
111000	σ_1	111000	σ_2	111000	σ_3	111000	σ_4	111000	σ_5	111000	σ_6
		111101	σ_6					000010	σ_1		
		111011	σ_6					000100	σ_1		

From this range, we select the n that results in the highest achievable code rate, as will be discussed in the next subsection.

Example 9: (DC-free $N = 6$ code) Consider the construction of an n-step minimal set for the DC-free $N = 6$ constraint. Using (11), we obtain the range of n as $[3, 5]$. Selecting all states as principal states, the minimal set for the 3-step FSM is shown in Table 16.

It is clear from this table that there is an unequal number of words in the states in this minimal set. As has been discussed, if we extend words in this table without due care, we may violate the prefix condition or cause greater imbalance to arise. For example, if we extend the word 101 in $W(\sigma_1)$, all other occurrences of the word 101 in the same row should be extended as well, otherwise they will become prefixes of the newly concatenated word. However, the extension of the word 101 in other columns in this row will make the imbalance more severe.

Motivated by the observation above, we perform concatenation of the all-one words in state $\sigma_1, \sigma_2, \sigma_3$ and of the all-zero words in state $\sigma_4, \sigma_5, \sigma_6$. The resulting table is shown in Table 17.

C. PRUNING AND ACHIEVABLE CODE RATE

Careful extension of words should reduce the imbalance between the number of words from different states while ensuring that the prefix condition remains satisfied. However, should an inequality among states remain, it is possible to truncate some of the words to obtain a pruned version of the extended minimal set that has the same number of words in each state. We denote this pruned set as $W^p(\Psi)$.

TABLE 18. Achievable code rates of different n values of n-step FSMs.

	$n = 3$	$n = 4$	$n = 5$
$\tilde{C}_n$	0.8448	0.8475	0.8331
$\tilde{\eta}_n$	99.45%	99.76%	98.07%

The number of words that must be truncated from state σ_j, denoted $u(\sigma_j)$, is

$$u(\sigma_j) = \left|W(\sigma_j)\right| - \min\{|W(\sigma_1)|, |W(\sigma_1)|, \ldots, \left|W(\sigma_{|\psi|})\right|\}. \tag{12}$$

Given Q^n and D^n, we can evaluate the probability of the i-th word in $W(\sigma_j)$, $1 \le j \le |\Psi|$, which we denote $p(W(\sigma_j))_i$. Then, $u(\sigma_j)$ words with the lowest probabilities in $W(\sigma_j)$ are eliminated. We denote the set of words in each state of $W^p(\Psi)$ as $W^p(\sigma_i)$, $1 \le i \le |\Psi|$, and the set of next states corresponding to words in $W^p(\sigma_i)$ as $H^p(\sigma_i)$.

Since some words that satisfy the constraint are not used, we are not able to achieve full capacity. The achievable code rate of $W^p(\Psi)$, denoted as $\tilde{C}_n$, is given by (13), as shown at the bottom of this page where $l(W^p(\sigma_k))_v$

$$\tilde{C}_n = \frac{\sum_{k=1}^{|\Psi|} \tilde{\pi}^p(\sigma_k) \times \left(\sum_{v=1}^{|W^p(\sigma_j)|-u(\sigma_j)} -p(W^p(\sigma_k))_v \times \log_2 p(W^p(\sigma_k))_v \right)}{\sum_{k=1}^{|\Psi|} \tilde{\pi}^p(\sigma_k) \times \left(\sum_{v=1}^{|W^p(\sigma_j)|-u(\sigma_j)} p(W^p(\sigma_k))_v \times l(W^p(\sigma_k))_v \right)} \tag{13}$$

TABLE 19. Codebook of a pruned version of the extended 3-step minimal set of a DC-free code with $N = 6$.

Source word	$\alpha(\sigma_1)$	$\beta(\sigma_1)$	$\alpha(\sigma_2)$	$\beta(\sigma_2)$	$\alpha(\sigma_3)$	$\beta(\alpha_3)$	$\alpha(\sigma_4)$	$\beta(\sigma_4)$	$\alpha(\sigma_5)$	$\beta(\sigma_5)$	$\alpha(\sigma_6)$	$\beta(\sigma_6)$
000	101	σ_2	101	σ_3	101	σ_4	101	σ_5	101	σ_6	000010	σ_2
001	110	σ_2	110	σ_3	110	σ_4	110	σ_5	000011	σ_3	000011	σ_3
10	111101	σ_5	010	σ_1	010	σ_2	010	σ_3	010	σ_4	010	σ_5
110	111110	σ_5	100	σ_1	100	σ_2	100	σ_3	100	σ_4	000100	σ_2
010	111011	σ_5	011	σ_3	011	σ_4	011	σ_5	011	σ_6	000001	σ_2
11100	111010	σ_3	111010	σ_4	111010	σ_5	000101	σ_2	000101	σ_3	000101	σ_3
011	111100	σ_3	111100	σ_4	001	σ_2	001	σ_3	001	σ_4	001	σ_5
1111	111001	σ_3	111001	σ_4	111001	σ_5	000110	σ_2	000110	σ_3	000110	σ_3
11101	111000	σ_1	111000	σ_2	111000	σ_3	111000	σ_4	111000	σ_5	111000	σ_6

TABLE 20. Highest average code rates of DC-free code with $N = 6$ codebooks with different size.

n_{max}	25	33	41
R	0.801	0.8047	0.8054
η	94.29%	94.72%	94.81%

denotes the length of the v-th word in $W^p(\sigma_k)$, and $\widetilde{\boldsymbol{\pi}}^p = [\widetilde{\pi}^p(\sigma_1), \widetilde{\pi}^p(\sigma_2), \ldots, \widetilde{\pi}^p(\sigma_{|\Psi|})]$ is the steady-state distribution of $W^p(\Psi)$. Note that $\widetilde{\boldsymbol{\pi}}^p$ is different from the steady-state distribution of the initial FSM of the constraint. It is evaluated similar to (8), but with $p(\alpha(\sigma_j))_i$ in (8) replaced with $p(W^p(\sigma_j))_i$. After evaluating $\widetilde{C}_n$, we choose to work with the $W^p(\Psi)$ with the highest $\widetilde{C}_n$ and the highest achievable efficiency $\widetilde{\eta}_n = \widetilde{C}_n/C$.

Example 10: (DC-free $N = 6$ code) Based on (13), using the approach outlined in Sections IV-B and IV-C where the all-one and all-zero words are extended and $u(\sigma_j)$ words are pruned, the achievable code rates of different n-step FSMs are shown in Table 18. Note that although for illustration we use $n = 3$ as an example throughout this section, $\widetilde{C}_n$ is highest with $n = 4$.

D. ENCODING

Since we now have $W^p(\Psi)$ which contains the same number of words in each state, we can perform partial extensions and NGH coding to obtain the codebook in a manner similar to that introduced in Section III-D. As above, the evaluation of the average code rate is given by (10). Within predetermined limits on n_{max} and/or l_{max}, an exhaustive search can be performed to determine the codebook with the highest $\bar{R}$.

Example 11: (DC-free $N = 6$ code) Based on Table 17 and (12), the words 111101 and 111011 from state σ_2, and the words 000010 and 000100 from state σ_5, are removed to result in an equal number of words in all states in the 3-step minimal set. If we use this 3-step minimal set as the codebook, and perform NGH coding to obtain the assignment of source words, we construct the codebook shown in Table 19 that has an efficiency of 92.8%.

By performing partial extensions over this 3-step minimal set, we are able to construct codebooks with higher average code rates. Some results are listed in Table 20.

Lastly, we note that the construction process introduced in this section can be used for a variety of constraints, and in some instances can result in a codebook with few principal states, or an extended minimal set with an equal number of words in all states so that so pruning is not needed. In the next section we focus on DC-free codes for VLC systems, and we show that appropriately designed DC-free codebooks can benefit from both of these conditions.

V. CASE STUDY: DC-FREE CODES FOR VLC

A. BACKGROUND, 4B6B AND 8B10B CODES

Visible light communication (VLC) that provides short-range free-space data transmission with light-emitting diodes has recently attracted much attention [13]–[18]. On-off keying (OOK) that represents binary data with the presence or absence of light pulses is commonly used in VLC systems due to its simplicity. In these systems, the brightness of the light is affected by the distribution of ones and zeros in the transmitted symbol sequence. Moreover, flicker is affected by the length of consecutive ones and zeros in the transmitted codewords and can be mitigated by limiting the runlength in the coded sequence. DC-free codes have also found applications in VLC, where DC-free 4B6B and 8B10B codes have been adopted in the standard to reduce flicker perception and adjust dimming control [13]. These codes ensure an equal number of ones and zeros in the transmitted symbol sequence which helps maintain the dimming level. DC-free codes also have an inherent RLL limit and therefore assist with flicker mitigation. As noted earlier, the maximum runlength of coded ones and zeros in DC-free codes with N different running digital sum (RDS) values is limited to $N - 1$.

The 4B6B code satisfies the DC-free constraint with $N = 5$; the codebook has 16 source words where each source word of length 4 is mapped to a codeword of length 6, resulting in the code rate $R = 4/6$ [13]. The capacity of the $N = 5$ constraint is 0.7925 [1], therefore the efficiency of the 4B6B code is $\eta = R/C = 84.12\%$.

The 8B10B codes are a class of rate $R = 8/10$ DC-free codes with $N = 6$ or $N = 7$, which are constraints with capacity 0.8495 and 0.8858 respectively. A survey of 8B10B codes can be found in [1]. The $N = 7$ code in [36] has gained considerable attention due to its structure which simplifies implementation.

B. CONSTRAINED SEQUENCE CODING FOR VLC

In Section III we showed that, with our proposed encoding method, codebooks that satisfy the DC-free $N = 5$ constraint

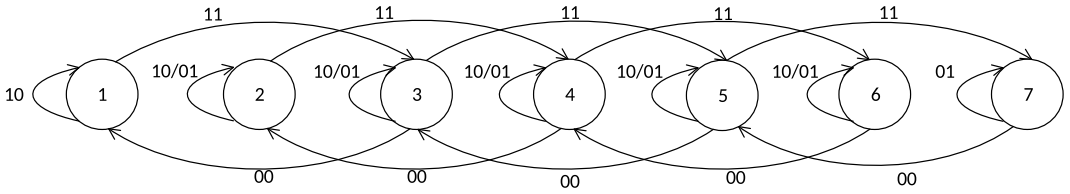


FIGURE 5. 2-step FSM of the DC-free constraint with $N = 7$.

can be constructed with over 99% efficiency and with fewer codewords than the 4B6B code noted above. Therefore, our codes are superior in terms of both efficiency and implementation complexity. In this subsection, we now focus on coding for the DC-free constraint with $N = 7$, and compare our results with the 8B10B codes.

We present codes constructed for VLC systems based on n-step FSMs. We show that based on n-step FSMs with an even n, as introduced in Section V, the number of principal states in a DC-free code can be reduced to $N/2$ when N is even and either $\lfloor N/2 \rfloor$ or $\lceil N/2 \rceil$ when N is odd, and that an extended minimal set with an equal number of words in each state can be obtained such that $\widetilde{\eta} = 100\%$.

The reduction of states is based on the observation that with DC-free constraints, when n is even, the n-step edge graphs subdivide into two non-intersecting FSMs. An example of this phenomenon is shown in Fig. 5, where it is evident that in this 2-step edge graph of the DC-free constraint with $N = 7$, the FSM comprised of the even-numbered states is not connected to the FSM comprised of the odd-numbered states. However, as discussed in [37], each of these smaller FSMs generate all constraint satisfying sequences, and therefore either one can be used as the basis for our variable-length code design. It can also be verified that the 2-step FSM comprised of the three even-numbered states has the steady-state probability distribution $\boldsymbol{\pi} = [0.2929.0.4142, 0.2929]$, whereas the 2-step FSM of the four odd-numbered states has the steady-state probability distribution $\boldsymbol{\pi} = [0.1464, 0.3536, 0.3536, 0.1464]$.

Following the construction technique introduced in Section V with the 2-step FSM, we now consider the construction process specifically for DC-free codes with any value of N. We show that it is always possible to construct an extended minimal set with a maximum achievable efficiency $\widetilde{\eta} = 100\%$ with only $N/2$ principal states when N is even, and with $\lfloor N/2 \rfloor$ or $\lceil N/2 \rceil$ principal states when N is odd, depending on whether we work with the set of even-numbered states or the set of odd-numbered states. We begin with the following example for $N = 7$ with the set of even-numbered states.

Example 12: (DC-free codes with $N = 7$) When we consider the set of even-numbered states, the minimal set of the 2-step FSM is shown in Table 21. The achievable code rate of this codebook is 0.8858, which is the capacity of DC-free constraint with $N = 7$, confirming that all constraint-satisfying sequences are generated by this three-state FSM.

TABLE 21. A 2-step minimal set of DC-free code $N = 7$.

$W(\sigma_2)$	$H(\sigma_2)$	$W(\sigma_4)$	$\beta(\sigma_4)$	$H(\sigma_6)$	$W(\sigma_6)$
10	σ_2	10	σ_4	10	σ_6
01	σ_2	01	σ_4	01	σ_6
11	σ_4	11	σ_6	00	σ_4
		00	σ_2		

TABLE 22. An extended 2-step minimal set of DC-free code $N = 7$.

$W(\sigma_2)$	$H(\sigma_2)$	$W(\sigma_4)$	$\beta(\sigma_4)$	$H(\sigma_6)$	$W(\sigma_6)$
10	σ_2	10	σ_4	10	σ_6
01	σ_2	01	σ_4	01	σ_6
1110	σ_4	1110	σ_6	0010	σ_4
1101	σ_4	1101	σ_6	0001	σ_4
1111	σ_6	1100	σ_4	0011	σ_6
1100	σ_2	0010	σ_2	0000	σ_2
		0001	σ_2		
		0011	σ_4		

TABLE 23. A DC-free $N = 7$ codebook, $\eta = 95.53\%$.

Source words	$\alpha(\sigma_2)$	$\beta(\sigma_2)$	$\alpha(\sigma_4)$	$\beta(\sigma_4)$	$\alpha(\sigma_6)$	$\beta(\sigma_6)$
10	10	σ_2	10	σ_4	10	σ_6
11	01	σ_2	01	σ_4	01	σ_6
0111	111101	σ_6	0010	σ_2	0010	σ_4
001	111100	σ_4	0011	σ_4	0011	σ_6
0110	111110	σ_6	0001	σ_2	0001	σ_4
0101	1101	σ_4	1101	σ_6	000010	σ_2
000	1100	σ_2	1100	σ_4	000011	σ_4
0100	1110	σ_4	1110	σ_6	000001	σ_2

TABLE 24. A 2-step minimal set of DC-free code $N = 7$ with the set of odd states as principal states.

$W(\sigma_1)$	$H(\sigma_1)$	$W(\sigma_3)$	$\beta(\sigma_3)$	$H(\sigma_5)$	$W(\sigma_5)$	$H(\sigma_7)$	$W(\sigma_7)$
10	σ_1	10	σ_3	10	σ_5	01	σ_7
11	σ_3	01	σ_3	01	σ_5	00	σ_5
		11	σ_5	11	σ_7		
		00	σ_1	00	σ_3		

We extend the words 11 in $W(\sigma_2), W(\sigma_4)$, and 00 in $W(\sigma_4), W(\sigma_6)$ by tracing the edges corresponding to 11 and 00 to construct Table 22. Then, we extend word 1111 in $W(\sigma_2)$, and 0000 in $W(\sigma_6)$ by once again tracing the edges corresponding to 11 and 00, and obtain an extended minimal set with $\xi = 9$ without causing the prefix problem. Note that with this extended minimal set, the achievable efficiency is 100% since no pruning is performed. If we use this minimal set as the codebook and perform the encoding procedure as outlined in Section V, we obtain the codebook in Table 23 with $\bar{R} = 0.8462$ and $\eta = 95.53\%$. By performing partial extensions with $n_{max} = 15$, we have constructed a codebook with $\bar{R} = 0.8535$ and $\eta = 96.35\%$.

TABLE 25. A DC-free $N = 7$ codebook, $\eta = 94.88\%$.

Source words	$\alpha(\sigma_1)$	$\beta(\sigma_1)$	$\alpha(\sigma_3)$	$\beta(\sigma_3)$	$\alpha(\sigma_5)$	$\beta(\sigma_5)$	$\alpha(\sigma_7)$	$\beta(\sigma_7)$
10	10	σ_1	10	σ_3	10	σ_5	000001	σ_3
010	1110	σ_3	1110	σ_5	0001	σ_3	0001	σ_5
0110	1101	σ_3	1101	σ_5	1101	σ_7	00000010	σ_1
0111	1100	σ_1	1100	σ_3	1100	σ_5	00000011	σ_3
00	111110	σ_5	01	σ_3	01	σ_5	01	σ_7
11000	111101	σ_5	111101	σ_7	000010	σ_1	000010	σ_3
11001	111100	σ_3	111100	σ_5	000011	σ_3	000011	σ_5
1101	11111101	σ_7	0010	σ_1	0010	σ_3	0010	σ_5
111	11111100	σ_5	0011	σ_3	0011	σ_5	0011	σ_7

Example 13: (DC-free codes with $N = 7$) We now consider the construction of a codebook with the set of odd-numbered states as principal states. The minimal set is shown in Table 24, where the principal states are $\sigma_1, \sigma_3, \sigma_5, \sigma_7$. As in the example above, we perform extensions by tracing the edges corresponding to 11 and 00 to obtain an extended minimal set with an equal number of words in each principal state. By performing NGH coding over this extended minimal set, we obtain a codebook with $\bar{R} = 0.8405$ and $\eta = 94.88\%$, which is shown in Table 25. By performing partial extensions with $n_{max} = 17$, we have constructed a codebook with $\bar{R} = 0.8468$ and $\eta = 95.59\%$.

While the above examples demonstrate the construction of extended minimal sets without pruning when N is odd, it is straightforward to verify that this approach can also be used with 2-step FSMs when N is even. In all cases, the all-one and all-zero words are extended until there are an equal number of words associated with each state. The fact that this is always possible is given in the proof of the following theorem.

Theorem 1: For DC-free constraints with any N, we can obtain an extended minimal set with an equal number of words in all states based on extension of the all-zero and all-one words, where only $N/2$ states are selected as principal states when N is even, and when either $\lfloor N/2 \rfloor$ or $\lceil N/2 \rceil$ states are selected as principal states when N is odd. These codes have achievable efficiency $\widetilde{\eta} = 100\%$, and are instantaneously decodable.

Proof: See Appendix A. ■

Recall that the 8B10B code employed in VLC has $R = 0.8$. Tables 23 and 25 present simple codes with code rates $\bar{R} = 0.8462$ and $\bar{R} = 0.8405$, respectively. With similar high code rates, the codes proposed with the single-state variable-length coding scheme in [27] include significantly more and longer words. Thus with multi-state encoding, we can construct codes with fewer and shorter codewords to satisfy DC-free constraints for VLC.

VI. CONCLUSION

We have proposed a generalized framework to construct multi-state variable-length constrained sequence codes that have capacity-approaching code rates and can be decoded with state-independent decoding. We first introduced the definition of concatenation and principal states based on an FSM description of the constraint. We then discussed the code construction process which includes establishing the minimal set, performing partial extensions, and NGH coding. Furthermore, we extended the proposed construction process to n-step FSMs to overcome an unequal number of words between states in minimal sets in some constraints. We then designed DC-free codes specifically for VLC systems based on n-step FSMs. Examples were given to show that codes satisfying a variety of constraints, including the DC-free constraint that is employed in VLC, can be constructed with high efficiency and low implementation complexity, compared to many codes in the literature.

ACKNOWLEDGMENT

The authors are grateful to the anonymous reviewers for their comments that resulted in improvement of this paper.

APPENDIX A
PROOF OF THEOREM 1

Proof: We consider 2-step FSMs, and consider odd and even N separately.

i) We first consider odd N with the set of $\lfloor \frac{N}{2} \rfloor$ even-numbered states as principal states, i.e. $\Psi = \{\sigma_2, \sigma_4, \ldots, \sigma_{N-1}\}$. It is readily seen that the number of words in each state in Ψ is $N_\Psi = \{3, 4, 4, \ldots, 4, 3\}$, because for a state $\sigma_j \in \{\sigma_4, \sigma_6, \ldots, \sigma_{N-3}\}$, $W(\sigma_j) = \{01, 10, 00, 11\}$, and for the other two states, $W(\sigma_2) = \{01, 10, 11\}$ and $W(\sigma_{N-1}) = \{01, 10, 00\}$. Starting from a state σ_j, another state $\sigma_i, i \in \{j+2, j-2\}$ is reached in a single extension with label 11 (when $i > j$) or 00 (when $i < j$). With each extension we reach another state in Ψ. Since $|\Psi| = \lfloor N/2 \rfloor$, the maximum number of extensions that result in the all-one or all-zero sequence is $\lfloor N/2 \rfloor - 1$.

We denote $\Delta_{N_{\sigma_j}}$ as the number of new words generated from an extension of the edge with label 11 or 00. Consider $\sigma_j = \sigma_2$, and consider the number of words that can occur as an extension of the edge 11. Since that edge has reached state σ_4, when $4 < N - 1$ there are four possible words: 1101, 1110, 1100, 1111 since $W(\sigma_4) = \{01, 10, 00, 11\}$, and hence $\Delta_{N_{\sigma_j}} = 4$. Since the word 1111 has reached state σ_6, when $6 < N - 1$ there are four extended words 111101, 111110, 111100, 111111, hence $\Delta_{N_{\sigma_j}} = 4$. Continuing in this manner, it can be deduced that in the first $\lfloor N/2 \rfloor - 2$ extensions, $\Delta_{N_{\sigma_j}} = 4$. In extension number $\lfloor N/2 \rfloor - 1$, however, $\Delta_{N_{\sigma_j}} = 3$ since it reaches state σ_{N-1} where $|W(\sigma_{N-1})| = 3$, and the extended words do not include

the all-one word. Therefore, the total number of words N_{σ_j} in state σ_j once the all-one word is no longer in the set is:

$$\begin{aligned} N_{\sigma_2} &= \sum_{k=2,4,6,\ldots,N-1} \Delta_{N_{\sigma_k}} \\ &= 3 + 4(\lfloor N/2 \rfloor - 2) + 3 - (\lfloor N/2 \rfloor - 1) \\ &= 3\lfloor N/2 \rfloor - 1. \end{aligned} \tag{14}$$

Similar analysis holds for σ_{N-1}. The first extension of the edge 00 from σ_{N-1} results in four extended words 0001, 0010, 0011, 0000 since $W(\sigma_{N-3}) = \{01, 10, 11, 00\}$, hence $\Delta_{N_{\sigma_j}} = 4$. It can be deduced that in the first $\lfloor N/2 \rfloor - 2$ extensions $\Delta_{N_{\sigma_j}} = 4$, and the all-zero word remains in the set. In extension number $\lfloor N/2 \rfloor - 1$, $\Delta_{N_{\sigma_j}} = 3$, and this is the first extension that does not include the all-zero word. Therefore, the total number of words N_{σ_j} in state σ_j once the all-zero word no longer appears in this state is also $N_{\sigma_j} = 3\lfloor N/2 \rfloor - 1$.

For $\sigma_j \in \{\sigma_4, \sigma_6, \ldots, \sigma_{N-3}\}$, both 11 and 00 in are traced during extensions. It can be verified that the number of extensions of the all-one word is $\frac{N-1}{2} - \frac{j}{2}$, where $\Delta_{N_{\sigma_j}} = 4$ in the first $\frac{N-1}{2} - \frac{j}{2} - 1$ extensions and $\Delta_{N_{\sigma_j}} = 3$ in the last extension, since it has reached state σ_{N-1}. Similarly, the number of extensions of the all-zero word is $\frac{j}{2} - 1$ where $\Delta_{N_{\sigma_j}} = 4$ in the first $\frac{j}{2} - 2$ extensions and $\Delta_{N_{\sigma_j}} = 3$ in the last extension, since it has reached state σ_2. Therefore, the total number of words N_{σ_j} in state σ_j once the all-one and the all-zero words are no longer in the set is:

$$\begin{aligned} N_{\sigma_j} = \; & 4 + 4(\lfloor N/2 \rfloor - \frac{j}{2} - 1) \\ & + 3 + 4(\frac{j}{2} - 2) + 3 - (\lfloor N/2 \rfloor - 1) \\ = \; & 3(\lfloor N/2 \rfloor) - 1. \end{aligned} \tag{15}$$

Thus if all principal states are extended just to the point where they no longer contain either the all-zero or all-one words, each of the principal states $\Psi = \{\sigma_2, \sigma_4, \ldots, \sigma_{N-1}\}$ have $3(\lfloor N/2 \rfloor) - 1$ words in the extended minimal set, and hence $\widetilde{\eta} = 100\%$ since no pruning is required to construct a set in which all principal states have the same number of words.

ii) When we choose odd-numbered N with the set of odd states, similar to the above analysis, the total number of words N_{σ_j} in state $\sigma_j \in \{\sigma_1, \sigma_N\}$ once the all-one or all-zero words are no longer in the set is

$$\begin{aligned} N_{\sigma_j} &= 2 + 4(\lceil N/2 \rceil - 2) + 2 - (\lceil N/2 \rceil - 1) \\ &= 3\lceil N/2 \rceil - 3, \end{aligned} \tag{16}$$

and the total number of words N_{σ_j} in state $\sigma_j \in \{\sigma_3, \sigma_5, \ldots, \sigma_{N-2}\}$ once the all-one or all-zero word is no longer in the set is

$$\begin{aligned} N_{\sigma_j} = \; & 4 + 4(\lceil N/2 \rceil - \frac{j+1}{2} - 1) \\ & + 2 + 4(\frac{j+1}{2} - 2) + 2 - (\lceil N/2 \rceil - 1) \\ = \; & 3(\lceil N/2 \rceil) - 3. \end{aligned} \tag{17}$$

Therefore all states have $3(\lceil N/2 \rceil) - 3$ words, and $\widetilde{\eta} = 100\%$.

iii) Similarly, when we choose even N with either the set of even-numbered or odd-numbered states, the total number of words N_{σ_j} in state $\sigma_j \in \{\sigma_1, \sigma_3, \ldots, \sigma_{N-1}\}$ or $\sigma_j \in \{\sigma_2, \sigma_4, \ldots, \sigma_N\}$ after all extensions is

$$N_{\sigma_j} = 3(\frac{N}{2}) - 2, \tag{18}$$

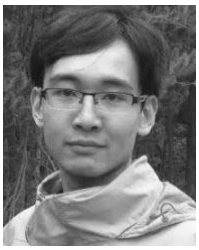
so there is the same number of words in all principal states of the extended minimal set and $\widetilde{\eta} = 100\%$.

iv) We now prove the words in the extended minimal set are prefix-free such that they are instantaneously decodable. First we observe that in the minimal set $W(\Psi)$, no word is a prefix of another. Therefore, the prefix problem could only have occurred if a word $W_q \in W(\sigma_u), W(\sigma_v)$ is extended in σ_u, but is not extended in σ_v. During the extensions described in this proof, only the all-one word or all-zero word is extended. Therefore, if the all-one word $W_q \in \{W(\sigma_u), W(\sigma_v)\}$, it is extended in σ_u and it is also extended in σ_v, since extensions continue until the all-one word is no longer in the set. Similar analysis holds for the all-zero word. Hence, the prefix problem is avoided and codebooks constructed based on these balanced extended minimal sets are instantaneously decodable. ■

REFERENCES

[1] K. A. S. Immink, *Codes for Mass Data Storage Systems*, 2nd ed. Eindhoven, The Netherlands: Shannon Foundation, 2004.
[2] F. R. Kschischang and T. Lutz, "A constrained coding approach to error-free half-duplex relay networks," *IEEE Trans. Inf. Theory*, vol. 59, no. 10, pp. 6258–6260, Oct. 2013.
[3] C. Cao, D. Li, and I. Fair, "Deep learning-based decoding for constrained sequence codes," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Abu Dhabi, United Arab Emirates, Dec. 2018, pp. 1–7.
[4] C. Cao, D. Li, and I. Fair, "Deep learning-based decoding of constrained sequence codes," *IEEE J. Sel. Areas Commun.*, to be published.
[5] S. Liu and F. R. Kschischang, "A constrained-coding alternative to MPPM," *IEEE Trans. Commun.*, vol. 60, no. 4, pp. 1013–1019, Apr. 2012.
[6] K. A. S. Immink, K. Cai, and J. H. Weber, "Dynamic threshold detection based on pearson distance detection," *IEEE Trans. Commun.*, vol. 66, no. 7, pp. 2958–2965, Jul. 2018.
[7] K. Cai, K. A. S. Immink, M. Zhang, and R. Zhao, "On the design of spectrum shaping codes for high-density data storage," *IEEE Trans. Consum. Electron.*, vol. 63, no. 4, pp. 477–482, Nov. 2017.
[8] R. Motwani, "Hierarchical constrained coding for floating-gate to floating-gate coupling mitigation in flash memory," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Houston, TX, USA, Dec. 2011, pp. 1–5.
[9] K. A. S. Immink and V. Skachek, "Minimum Pearson distance detection using mass-centered codewords in the presence of unknown varying offset," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 9, pp. 2510–2517, Sep. 2016.
[10] H. Zhou, A. Jiang, and J. Bruck. (2012). "Balanced modulation for non-volatile memories." [Online]. Available: https://arxiv.org/abs/1209.0744
[11] T. G. Swart, J. H. Weber, and K. A. S. Immink, "Prefixless q-ary balanced codes with fast syndrome-based error correction," *IEEE Trans. Inf. Theory*, vol. 64, no. 4, pp. 2431–2443, Apr. 2018.
[12] K. A. S. Immink and K. Cai, "Design of capacity-approaching constrained codes for DNA-based storage systems," *IEEE Commun. Lett.*, vol. 22, no. 2, pp. 224–227, Feb. 2018.
[13] *IEEE Standard for Local and Metropolitan Area Networks—Part 15.7: Short-Range Wireless Optical Communication Using Visible Light*, IEEE Standard 802.15.7, 2011, pp. 248–271.
[14] H. Wang and S. Kim, "New RLL decoding algorithm for multiple candidates in visible light communication," *IEEE Photon. Technol. Lett.*, vol. 27, no. 1, pp. 15–17, Jan. 1, 2015.
[15] H. Wang and S. Kim, "Soft-input soft-output run-length limited decoding for visible light communication," *IEEE Photon. Technol. Lett.*, vol. 28, no. 3, pp. 225–228, Feb. 1, 2016.

[16] H. Wang and S. Kim, "Bit-level soft run-length limited decoding algorithm for visible light communication," *IEEE Photon. Technol. Lett.*, vol. 28, no. 3, pp. 237–240, Feb. 1, 2016.
[17] C. E. Mejia, C. N. Georghiades, M. M. Abdallah, and Y. H. Al-Badarneh, "Code design for flicker mitigation in visible light communications using finite state machines," *IEEE Trans. Commun.*, vol. 65, no. 5, pp. 2091–2100, May 2017.
[18] C. E. Mejia, C. N. Georghiades, and Y. H. Al-Badarneh, "Code design in visible light communications using color-shift-keying constellations," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Washington, DC, USA, Dec. 2016, pp. 1–7.
[19] P. A. Franaszek, "Sequence-state encoding for digital transmission," *Bell Syst. Tech. J.*, vol. 47, pp. 143–157, Jan. 1968.
[20] M.-P. Béal, "The method of poles: A coding method for constrained channels," *IEEE Trans. Inf. Theory*, vol. 36, no. 4, pp. 763–772, Jul. 1990.
[21] C. D. Heegard, B. H. Marcus, and P. H. Siegel, "Variable-length state splitting with applications to average runlength-constrained (ARC) codes," *IEEE Trans. Inf. Theory*, vol. 37, no. 3, pp. 759–777, May 1991.
[22] J. H. Weber, T. G. Swart, and K. A. S. Immink, "Simple systematic Pearson coding," in *Proc. IEEE Int. Symp. Inf. Theory*, Barcelona, Spain, Jul. 2016, pp. 385–389.
[23] K. A. S. Immink and J. H. Weber, "Very efficient balanced codes," *IEEE J. Sel. Areas Commun.*, vol. 28, no. 2, pp. 188–192, Feb. 2010.
[24] A. Steadman and I. Fair, "Variable-length constrained sequence codes," *IEEE Commun. Lett.*, vol. 17, no. 1, pp. 139–142, Jan. 2013.
[25] A. Steadman and I. Fair, "Simplified search and construction of capacity-approaching variable-length constrained sequence codes," *IET Commun.*, vol. 10, no. 14, pp. 1697–1704, 2016.
[26] C. Cao and I. Fair, "Construction of minimal sets for capacity-approaching variable-length constrained sequence codes," in *Proc. Asilomar Conf. Signals, Syst., Comput.*, Monterey, CA, USA, Nov. 2016, pp. 255–259.
[27] C. Cao and I. Fair, "Minimal sets for capacity-approaching variable-length constrained sequence codes," *IEEE Trans. Commun.*, vol. 67, no. 2, pp. 890–902, Feb. 2019.
[28] C. Cao and I. Fair, "Mitigation of inter-cell interference in flash memory with capacity-approaching variable-length constrained sequence codes," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 9, pp. 2366–2377, Sep. 2016.
[29] C. Cao and I. Fair, "Capacity-approaching variable-length Pearson codes," *IEEE Commun. Lett.*, vol. 22, no. 7, pp. 1310–1313, Jul. 2018.
[30] K. A. S. Immink, J.-Y. Kim, S.-W. Suh, and S. K. Ahn, "Efficient DC-free RLL codes for optical recording," *IEEE Trans. Commun.*, vol. 51, no. 3, pp. 326–331, Mar. 2003.
[31] C. Jamieson and I. J. Fair, "Construction of constrained codes for state-independent decoding," *IEEE J. Sel. Areas Commun.*, vol. 28, no. 2, pp. 193–199, Feb. 2010.
[32] K. A. S. Immink, "Some statistical properties of maxentropic runlength-limited sequences," *Philips J. Res.*, vol. 38, no. 3, pp. 138–149, 1983.
[33] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, 1948.
[34] G. Böcherer, "Capacity-achieving probabilistic shaping for noisy and noiseless channels," Ph.D. dissertation, RWTH Aachen Univ., Aachen, Germany, 2012. [Online]. Available: http://www.georg-boecherer.de/capacityAchievingShaping.pdf
[35] G. Böcherer. (Dec. 2010). *Geometric Huffman Coding*. [Online]. Available: http://www.georg-boecherer.de/ghc
[36] A. X. Widmer and P. A. Franaszek, "A DC-balanced, partitioned-block, 8 B/10 B transmission code," *IBM J. Res. Develop.*, vol. 27, no. 5, pp. 440–451, Sep. 1983.
[37] I. J. Fair, Y. Zhu, and A. P. Hughes, "Spectra of multimode coded signals," *IEE Proc.-Commun.*, vol. 153, no. 3, pp. 383–391, Jun. 2006.



CONGZHE CAO received the B.Eng. degree in communications engineering from the University of Science and Technology Beijing, Beijing, China, in 2012, and the M.Eng. degree in communications engineering from the Beijing Institute of Technology, Beijing, in 2014. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Alberta, Canada. From 2016 to 2017, he was a Research Intern with Mitsubishi Electric Research Laboratories, Cambridge, MA, USA, where he was involved in designing advanced coding techniques for the next-generation wireless and fiber optic communication systems. His research interests include constrained sequence coding and error control coding for communication and emerging data storage systems, machine learning, deep learning, and information theory.

IVAN FAIR received the B.Sc. and M.Sc. degrees from the University of Alberta, in 1985 and 1989, respectively, and the Ph.D. degree from the University of Victoria, in 1995, all in electrical and computer engineering. He was with Bell Northern Research, Ltd., from 1985 to 1987, and with MPR TelTech, Ltd., from 1989 to 1991, where he was involved in various aspects of communication system design and implementation. In 1995, he joined the Technical University of Nova Scotia (since amalgamated with Dalhousie University) as an Assistant Professor and was promoted to Associate Professor. In 1998, he joined the University of Alberta, where he is currently a Professor. His research interest includes the coding for reliable digital communications.