Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Code design for run-length control in visible light communication*

Zong-yan LI†‡[1], Hong-lu YU†[1], Bao-ling SHAN†[2], De-xuan ZOU[3], Shi-yin LI[1]

[1]*School of Information and Control Engineering, China University of Mining and Technology, Xuzhou 221116, China*

[2]*School of Electrical and Data Engineering, University of Technology Sydney, Sydney 2007, Australia*

[3]*School of Electrical Engineering and Automation, Jiangsu Normal University, Xuzhou 221116, China*

†E-mail: lizongyan@cumt.edu.cn; yuhonglu1993@126.com; baoling.shan@student.uts.edu.au

Received Sept. 26, 2019; Revision accepted Mar. 30, 2020; Crosschecked Aug. 7, 2020

Abstract: Run-length limited (RLL) codes can facilitate reliable data transmission and provide flicker-free illumination in visible light communication (VLC) systems. We propose novel high-rate RLL codes, which can improve error performance and mitigate flicker. Two RLL coding schemes are developed by designing the finite-state machine to further enhance the coding gain by improving the minimum Hamming distance and using the state-splitting method to realize small state numbers. In our RLL code design, the construction of the codeword set is critical. This codeword set is designed considering the set-partitioning algorithm criterion. The flicker control and minimum Hamming distance of the various proposed RLL codes are described in detail, and the flicker performances of different codes are compared based on histograms. Simulations are conducted to evaluate the proposed RLL codes in on-off keying modulation VLC systems. Simulation results demonstrate that the proposed RLL codes achieve superior error performance to the existing RLL codes.

Key words: Visible light communication; Run-length limited codes; Finite-state machine; Minimum Hamming distance

https://doi.org/10.1631/FITEE.1900526 **CLC number:** TN911.22

1 Introduction

Visible light communication (VLC), in which the visible light spectrum (380–780 nm) is used to support the integration of illumination and communication, has emerged as a technique with significant potential in short-range indoor wireless communication systems (Rajagopal et al., 2012). Because the signal is transmitted by light-emitting diodes (LEDs) over optical channels and the received signal is detected by photo diodes (PDs) that detect the information, the main challenges of VLC are flicker mitigation and dimming control. To overcome these challenges, many researchers have developed improved approaches for run-length limited (RLL) and forward error correction (FEC) codes. In these approaches, the emphasis is on achieving reliable data transmission while preventing flicker and providing good dimming control (IEEE, 2011; Lu and Li, 2016; Babar et al., 2018).

In FEC coding, several coding schemes have been proposed to improve the transmission performance of VLC systems, including the Reed-Muller codes (Kim and Jung, 2011, 2013), rate-compatible convolutional codes (Kim and Park, 2014), low-density parity-check codes (Kim, 2015), turbo codes (Lee and Kwon, 2012), polar codes (Fang et al., 2017; Wang and Kim, 2018), and constrained sequence codes (Cao et al., 2019).

‡ Corresponding author

* Project supported by the Fundamental Research Funds for the Central Universities, China (No. 2020QN15)

ORCID: Zong-yan LI, https://orcid.org/0000-0001-8089-5921

In RLL coding schemes, such as the Manchester codes, FM0/FM1, 4B/6B, 8B/10B, and Miller codes, long runs of 1 s and 0 s are eliminated, which may cause LED flicker and synchronization clock detection problems (IEEE, 2011; Lu and Li, 2016). These codes mitigate flicker by maintaining a constant average illumination brightness and attenuating low-frequency components of the visible light signal. RLL constraints limit the number of encoded bits between consecutive transitions, and are equivalently referred to as the (k_1, k_2) constraints, where k_1 and k_2 denote the minimum and maximum numbers of 0 s (or 1 s) between consecutive 1 s (or 0 s) in a binary sequence (Immink, 2004), respectively. However, research on the error performance of RLL codes in VLC systems is limited.

To improve the 8B/10B code bit-error rate (BER) performance, the cyclic redundancy check decoding technique uses hard-decision decoding (Widner and Franaszek, 1983). Wang and Kim (2016) proposed a soft-input soft-output decoding strategy for 4B/6B codes, and demonstrated the effectiveness in a soft-decoded VLC system. Lu and Li (2018) proposed enhanced Miller (eMiller) codes, studied the eMiller coding scheme, and analyzed the BER and LED flicker problems. However, this scheme develops only a class of code-rate 1/2 RLL codes with one-bit input and two-bit output. Mejia et al. (2017) proposed RLL codes using a finite-state machine (FSM) to obtain high coding gains; however, the presented coding methods consider the number of codeword bits in the output to be even, and the number of states is high. Cao and Fair (2019a) discussed single-state and multi-state line codes based on FSM for VLC. Furthermore, multi-state coding with state-independent decoding of constrained sequence codes was proposed by Cao and Fair (2019b).

In this study, inspired by the run-length constraints of the RLL codes, we design a new class of $(0, k)$ RLL coding schemes using an FSM to further improve the BER performance. We propose RLL coding methods and two algorithms that can mitigate flicker and provide high coding gains.

Our main contributions can be summarized as follows:

1. We propose a new high-rate $(n-1)/n$ RLL coding scheme ($n > 2$) to achieve flicker mitigation. Each RLL code, based on the trellis structure and considering run-length constraints, is realized using the principle of FSM. The RLL codes put constraints on the maximum long runs of 1 s and 0 s in a coded sequence to mitigate flicker. The trellis structure can depict the coding process in detail. The computed minimum Hamming distances of the proposed RLL codes demonstrate better BER performance compared with other reported codes.

2. A set-partitioning mapping algorithm is proposed for n-bit binary codeword sets. Furthermore, an RLL coding scheme with a large minimum Hamming distance and small state numbers is proposed for $(n-1)/n$ RLL codes by introducing the state-splitting method and the set-partitioning mapping algorithm. We analyze the flicker control of the optimal RLL codes and demonstrate that flicker suppression can be achieved for VLC applications.

3. We propose a high-coding-gain RLL coding scheme by reducing the number of input information bits or increasing the number of codeword bits in the output. We design two high-coding-gain algorithms for code-rate $(n-2)/n$ RLL codes with different numbers of states. The first algorithm designs RLL codes for small state numbers in the FSM, whereas the second one designs RLL codes by reducing the number of states. The flicker control of the higher-coding-gain RLL codes, which can mitigate flicker, is analyzed. Simulation results further verify the superior BER performance.

2 RLL code design using finite-state machines

In this section, we describe the principles based on which RLL codes using an FSM are designed. We then propose a high-rate $(n-1)/n$ RLL coding scheme, which achieves flicker mitigation and better BER performance for $n > 2$.

2.1 Finite-state machines

Fig. 1 shows the FSM of a direct-current (DC) balanced code with N states, which includes states, edges, and labels. The running digital sum (RDS) of the DC-free encoded sequence is limited (Immink, 2004), where the RDS is the accumulation of the encoded bit weights in a sequence (logic one is represented by weight +1, and logic zero is represented by weight −1). From Fig. 1, the RDS can take any one of the N possible values.

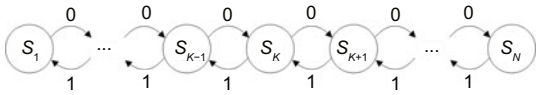

Fig. 1 General finite-state machine of DC-balanced codes

In Immink (2004), with the increase in the number of states, the number of encoded output sequences generated from the general FSM (Fig. 1) also increases. The number of low-frequency components of the power spectral density increases, thus increasing the flicker. The increase in the number of encoded sequences implies a larger minimum Hamming distance for the encoded output sequences. Therefore, there is a trade-off between flicker control and coding gain.

2.2 High-rate RLL code design with two states

For the high-rate $(n-1)/n$ RLL codes, we design an FSM with two states, following the trellis structure of rate $(n-1)/n$ RLL codes (Li et al., 2020). $\mathcal{C}_{S_{i,j}}$ denotes the codeword set of all possible codewords from the current state S_i to the next state S_j, where S_i (S_j) is for the i^{th} (j^{th}) state. Consider an n-bit codeword set, $\mathcal{C}=\{\underbrace{00\ldots0}_{n},\underbrace{00\ldots1}_{n},\ldots,\underbrace{11\ldots1}_{n}\}$, which contains 2^n binary numbers; $c_i\in\mathcal{C}$, where $i\in\{0,1,\ldots,2^n-1\}$.

A possible FSM for high-rate $(n-1)/n$ RLL codes is designed in Fig. 2. The relationship between the code rate R_{c} and codeword sets $\mathcal{C}_{S_{1,2}}$ and $\mathcal{C}_{S_{2,1}}$ is depicted in Table 1. Consider the FSM of rate 2/3 RLL codes as an example: $\mathcal{C}_{S_{1,2}}=\{c_1,c_2,c_4,c_7\}$ and $\mathcal{C}_{S_{2,1}}=\{c_0,c_3,c_5,c_6\}$. It is easy to determine that the minimum Hamming distance, $d_{\min}$, of these high-rate RLL codes is two, which is also listed in Table 1. Note that the maximum run-length (MRL) value, k, of these RLL codes does not exceed $3n-2$.

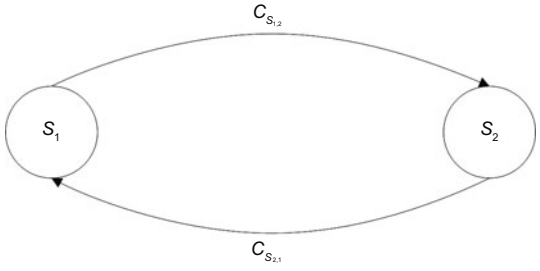

Fig. 2 FSM of rate $(n-1)/n$ RLL codes with two states

Table 1 Relationship between the code rate R_{c} and codeword sets $\mathcal{C}_{S_{1,2}}$ and $\mathcal{C}_{S_{2,1}}$ of high-rate $(n-1)/n$ RLL codes based on an FSM with two states

R_{c}	Codeword set	$d_{\min}$
2/3	$\mathcal{C}_{S_{1,2}}=\{c_1,c_2,c_4,c_7\}$, $\mathcal{C}_{S_{2,1}}=\{c_0,c_3,c_5,c_6\}$	2
3/4	$\mathcal{C}_{S_{1,2}}=\{c_1,c_2,c_4,c_7,c_8,c_{11},c_{13},c_{14}\}$, $\mathcal{C}_{S_{2,1}}=\{c_0,c_3,c_5,c_6,c_9,c_{10},c_{12},c_{15}\}$	2
4/5	$\mathcal{C}_{S_{1,2}}=\{c_1,c_2,c_4,c_7,c_8,c_{11},c_{13},c_{14},$ $c_{16},c_{19},c_{21},c_{22},c_{25},c_{26},c_{28},c_{31}\}$, $\mathcal{C}_{S_{2,1}}=\{c_0,c_3,c_5,c_6,c_9,c_{10},c_{12},c_{15},$ $c_{17},c_{18},c_{20},c_{23},c_{24},c_{27},c_{29},c_{30}\}$	2
5/6	$\mathcal{C}_{S_{1,2}}=\{c_1,c_2,c_4,c_7,c_8,c_{11},c_{13},c_{14},$ $c_{16},c_{19},c_{21},c_{22},c_{25},c_{26},c_{28},c_{31},$ $c_{32},c_{35},c_{37},c_{38},c_{41},c_{42},c_{44},c_{47},$ $c_{49},c_{50},c_{52},c_{55},c_{56},c_{59},c_{61},c_{62}\}$, $\mathcal{C}_{S_{2,1}}=\{c_0,c_3,c_5,c_6,c_9,c_{10},c_{12},c_{15},$ $c_{17},c_{18},c_{20},c_{23},c_{24},c_{27},c_{29},c_{30},$ $c_{33},c_{34},c_{36},c_{39},c_{40},c_{43},c_{45},c_{46},$ $c_{48},c_{51},c_{53},c_{54},c_{57},c_{58},c_{60},c_{63}\}$	2
⋮	⋮	⋮

The mapping relationship between the codeword sets and input information bits of rate 2/3 RLL codes is presented in Fig. 3. The same mapping relationship can be used for other high-rate $(n-1)/n$ RLL codes.

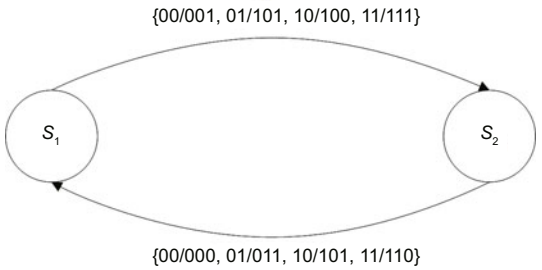

Fig. 3 Mapping relationship between the input information bits and codeword sets of rate 2/3 RLL codes

3 RLL code design with large $d_{\min}$

In this section, we propose two coding schemes for rate $(n-1)/n$ and $(n-2)/n$ RLL codes, respectively.

3.1 High-rate codes with $d_{\min}=3$ and small state numbers

For an appropriate code rate, $(n-1)/n$, the main process in the coding scheme involves the design of the set-partitioning algorithm and the FSM of $(0,k)$

RLL codes for a high coding gain and small state numbers.

3.1.1 Set-partitioning algorithm

To introduce our design, the following codeword-set definitions are required: Consider an n-bit codeword set, $\mathcal{C}$, which can be partitioned into l levels ($l \in \{0,1,2\}$), and $|\mathcal{C}| = 2^n$. Then, at partitioning level $l = 1$, we obtain a subset $\mathcal{C}(\alpha_0)$, which can be expressed as follows:

$$\mathcal{C}(\alpha_0) = \{c_i, c_i \in \mathcal{C}\},\ \alpha_0 \in \{0,1\}. \tag{1}$$

At partitioning level $l = 2$, we obtain a subset $\mathcal{C}(\alpha_0, \alpha_1)$, which can be expressed as follows:

$$\mathcal{C}(\alpha_0, \alpha_1) = \{c_i, c_i \in \mathcal{C}(\alpha_0)\} \tag{2}$$

and $\forall \mathcal{C}(\alpha_0)$, when $|\mathcal{C}(\alpha_0)| > 4$, $\alpha_1 \in \{0,1,\ldots,4p-1\}$; when $|\mathcal{C}(\alpha_0)| = 4$, $\alpha_1 \in \{0,1,2,3\}$. Set $q = \log_2(|\mathcal{C}(\alpha_0)|/4)$. Then we have

$$p = \begin{cases} q, & q \text{ is even or equal to 1,} \\ q-1, & \text{otherwise.} \end{cases} \tag{3}$$

A short summary of the set-partitioning process is depicted in Algorithm 1.

Fig. 4 shows the set-partitioning process of a 3-bit codeword set $\mathcal{C}$ based on Algorithm 1. At level $l = 1$, we have $\mathcal{C}(0) = \{c_1, c_2, c_4, c_7\}$ and $\mathcal{C}(1) = \{c_0, c_3, s_5, c_6\}$. At level $l = 2$, we have $\mathcal{C}(0,0) = \{c_1\}$, $\mathcal{C}(0,1) = \{c_2\}$, $\mathcal{C}(0,2) = \{c_4\}$, $\mathcal{C}(0,3) = \{c_7\}$, $\mathcal{C}(1,0) = \{c_0\}$, $\mathcal{C}(1,1) = \{c_3\}$, $\mathcal{C}(1,2) = \{c_5\}$, and $\mathcal{C}(1,3) = \{c_6\}$.

Algorithm 1 Set-partitioning of an n-bit codeword set $\mathcal{C}$

Initialization: select $R_c = (n-1)/n$, $n > 2$
for partitioning level l from 1 to 2 **do**
  if $l = 1$ **then**
    Obtain $\mathcal{C}(\alpha_0) = \{c_i, c_i \in \mathcal{C}\}, \alpha_0 \in \{0,1\}$
    if $\log_2(|\mathcal{C}(\alpha_0)|/4)$ is even or equal to 1 **then**
      $p = \log_2(|\mathcal{C}(\alpha_0)|/4)$
    else
      $p = \log_2(|\mathcal{C}(\alpha_0)|/4) - 1$
    end if
  else
    if $|\mathcal{C}(\alpha_0)| = 4$ **then**
      Obtain $\mathcal{C}(\alpha_0, \alpha_1) = \{c_i, c_i \in \mathcal{C}(\alpha_0)\}$, $\alpha_1 \in \{0,1,2,3\}$
    end if
    if $|\mathcal{C}(\alpha_0)| > 4$ **then**
      Obtain $\mathcal{C}(\alpha_0, \alpha_1) = \{c_i, c_i \in \mathcal{C}(\alpha_0)\}$, $\alpha_1 \in \{0,1,\ldots,4p-1\}$
    end if
  end if
end for

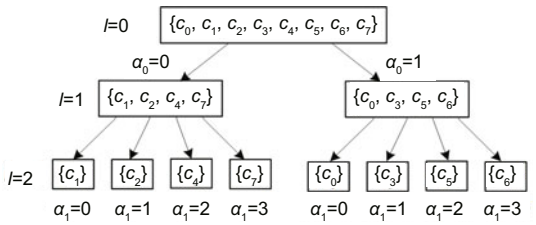

Fig. 4 Set-partitioning process of a 3-bit codeword set

Fig. 5 shows the set-partitioning process of a 4-bit codeword set $\mathcal{C}$ based on Algorithm 1. At level $l = 1$, we have $\mathcal{C}(0) = \{c_1, c_2, c_4, c_7, c_8, c_{11}, c_{13}, c_{14}\}$ and $\mathcal{C}(1) = \{c_0, c_3, s_5, c_6, c_9, c_{10}, c_{12}, c_{15}\}$. At level $l = 2$, we have $\mathcal{C}(0,0) = \{c_1, c_{14}\}$, $\mathcal{C}(0,1) = \{c_2, c_{13}\}$, $\mathcal{C}(0,2) = \{c_4, c_{11}\}$, $\mathcal{C}(0,3) = \{c_7, c_8\}$, $\mathcal{C}(1,0) = \{c_0, c_{15}\}$, $\mathcal{C}(1,1) = \{c_3, c_{12}\}$, $\mathcal{C}(1,2) = \{c_5, c_{10}\}$, and $\mathcal{C}(1,3) = \{c_6, c_9\}$.

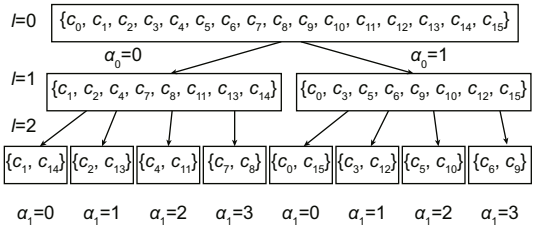

Fig. 5 Set-partitioning process of a 4-bit codeword set

Let $d_{\min}^{1,\alpha_0}$ denote the minimum intra-set distance of subset $\mathcal{C}(\alpha_0)$ at partitioning level 1, and $d_{\min}^{2,(\alpha_0,\alpha_1)}$ denote that of subset $\mathcal{C}(\alpha_0, \alpha_1)$ at partitioning level 2. They can be defined as

$$d_{\min}^{1,\alpha_0} = \min_{i \neq i'} \{d(c_i, c_{i'}),\ \forall (c_i, c_{i'}) \in \mathcal{C}(\alpha_0) \text{ and } \forall \mathcal{C}(\alpha_0)\}, \tag{4}$$

$$d_{\min}^{2,(0,\alpha_1)} = \min_{i \neq i'} \{d(c_i, c_{i'}),\ \forall (c_i, c_{i'}) \in \mathcal{C}(0, \alpha_1) \text{ and } \forall \mathcal{C}(0, \alpha_1)\}, \tag{5}$$

$$d_{\min}^{2,(1,\alpha_1)} = \min_{i \neq i'} \{d(c_i, c_{i'}),\ \forall (c_i, c_{i'}) \in \mathcal{C}(1, \alpha_1) \text{ and } \forall \mathcal{C}(1, \alpha_1)\}, \tag{6}$$

where $d(c_i, c_{i'}) = \sum_{t=0}^{|c_i|-1} (c_{i',t} \oplus c_{i,t})$ ($\oplus$ denotes the modulo 2 plus).

Furthermore, the minimum intra-set distances at the partitioning levels of $l = 1, 2$ are defined as

$$d^1_{\min} = \min\{d^{1,0}_{\min}, d^{1,1}_{\min}\}, \tag{7}$$

$$d^2_{\min} = \min\{d^{2,(0,\alpha_1)}_{\min}, d^{2,(1,\alpha_1)}_{\min}\}. \tag{8}$$

Note that when $l = 0$, $d^0_{\min}$ is the minimum distance of the codeword set $\forall\mathcal{C}$ and $d^0_{\min} = 1$.

Based on the set-partitioning methodology, it is desirable to obtain $d^0_{\min} < d^1_{\min} < d^2_{\min}$ such that the BER performance can be improved because of a large minimum intra-set distance. This is because the asymptotic error probability at high signal-to-noise ratio (SNR) under an additive white Gaussian noise (AWGN) channel can be approximated by (Simon and Divsalar, 2006)

$$P_{\mathrm{e}} \approx Q\left(\sqrt{\frac{2R_{\mathrm{c}}d_{\min}E_{\mathrm{b}}}{N_0}}\right), \tag{9}$$

where $Q(x) = \frac{1}{\sqrt{2\pi}}\int_x^\infty \exp(-t^2/2)\mathrm{d}t$, E_{b}/N_0 the ratio of signal bit energy to noise spectral density, R_{c} the code rate, and $N_0/2 = \sigma^2$ the variance of the AWGN.

3.1.2 FSM design for rate $(n-1)/n$ RLL codes with $d_{\min} = 3$ and small state numbers

The proposed FSM design for the RLL codes involves two stages in Algorithm 2. The codeword set is first designed, resulting in several codeword sets that satisfy the criterion $d^0_{\min} < d^1_{\min} < d^2_{\min}$. The state splitting technique based on Table 1 is then applied to design the RLL-code FSM to obtain a large minimum distance and small state numbers.

Stage 1: design the codeword sets such that $d^0_{\min} < d^1_{\min} < d^2_{\min}$.

Figs. 4 and 5 display the designs of the 3- and 4-bit codeword sets at 0, 1, and 2 partitioning levels, respectively. Applying the set-partitioning criterion is intended to ensure a monotonically increasing minimum intra-set distance, such as $d^0_{\min} < d^1_{\min} < d^2_{\min}$. As previously mentioned, $d^0_{\min} = 1$. We establish that $d^0_{\min} < d^1_{\min} < d^2_{\min}$ is strictly ensured. Thus, the codeword sets can be partitioned as shown in Table 1. Then, at partitioning level 1, $d^1_{\min} = 2$ can be obtained. At partitioning level 2, we design $d^2_{\min} = 4$. In Table 2, the designed various codeword sets of high-rate RLL codes at partitioning level 2 with $d^2_{\min} = 4$ are listed.

Table 2 Various codeword sets of the high-rate RLL codes at partitioning level 2

R_{c}	$\mathcal{C}(0,\alpha_1)$ and $\mathcal{C}(1,\alpha_1)$	$d^2_{\min}$
2/3	$\mathcal{C}(0,0)=\{c_1\},\mathcal{C}(0,1)=\{c_2\}$, $\mathcal{C}(0,2)=\{c_4\},\mathcal{C}(0,3)=\{c_7\}$, $\mathcal{C}(1,0)=\{c_0\},\mathcal{C}(1,1)=\{c_3\}$, $\mathcal{C}(1,2)=\{c_5\},\mathcal{C}(1,3)=\{c_6\}$	–
3/4	$\mathcal{C}(0,0)=\{c_1,c_{14}\},\mathcal{C}(0,1)=\{c_2,c_{13}\}$, $\mathcal{C}(0,2)=\{c_4,c_{11}\},\mathcal{C}(0,3)=\{c_7,c_8\}$, $\mathcal{C}(1,0)=\{c_0,c_{15}\},\mathcal{C}(1,1)=\{c_3,c_{12}\}$, $\mathcal{C}(1,2)=\{c_5,c_{10}\},\mathcal{C}(1,3)=\{c_6,c_9\}$	4
4/5	$\mathcal{C}(0,0)=\{c_1,c_{14}\},\mathcal{C}(0,1)=\{c_2,c_{13}\}$, $\mathcal{C}(0,2)=\{c_4,c_{11}\},\mathcal{C}(0,3)=\{c_7,c_8\}$, $\mathcal{C}(0,4)=\{c_{16},c_{31}\},\mathcal{C}(0,5)=\{c_{19},c_{28}\}$, $\mathcal{C}(0,6)=\{c_{21},c_{26}\},\mathcal{C}(0,7)=\{c_{22},c_{25}\}$, $\mathcal{C}(1,0)=\{c_0,c_{15}\},\mathcal{C}(1,1)=\{c_3,c_{12}\}$, $\mathcal{C}(1,2)=\{c_5,c_{10}\},\mathcal{C}(1,3)=\{c_6,c_9\}$, $\mathcal{C}(1,4)=\{c_{17},c_{30}\},\mathcal{C}(1,5)=\{c_{18},c_{29}\}$, $\mathcal{C}(1,6)=\{c_{20},c_{27}\},\mathcal{C}(1,7)=\{c_{23},c_{24}\}$	4
5/6	$\mathcal{C}(0,0)=\{c_1,c_{14},c_{50},c_{61}\}$, $\mathcal{C}(0,1)=\{c_2,c_{13},c_{52},c_{59}\}$, $\mathcal{C}(0,2)=\{c_4,c_{11},c_{55},c_{56}\}$, $\mathcal{C}(0,3)=\{c_7,c_8,c_{49},c_{62}\}$, $\mathcal{C}(0,4)=\{c_{16},c_{31},c_{32},c_{47}\}$, $\mathcal{C}(0,5)=\{c_{19},c_{28},c_{35},c_{44}\}$, $\mathcal{C}(0,6)=\{c_{21},c_{26},c_{37},c_{42}\}$, $\mathcal{C}(0,7)=\{c_{22},c_{25},c_{38},c_{41}\}$, $\mathcal{C}(1,0)=\{c_0,c_{15},c_{51},c_{60}\}$, $\mathcal{C}(1,1)=\{c_3,c_{12},c_{48},c_{63}\}$, $\mathcal{C}(1,2)=\{c_5,c_{10},c_{54},c_{57}\}$, $\mathcal{C}(1,3)=\{c_6,c_9,c_{53},c_{58}\}$, $\mathcal{C}(1,4)=\{c_{17},c_{30},c_{34},c_{45}\}$, $\mathcal{C}(1,5)=\{c_{18},c_{29},c_{36},c_{43}\}$, $\mathcal{C}(1,6)=\{c_{20},c_{27},c_{39},c_{40}\}$, $\mathcal{C}(1,7)=\{c_{23},c_{24},c_{33},c_{46}\}$	4
⋮	⋮	⋮

Stage 2: based on the results of stage 1, design the FSM for $(0,k)$ RLL codes with $d_{\min} = 3$ and a small state number N.

Based on the results of stage 1 and Fig. 3, we design the FSM of $(0,k)$ RLL codes with $d_{\min} = 3$ and a small state number N. Select all the codeword sets $\mathcal{C}(\alpha_0,\alpha_1)$ at partitioning level 2 and set $m = (\max\{\alpha_1\}+1)/2$. Then, the state number $N = 2m$.

First, we split states S_1 and S_2 into m states, i.e., S_1^θ and S_2^θ, $\theta \in \{1, 2, \ldots, m\}$. We then distribute the codeword sets such that for all the possible sequences in the FSM, $d_{\min}$ is equal to 3. The algorithm begins by selecting states S_1 and S_2, and all the codeword sets are at level 2. Note that we need to avoid the same subset arriving at a common state in the FSM design process. It is crucial to ensure that the MRL of the encoded sequence does not exceed $4n-2$ to achieve flicker control.

Algorithm 2 Finite-state machine design

Initialization: select $R_c = (n-1)/n$, $n > 2$
Stage 1: design the codeword sets such that $d^0_{\min} < d^1_{\min} < d^2_{\min}$
for partitioning level l varying from 0 to 2 **do**
  if $l = 0$ **then**
    Obtain $d^0_{\min} = 1$
  end if
  if $l = 1$ **then**
    Based on Fig. 2 and Table 1, select codeword set $\mathcal{C}_{S_{1,2}}$ as $\mathcal{C}(0)$ and $\mathcal{C}_{S_{2,1}}$ as $\mathcal{C}(1)$
    Obtain $d^1_{\min} = 2$
  end if
  if $l = 2$ **then**
    if $|\mathcal{C}(\alpha_0)| = 4$ **then**
      Calculate $d^2_{\min}$
      Set $|\mathcal{C}(\alpha_0, \alpha_1)| = 1$, $\alpha_1 \in \{0, 1, 2, 3\}$
    end if
    if $|\mathcal{C}(\alpha_0)| > 4$ **then**
      Set $d^2_{\min} = 4$
    end if
  end if
end for
Stage 2: based on the results of stage 1, design the FSM of $(0, k)$ RLL codes with $d_{\min} = 3$ and small state number N
From stage 1, select all the codeword sets, $\mathcal{C}(\alpha_0, \alpha_1)$, at partitioning level 2, and set $m = (\max\{\alpha_1\} + 1)/2$
From Fig. 2, select two states S_1 and S_2
for a given m **do**
  Split S_1 and S_2 into m states, i.e., S_1^θ and S_2^θ ($\theta \in \{1, 2, \ldots, m\}$)
  loop
    Connect codeword subsets $\mathcal{C}(\alpha_0, \alpha_1)$
    Ensure that the MRL is $k \le 4n - 2$
    Ensure that no codeword set is connected twice to a state
    Obtain $d_{\min} = 3$ for all the FSM codeword sequences
  end loop
end for

Using Eq. (2), we can ensure that the rate 2/3 and 3/4 RLL codes have the same $\max\{\alpha_1\} = 3$. Furthermore, from stage 2, we know that $m = 2$. Thus, as shown in Fig. 6, both S_1 and S_2 must be split into two states. Fig. 6 displays the FSM of rate 2/3 RLL codes with $d_{\min} = 3$ and the state number $N = 4$. As shown in Fig. 6a, we first construct the FSM with states S_1^1 and S_1^2, which connect all the codeword sets of $\mathcal{C}(0, \alpha_1)$ and $\mathcal{C}(1, \alpha_1)$, respectively. Furthermore, the codeword sets between S_1^θ and S_2^θ are connected as depicted in Fig. 6b, where $\theta = 1, 2$.

Similarly, the design of the FSM for rate 3/4 RLL codes with $d_{\min} = 3$ and state number $N = 4$ is shown in Fig. 7.

According to the FSMs in Figs. 6 and 7, the possible mapping relationship between the codeword sets and input information bits is depicted in Table 3. Similarly, in Appendix A, the possible mapping codes for the rate 4/5 and rate 5/6 RLL codes are presented in Tables A1 and A2, respectively.

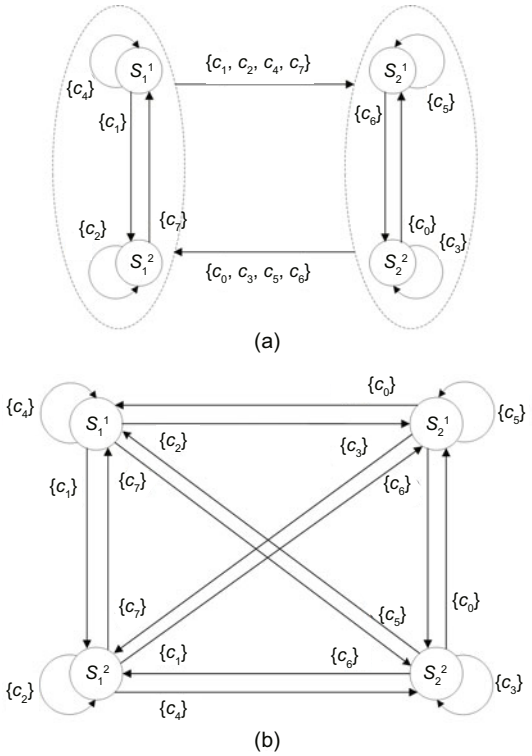

Fig. 6 FSM of rate 2/3 RLL codes with $d_{\min} = 3$ and $N = 4$: (a) splitting S_1 and S_2 into two states; (b) four states connected by the codeword subset with $d_{\min} = 3$

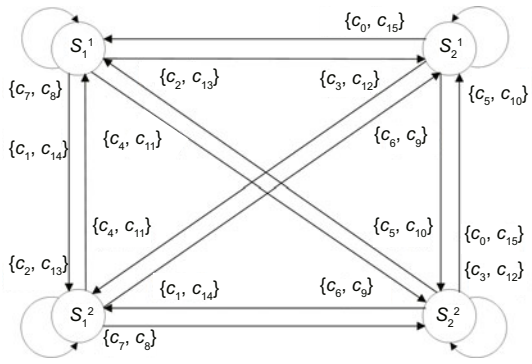

Fig. 7 FSM of rate 3/4 RLL codes with $d_{\min} = 3$ and $N = 4$

3.2 High-rate codes with $d_{\min} = 4$ and small state numbers

FSM design of RLL codes with $d_{\min} = 4$ is summarized in Algorithm 3. Algorithm 3 begins by selecting all the codeword sets $\mathcal{C}(\alpha_0, \alpha_1)$ at partitioning level 2 and calculates the total number of states N. Based on the FSM of RLL codes with $d_{\min} = 3$ and the coding process of stage 2, we present an example including two choices:

Choice 1: we retain the connection of codeword subset $\mathcal{C}(0, \alpha_1)$ or $\mathcal{C}(1, \alpha_1)$, and eliminate the connection of the codeword sets between $\mathcal{C}(0, \alpha_1)$ and $\mathcal{C}(1, \alpha_1)$.

Choice 2: we retain all the states S_1^θ and S_2^θ, $\theta = 1, 2, \ldots, m$. We then select all the codeword sets $\mathcal{C}(0, \alpha_1)$ or $\mathcal{C}(1, \alpha_1)$. Considering $\mathcal{C}(0, \alpha_1)$ as an example, we need to split each $\mathcal{C}(0, \alpha_1)$ into two codeword subsets $\mathcal{C}_1(0, \alpha_1)$ and $\mathcal{C}_2(0, \alpha_1)$; meanwhile, $|\mathcal{C}_1(0, \alpha_1)| = |\mathcal{C}_2(0, \alpha_1)|$ should be satisfied. Furthermore, we connect all the codeword sets $\mathcal{C}_1(0, \alpha_1)$ and $\mathcal{C}_2(0, \alpha_1)$ under the MRL constraint, and ensure that no codeword set is connected to a state twice. The same method can be used for all the codeword sets $\mathcal{C}(1, \alpha_1)$.

Fig. 8 shows the FSM of rate 2/4 RLL codes with $d_{\min} = 4$. For choice 1, Figs. 8a and 8b present two FSMs with state number $N = 2$. For choice 2, Fig. 8c presents the FSM with state number $N = 4$, considering $\mathcal{C}(0, \alpha_1)$ as an example.

Table 3 Trellis diagram and mapping relationship between the codeword sets and information bits of rate 2/3 and rate 3/4 codes

R_c	Current state	Next state	Output codeword set	Input bit	$d_{\min}$
2/3	S_1^1	S_1^1	$\mathcal{C}(0,2)=\{c_4\}$	00	3
		S_1^2	$\mathcal{C}(0,0)=\{c_1\}$	01	
		S_2^1	$\mathcal{C}(0,1)=\{c_2\}$	10	
		S_2^2	$\mathcal{C}(0,3)=\{c_7\}$	11	
	S_1^2	S_1^1	$\mathcal{C}(0,3)=\{c_7\}$	00	
		S_1^2	$\mathcal{C}(0,1)=\{c_2\}$	01	
		S_2^1	$\mathcal{C}(0,0)=\{c_1\}$	10	
		S_2^2	$\mathcal{C}(0,2)=\{c_4\}$	11	
	S_2^1	S_1^1	$\mathcal{C}(1,0)=\{c_0\}$	00	
		S_1^2	$\mathcal{C}(1,1)=\{c_3\}$	01	
		S_2^1	$\mathcal{C}(1,2)=\{c_5\}$	10	
		S_2^2	$\mathcal{C}(1,3)=\{c_6\}$	11	
	S_2^2	S_1^1	$\mathcal{C}(1,2)=\{c_5\}$	00	
		S_1^2	$\mathcal{C}(1,3)=\{c_6\}$	01	
		S_2^1	$\mathcal{C}(1,0)=\{c_0\}$	10	
		S_2^2	$\mathcal{C}(1,1)=\{c_3\}$	11	
3/4	S_1^1	S_1^1	$\mathcal{C}(0,3)=\{c_7,c_8\}$	000, 001	3
		S_1^2	$\mathcal{C}(0,0)=\{c_1,c_{14}\}$	010, 011	
		S_2^1	$\mathcal{C}(0,1)=\{c_2,c_{13}\}$	100, 101	
		S_2^2	$\mathcal{C}(0,2)=\{c_4,c_{11}\}$	110, 111	
	S_1^2	S_1^1	$\mathcal{C}(0,2)=\{c_4,c_{11}\}$	000, 001	
		S_1^2	$\mathcal{C}(0,1)=\{c_2,c_{13}\}$	010, 011	
		S_2^1	$\mathcal{C}(0,0)=\{c_1,c_{14}\}$	100, 101	
		S_2^2	$\mathcal{C}(0,3)=\{c_7,c_8\}$	110, 111	
	S_2^1	S_1^1	$\mathcal{C}(1,0)=\{c_0,c_{15}\}$	000, 001	
		S_1^2	$\mathcal{C}(1,1)=\{c_3,c_{12}\}$	010, 011	
		S_2^1	$\mathcal{C}(1,2)=\{c_5,c_{10}\}$	100, 101	
		S_2^2	$\mathcal{C}(1,4)=\{c_6,c_9\}$	110, 111	
	S_2^2	S_1^1	$\mathcal{C}(1,2)=\{c_5,c_{10}\}$	000, 001	
		S_1^2	$\mathcal{C}(1,3)=\{c_6,c_9\}$	010, 011	
		S_2^1	$\mathcal{C}(1,0)=\{c_0,c_{15}\}$	100, 101	
		S_2^2	$\mathcal{C}(1,1)=\{c_3,c_{12}\}$	110, 111	

4 Flicker analysis and simulation results

In this section, we present the schematic of the VLC system. Furthermore, we present the flicker analysis and simulation results to verify the effectiveness of the proposed RLL codes using the VLC system model depicted in Fig. 9.

Algorithm 3 FSM design for RLL codes with $d_{\min} = 4$

Initialization: select $R_c = (n-2)/n$, $n > 2$
From stage 1, select all the codeword sets $\mathcal{C}(\alpha_0, \alpha_1)$ at partitioning level 2, and set $m = (\max\{\alpha_1\}+1)/2$
Choice 1:
for $N = m$ **do**
    From stage 2, retain the connection of codeword subsets $\mathcal{C}(0, \alpha_1)$ or $\mathcal{C}(1, \alpha_1)$, and eliminate the connection of the codeword sets between $\mathcal{C}(0, \alpha_1)$ and $\mathcal{C}(1, \alpha_1)$
end for
Choice 2:
for $N = 2m$ **do**
    From stage 2, retain all the states S_1^θ and S_2^θ, $\theta = 1, 2, \ldots, m$
    Select all the codeword sets $\mathcal{C}(0, \alpha_1)$, and split each codeword set $\mathcal{C}(0, \alpha_1)$ into two codeword subsets $\mathcal{C}_1(0, \alpha_1)$ and $\mathcal{C}_2(0, \alpha_1)$, which must satisfy $|\mathcal{C}_1(0, \alpha_1)| = |\mathcal{C}_2(0, \alpha_1)|$
    loop
        Connect codeword sets $\mathcal{C}_1(0, \alpha_1)$ or $\mathcal{C}_2(0, \alpha_1)$
        Ensure that the MRL $k \le 4n - 2$
        Ensure that no codeword set is connected twice to a state
        Obtain $d_{\min} = 4$ for all the FSM codeword sequences
    end loop
end for

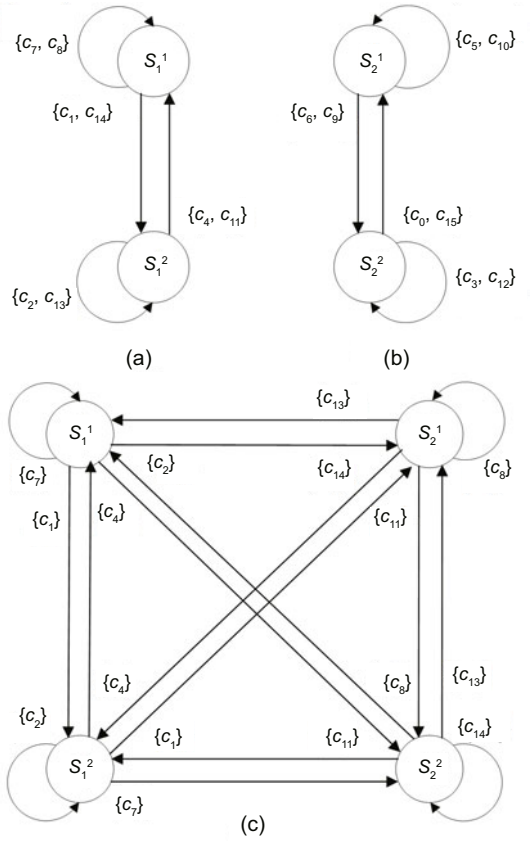

Fig. 8 FSM of rate 2/4 RLL codes with $d_{\min} = 4$ and $N = 2, 4$: (a) splitting S_1 into two states; (b) splitting S_2 into two states; (c) four states connected by the codeword subset

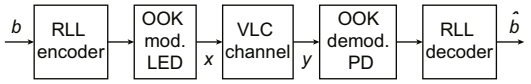

Fig. 9 Schematic of the VLC system

4.1 System model

The on-off keying (OOK) modulated VLC system shown in Fig. 9 relies on RLL codes to achieve error correction and flicker suppression. At the transmitter, the transmitted binary data sequence b is inputted to the RLL encoder to generate RLL-encoded bits, which are inputted to the OOK modulator through an LED. The OOK modulated output x travels through the VLC channel. The receiver is within the field-of-view (FOV) and in the line-of-sight (LOS) of the transmitting LED. Inter-symbol interference (ISI), dispersion, and light reflections are not considered in this study because the focus is on RLL code design. The PD received signal is expressed as $y = x + n$, where n represents the AWGN with zero mean and variance σ^2. The received signal is then processed by the OOK demodulator. The demodulated data is inputted to an RLL decoder, which uses the soft Viterbi algorithm. Our aim is to design RLL codes to assist the VLC receiver in achieving better BER performance, while constraining the length of consecutive 0 s and 1 s to enable the recovery of the synchronization clock and avoid LED flicker.

4.2 Flicker analysis

Because the VLC system performs the dual functions of illumination and communication, flicker mitigation and dimming control should be considered while providing data communication. When the flickering time period is within the maximum flickering time period (MFTP), which is approximately 5 ms (Berman et al., 1991), flicker is imperceptible to the human eye. The high-data-rate VLC system, considered as an example, satisfies the MFTP; i.e., if the data rate is Q kb/s, then the number of consecutive bits $M \leq 5Q$.

Flicker is caused by changes in the brightness between every two M bits. It is generally recognized that brightness change with a frequency higher than 1/MFTP is beyond human-eye perception. To evaluate the impact of these fluctuations, we compute the brightness level over M bits. For convenience, we introduce the concept of super symbol M, which is formed by M consecutive bits. Results are displayed in Figs. 10–23 (random bit generation), where $M = 800,\ 2000,\ 5000,\ 7500$, or 10 000; note that the sample symbol M meets the MFTP criterion. It can be observed that brightness fluctuation continues to exist, even when using multiple rate RLL codes for $M = 800$. As illustrated in Figs. 10–23, the flickering fluctuations of all the proposed RLL codes become fairly consistent (50%) when M is large enough, such as $M = 5000,\ 7500$, or 10 000.

From Figs. 10–13, although the difference in the performance among schemes is not significant, the flickering fluctuation of the high-rate RLL codes is slightly smaller than that of the low-rate ones for $d_{\min} = 2$. As shown in Figs. 14–17, the flickering fluctuation of high-rate RLL codes is also smaller than that of the low-rate ones for $d_{\min} = 3$.

From Figs. 18–23, the flickering fluctuation for small state numbers N is slightly smaller than that of the larger ones for the RLL codes at the same rate. The flickering fluctuation of the high-rate RLL

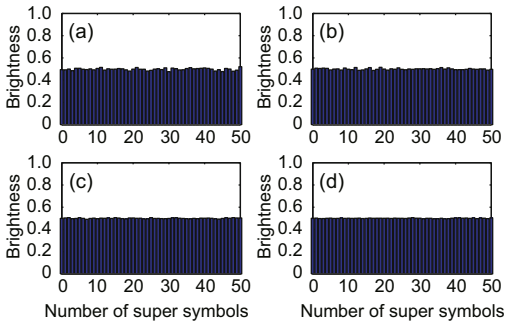

Fig. 10 Brightness in rate 2/3 RLL codes with $N = 2$ and $d_{\min} = 2$: (a) $M = 800$; (b) $M = 2000$; (c) $M = 7500$; (d) $M = 10\,000$

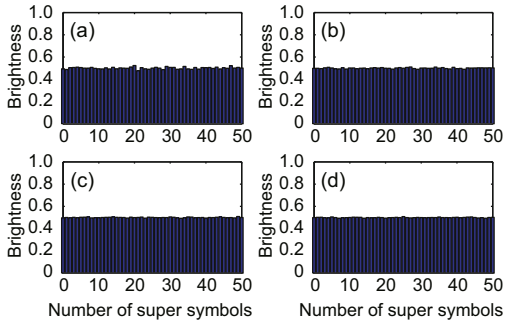

Fig. 11 Brightness in rate 3/4 RLL codes with $N = 2$ and $d_{\min} = 2$: (a) $M = 800$; (b) $M = 2000$; (c) $M = 5000$; (d) $M = 7500$

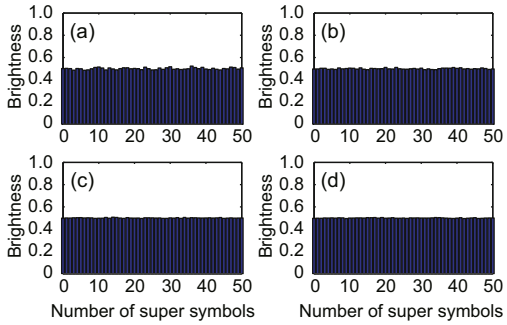

Fig. 12 Brightness in rate 4/5 RLL codes with $N = 2$ and $d_{\min} = 2$: (a) $M = 800$; (b) $M = 2000$; (c) $M = 5000$; (d) $M = 7500$

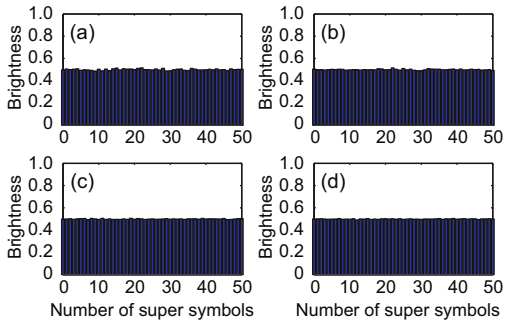

Fig. 13 Brightness in rate 5/6 RLL codes with $N = 2$ and $d_{\min} = 2$: (a) $M = 800$; (b) $M = 2000$; (c) $M = 5000$; (d) $M = 7500$

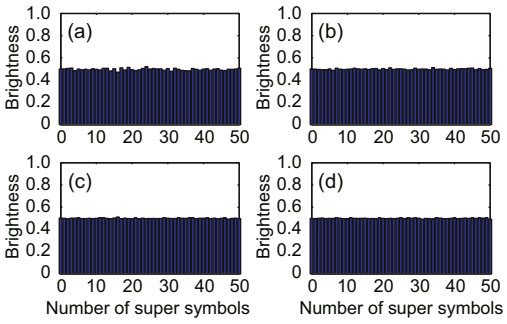

Fig. 14 Brightness in rate 2/3 RLL codes with $N = 4$ and $d_{\min} = 3$: (a) $M = 800$; (b) $M = 2000$; (c) $M = 7500$; (d) $M = 10\,000$

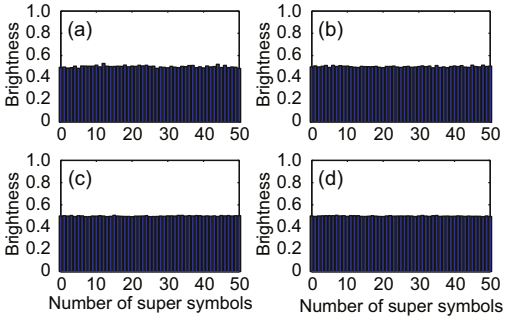

Fig. 15 Brightness in rate 3/4 RLL codes with $N = 4$ and $d_{\min} = 3$: (a) $M = 800$; (b) $M = 2000$; (c) $M = 5000$; (d) $M = 7500$

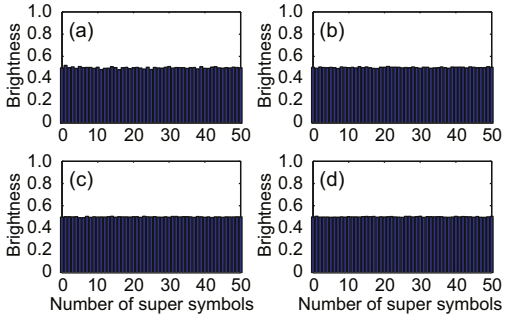

Fig. 16 Brightness in rate 4/5 RLL codes with $N = 8$ and $d_{\min} = 3$: (a) $M = 800$; (b) $M = 2000$; (c) $M = 5000$; (d) $M = 7500$

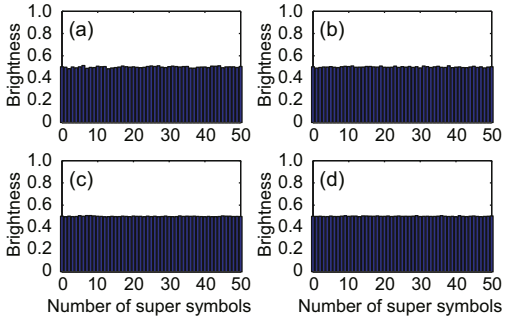

Fig. 17 Brightness in rate 5/6 RLL codes with $N = 8$ and $d_{\min} = 3$: (a) $M = 800$; (b) $M = 2000$; (c) $M = 5000$; (d) $M = 7500$

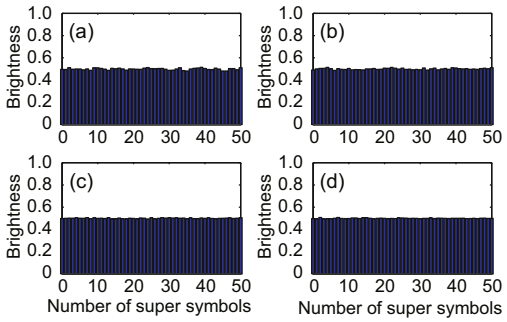

Fig. 18 Brightness in rate 2/4 RLL codes with $N = 2$ and $d_{\min} = 4$: (a) $M = 800$; (b) $M = 2000$; (c) $M = 5000$; (d) $M = 7500$

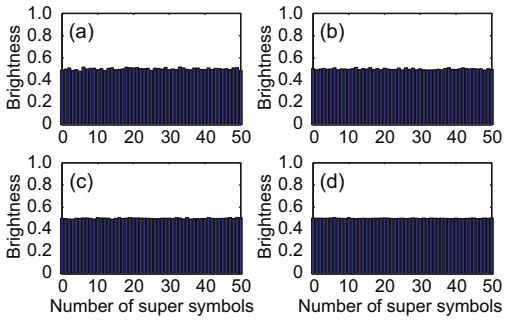

Fig. 19 Brightness in rate 2/4 RLL codes with $N = 4$ and $d_{\min} = 4$: (a) $M = 800$; (b) $M = 2000$; (c) $M = 5000$; (d) $M = 7500$

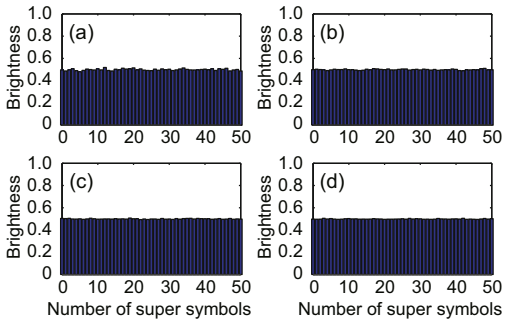

Fig. 20 Brightness in rate 3/5 RLL codes with $N = 4$ and $d_{\min} = 4$: (a) $M = 800$; (b) $M = 2000$; (c) $M = 5000$; (d) $M = 7500$

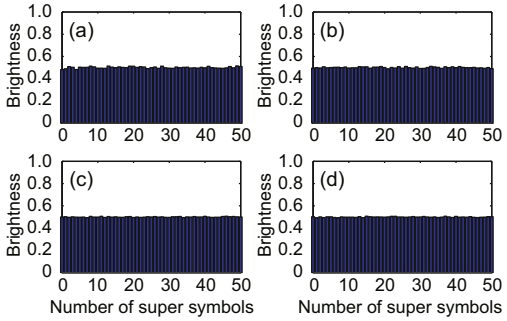

Fig. 21 Brightness in rate 3/5 RLL codes with $N = 8$ and $d_{\min} = 4$: (a) $M = 800$; (b) $M = 2000$; (c) $M = 5000$; (d) $M = 7500$

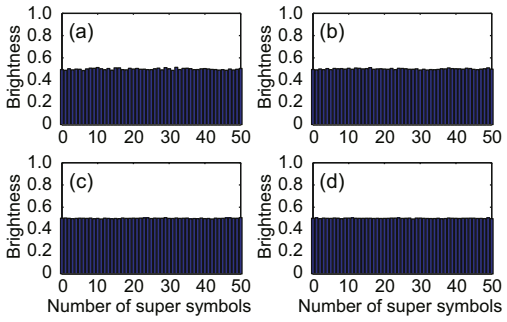

Fig. 22 Brightness in rate 4/6 RLL codes with $N = 4$ and $d_{\min} = 4$: (a) $M = 800$; (b) $M = 2000$; (c) $M = 5000$; (d) $M = 7500$

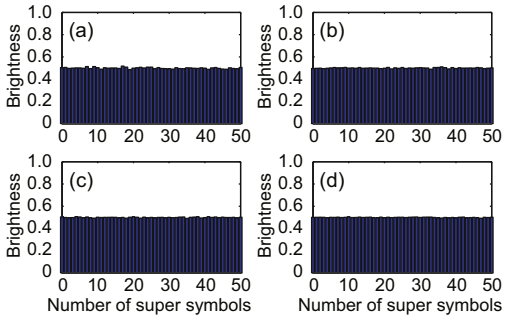

Fig. 23 Brightness in rate 4/6 RLL codes with $N = 8$ and $d_{\min} = 4$: (a) $M = 800$; (b) $M = 2000$; (c) $M = 5000$; (d) $M = 7500$

codes remains slightly smaller than that of the low-rate ones for $d_{\min} = 4$.

Similarly, the brightness level for other referenced RLL codes is shown in Figs. B1–B4 in Appendix B.

As M bits deliver a fairly uniform brightness level in all the proposed RLL codes shown in Figs. 10–23, flicker mitigation can be well handled. Because the data rates of VLC systems are in the order of gigabits per second, M can be much higher than 10 000. From Figs. 10–23, the approximate MFTP T_M value and the corresponding approximate M value, when the data rate is $Q^{'} = 5 \times 10^6$ bits/s for each studied code, are presented in Table 4.

Based on Table 4 and Figs. 10–23, to obtain fairly consistent flicker performance, we observe that the proposed RLL codes that have approximate $T_M = 1$ ms must transmit M bits at about 10 times and 25 times the optical clock rate compared to the referenced RLL codes with $T_M = 0.1$ ms and $T_M = 0.04$ ms, respectively. The proposed RLL codes with $T_M = 2.0$ ms require 20 times and 50 times the optical clock rate compared to the referenced RLL codes with $T_M = 0.1$ ms and $T_M =$

0.04 ms for the same brightness level, respectively. Meanwhile, we can determine that all the proposed RLL codes have approximate T_M with Miller codes. Furthermore, as shown in Table 4, these codes can significantly provide mitigate flicker and dimming constraints for high-rate data applications, which correspond to a larger M, and can satisfy the flickering time period within the MFTP, which is approximately 5 ms. This is easily satisfied, because the existing VLCs can provide a data rate of up to 1 Gb/s.

Dimming control is a good feature of VLC systems. In VLC systems, the communication function is included by modulating the visible light spectrum of the LEDs used for illumination. As stated in the IEEE 802.15.7 standard, several widely accepted methods, such as compensation symbols and pulse position modulation (PPM), are available for realizing brightness control. Each method has its own advantages and disadvantages. However, the above mentioned methods cannot perform error correction.

Our proposed RLL codes are designed to provide strong error-control ability and flicker control, which are the basic functions of RLL codes in VLC systems. One of the dimming control methods, adding compensation symbols, can be applied to our proposed RLL codes. Furthermore, the design of other dimming control methods is an interesting topic for further study.

4.3 Simulation results

To simulate the BER performance, we varied the noise variance σ^2 by changing the SNR using the normalized average energy per bit E_b for OOK-modulated VLC systems. Table 5 lists the simulation parameters of the proposed RLL codes and several existing ones.

Fig. 24 shows the BER curves of the four proposed RLL codes with $d_{\min}=2$ and $N=2$, and their comparison with the BER curves of rate 1/2 Miller, FM0/FM1, eMiller codes (Lu and Li, 2018), rate 3/4 codes with $d_{\min}=2$ and $N=2$ (Mejia et al., 2017), and rate 4/6 codes with $d_{\min}=2$ and $N=3$ (Mejia et al., 2017). It can be observed that the BER performance of the proposed RLL codes was significantly better than those of the rate 1/2 Miller, FM0/FM1, and eMiller codes. Furthermore, the proposed RLL codes achieved gains of approximately 1.5, 2.0, and 4.0 dB compared with the rate 1/2 eMiller design, rate 1/2 FM0/FM1 design, and the rate 1/2 Miller design, respectively, at a BER of 10^{-3}. With the increase in code rate R_c, the coding gain of the proposed RLL codes was more obvious

Table 4 Approximate MFTP T_M values and the corresponding approximate M values when the data rate is $Q^{'}=5\times10^6$ bits/s for each code

Code	R_c ($d_{\min}$, N)	M	$T_M=M/Q^{'}$ (ms)
Miller	1/2 (1, 4)	7500	1.5
FM0/FM1 (Lu and Li, 2018)	1/2 (2, 4)	500	0.1
eMiller (Lu and Li, 2018)	1/2 (2, 4)	500	0.1
Code (Mejia et al., 2017)	3/4 (2, 2)	200	0.04
	4/6 (2, 3)	200	0.04
	4/6 (4, 8)	500	0.10
Proposed code	2/3 (2, 2)	10 000	2.0
	3/4 (2, 2)	7500	1.5
	4/5 (2, 2)	7500	1.5
	5/6 (2, 2)	7500	1.5
	2/3 (3, 4)	10 000	2.0
	3/4 (3, 4)	7500	1.5
	4/5 (3, 8)	7500	1.5
	5/6 (3, 8)	5000	1.0
	2/4 (4, 2)	7500	1.5
	2/4 (4, 4)	7500	1.5
	3/5 (4, 4)	5000	1.0
	3/5 (4, 8)	7500	1.5
	4/6 (4, 4)	5000	1.0
	4/6 (4, 8)	7500	1.5

Table 5 Simulated parameters for several codes

Code	Modulator	R_c (input bit, output bit)	$d_{\min}$	N
Miller	OOK	1/2 (1, 2)	1	4
FM0/FM1 (Lu and Li, 2018)	OOK	1/2 (1, 2)	2	4
eMiller (Lu and Li, 2018)	OOK	1/2 (1, 2)	2	4
Code (Mejia et al., 2017)	OOK	3/4 (3, 4)	2	2
	OOK	4/6 (4, 6)	2	3
	OOK	4/6 (4, 6)	4	8
Proposed code	OOK	2/3 (2, 3)	2	2
	OOK	3/4 (3, 4)	2	2
	OOK	4/5 (4, 5)	2	2
	OOK	5/6 (5, 6)	2	2
	OOK	2/3 (2, 3)	3	4
	OOK	3/4 (3, 4)	3	4
	OOK	4/5 (4, 5)	3	8
	OOK	5/6 (5, 6)	3	8
	OOK	2/4 (2, 4)	4	2
	OOK	2/4 (2, 4)	4	4
	OOK	3/5 (3, 5)	4	4
	OOK	3/5 (3, 5)	4	8
	OOK	4/6 (4, 6)	4	4
	OOK	4/6 (4, 6)	4	8

at high SNRs. Moreover, the four proposed RLL codes achieved gains of approximately 1.1, 1.3, 1.6, and 1.75 dB compared with the rate 3/4 referenced codes with $d_{\min}=2$ and $N=2$ at a BER of 2×10^{-5}. We also compared our proposed RLL codes with the rate 4/6 referenced codes for $d_{\min}=2$ and $N=3$ at a BER of 5×10^{-6}; the coding gains obtained were approximately 0.4, 0.8, 1.1, and 1.3 dB.

Fig. 25 presents the BER performance of our other four proposed RLL codes with $d_{\min}=3$. The coding gains of different RLL codes with the same state number N were similar; the coding gain increased with the increase in the state number at high SNRs. However, the coding gain of the RLL codes decreased with the increase in the code rate, when SNR was less than 8 dB. Moreover, from Figs. 24 and 25, the BER performance of the proposed RLL codes with $d_{\min}=3$ was significantly better than those of the same code-rate RLL codes with $d_{\min}=2$.

In Fig. 26, we compared our proposed different rate RLL codes with $d_{\min}=4$ with other reported codes (Mejia et al., 2017). It can be observed that with the increase in state number N of the same rate codes, the BER performance became better. At a BER of 1×10^{-5}, rate 2/4 codes with $N=4$ achieved 0.4 dB gain, compared with those with $N=2$; rate 3/5 codes with $N=8$ achieved 0.8 dB gain, compared with those with $N=4$; rate 4/6 codes with $N=8$ achieved approximately 0.5 dB gain, compared with those with $N=4$. The proposed rate 4/6 RLL codes with $N=8$ achieved 0.6 dB coding gain compared with other reported codes with $d_{\min}=4$ (Mejia et al., 2017).

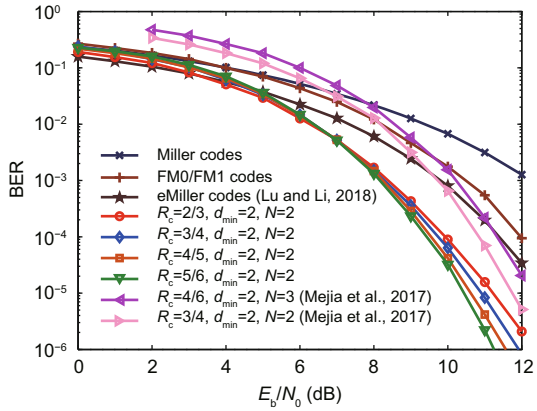

Fig. 24 BER comparison with the Viterbi algorithm of various codes

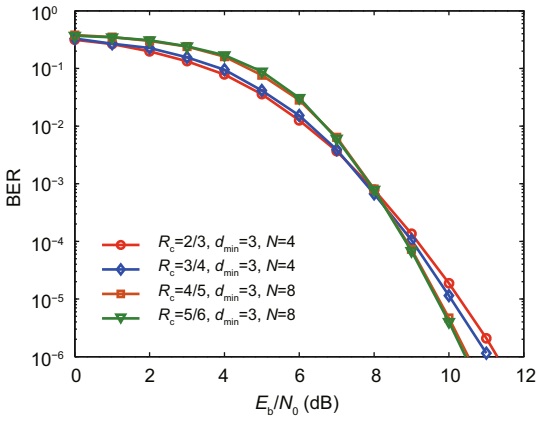

Fig. 25 BER comparison of RLL codes with different code rates with $d_{\min}=3$

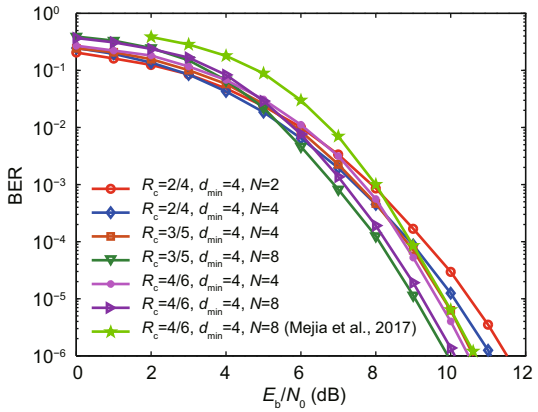

Fig. 26 BER comparison of RLL codes with different code rates with $d_{\min}=4$

5 Conclusions

In this study, a high-rate RLL coding scheme has been proposed for OOK-modulated VLC systems. The state splitting method and codeword-set partitioning criterion have been introduced in the design of two FSMs based on the FSM of high-rate RLL codes with $N=2$. The first RLL code has been proposed to improve the coding gain. This RLL coding design includes two stages: the consideration of the set-partitioning criterion of the codeword set and FSM design with small state numbers. Other RLL codes have been proposed to enhance the coding gain based on the first scheme. Moreover, two different choices for RLL code design have been provided with different state numbers for the FSM. The flicker control of the proposed RLL codes has been analyzed; it approached the 50% desirable brightness level in high-data-rate applications. The proposed FSM-

based RLL code design methods have presented a trade-off between flicker mitigation and BER performance. The minimum Hamming distances of the various proposed RLL codes have been computed. Furthermore, the BER performances of the various proposed RLL codes have been evaluated using a VLC system. Simulation results established that the various proposed coding schemes achieved superior error performance to existing codes.

Contributors

Zong-yan LI designed the research. Zong-yan LI and Hong-lu YU processed the data. Zong-yan LI drafted the manuscript. Bao-ling SHAN, De-xuan ZOU, and Shi-yin LI helped organize the manuscript. Zong-yan LI revised and finalized the paper.

Compliance with ethics guidelines

Zong-yan LI, Hong-lu YU, Bao-ling SHAN, De-xuan ZOU, and Shi-yin LI declare that they have no conflict of interest.

References

Babar Z, Mohd Izhar MA, Nguyen HV, et al., 2018. Unary-coded dimming control improves ON-OFF keying visible light communication. *IEEE Trans Commun*, 66(1):255-264. https://doi.org/10.1109/TCOMM.2017.2759271

Berman SM, Greenhouse DS, Bailey IL, et al., 1991. Human electroretinogram responses to video displays, fluorescent lighting, and other high frequency sources. *Optom Vis Sci*, 68(8):645-662. https://doi.org/10.1097/00006324-199108000-00012

Cao CZ, Fair I, 2019a. Construction of multi-state capacity-approaching variable-length constrained sequence codes with state-independent decoding. *IEEE Access*, 7:54746-54759. https://doi.org/10.1109/ACCESS.2019.2913339

Cao CZ, Fair I, 2019b. Minimal sets for capacity-approaching variable-length constrained sequence codes. *IEEE Trans Commun*, 67(2):890-902. https://doi.org/10.1109/TCOMM.2018.2873345

Cao CZ, Li DS, Fair I, 2019. Deep learning-based decoding of constrained sequence codes. *IEEE J Sel Areas Commun*, 37(11):2532-2543. https://doi.org/10.1109/JSAC.2019.2933954

Fang JB, Che Z, Jing ZL, et al., 2017. An efficient flicker-free FEC coding scheme for dimmable visible light communication based on polar ocdes. *IEEE Photon J*, 9(3):7903310. https://doi.org/10.1109/JPHOT.2017.2689744

IEEE, 2011. IEEE Standard for Local and Metropolitan Area Networks—Part 15.7: Short-Range Wireless Optical Communication Using Visible Light. https://doi.org/10.1109/IEEESTD.2011.6016195

Immink KS, 2004. Codes for Mass Data Storage Systems. Shannon Foundation, Eindhoven, the Netherlands, p.95-99.

Kim S, 2015. Adaptive FEC codes suitable for variable dimming values in visible light communication. *IEEE Photon Technol Lett*, 27(9):967-969. https://doi.org/10.1109/LPT.2014.2361171

Kim S, Jung SY, 2011. Novel FEC coding scheme for dimmable visible light communication based on the modified Reed-Muller codes. *IEEE Photon Technol Lett*, 23(20):1514-1516. https://doi.org/10.1109/LPT.2011.2163625

Kim S, Jung SY, 2013. Modified Reed-Muller coding scheme made from the bent function for dimmable visible light communications. *IEEE Photon Technol Lett*, 25(1):11-13. https://doi.org/10.1109/LPT.2012.2226210

Kim S, Park H, 2014. A coding scheme for visible light communication with wide dimming range. *IEEE Photon Technol Lett*, 26(5):465-468. https://doi.org/10.1109/LPT.2013.2296934

Lee SH, Kwon JK, 2010. Turbo code-based error correction scheme for dimmable visible light communication systems. *IEEE Photon Technol Lett*, 24(17):1463-1465. https://doi.org/10.1109/LPT.2012.2199104

Li ZY, Yu HL, Shan BL, et al., 2020. New run-length limited codes in on-off keying visible light communication systems. *IEEE Wirel Commun Lett*, 9(2):148-151. https://doi.org/10.1109/LWC.2019.2946146

Lu XX, Li J, 2016. Achieving FEC and RLL for VLC: a concatenated convolutional-Miller coding mechanism. *IEEE Photon Technol Lett*, 28(9):1030-1033. https://doi.org/10.1109/LPT.2016.2524578

Lu XX, Li J, 2018. New Miller codes for run-length control in visible light communications. *IEEE Trans Wirel Commun*, 17(3):1798-1810. https://doi.org/10.1109/TWC.2017.2785398

Mejia CE, Georghiades CN, Abdallah MM, et al., 2017. Code design for flicker mitigation in visible light communications using finite state machines. *IEEE Trans Commun*, 65(5):2091-2100. https://doi.org/10.1109/TCOMM.2017.2657518

Rajagopal S, Roberts RD, Lim SK, 2012. IEEE 802.15.7 visible light communication: modulation schemes and dimming support. *IEEE Commun Mag*, 50(3):72-82. https://doi.org/10.1109/MCOM.2012.6163585

Simon M, Divsalar D, 2006. Some interesting observation for certain line codes with application to RFID. *IEEE Trans Commun*, 54(4):583-586. https://doi.org/10.1109/TCOMM.2006.873063

Wang H, Kim S, 2016. Soft-input run-length limited decoding for visible light communication. *IEEE Photon Technol Lett*, 28(3):225-228. https://doi.org/10.1109/LPT.2015.2492607

Wang H, Kim S, 2018. Adaptive puncturing method for dimming in visible light communication with polar codes. *IEEE Photon Technol Lett*, 30(20):1780-1783. https://doi.org/10.1109/LPT.2018.2867371

Widner A, Franaszek P, 1983. A DC-balanced, partitioned-block, 8B/10B transmission code. *IBM J Res Dev*, 27(5):440-451. https://doi.org/10.1147/rd.275.0440

Appendix A: Possible mapping codes for the rate 4/5 and rate 5/6 RLL codes

Mapping relationships between the output codeword set and the input information bits are shown in Tables A1 and A2. See the next page for Table A2.

Table A1 Trellis diagram and mapping relationship between the output codeword set and input bits of rate 4/5 codes with $d_{\min}=3$

Current state	Next state	Output codeword set	Input bits	Current state	Next state	Output codeword set	Input bits
S_1^1	S_1^1	$\mathcal{C}(0,0)=\{c_1,c_{14}\}$	0000, 0001	S_2^1	S_2^1	$\mathcal{C}(1,4)=\{c_{17},c_{30}\}$	0000, 0001
	S_1^2	$\mathcal{C}(0,1)=\{c_2,c_{13}\}$	0010, 0011		S_2^2	$\mathcal{C}(1,5)=\{c_{18},c_{29}\}$	0010, 0011
	S_1^3	$\mathcal{C}(0,2)=\{c_4,c_{11}\}$	0100, 0101		S_2^3	$\mathcal{C}(1,6)=\{c_{20},c_{27}\}$	0100, 0101
	S_1^4	$\mathcal{C}(0,3)=\{c_7,c_8\}$	0110, 0111		S_2^4	$\mathcal{C}(1,7)=\{c_{23},c_{24}\}$	0110, 0111
	S_1^5	$\mathcal{C}(0,4)=\{c_{16},c_{31}\}$	1000, 1001		S_2^5	$\mathcal{C}(1,0)=\{c_0,c_{15}\}$	1000, 1001
	S_1^6	$\mathcal{C}(0,5)=\{c_{19},c_{28}\}$	1010, 1011		S_2^6	$\mathcal{C}(1,1)=\{c_3,c_{12}\}$	1010, 1011
	S_1^7	$\mathcal{C}(0,6)=\{c_{21},c_{26}\}$	1100, 1101		S_2^7	$\mathcal{C}(1,2)=\{c_5,c_{10}\}$	1100, 1101
	S_1^8	$\mathcal{C}(0,7)=\{c_{22},c_{25}\}$	1110, 1111		S_2^8	$\mathcal{C}(1,3)=\{c_6,c_9\}$	1110, 1111
S_1^2	S_1^1	$\mathcal{C}(0,3)=\{c_7,c_8\}$	0000, 0001	S_2^2	S_2^1	$\mathcal{C}(1,7)=\{c_{23},c_{24}\}$	0000, 0001
	S_1^2	$\mathcal{C}(0,2)=\{c_4,c_{11}\}$	0010, 0011		S_2^2	$\mathcal{C}(1,6)=\{c_{20},c_{27}\}$	0010, 0011
	S_1^3	$\mathcal{C}(0,1)=\{c_2,c_{13}\}$	0100, 0101		S_2^3	$\mathcal{C}(1,5)=\{c_{18},c_{29}\}$	0100, 0101
	S_1^4	$\mathcal{C}(0,0)=\{c_1,c_{14}\}$	0110, 0111		S_2^4	$\mathcal{C}(1,4)=\{c_{17},c_{30}\}$	0110, 0111
	S_1^5	$\mathcal{C}(0,7)=\{c_{22},c_{25}\}$	1000, 1001		S_2^5	$\mathcal{C}(1,3)=\{c_6,c_9\}$	1000, 1001
	S_1^6	$\mathcal{C}(0,6)=\{c_{21},c_{26}\}$	1010, 1011		S_2^6	$\mathcal{C}(1,2)=\{c_5,c_{10}\}$	1010, 1011
	S_1^7	$\mathcal{C}(0,5)=\{c_{19},c_{28}\}$	1100, 1101		S_2^7	$\mathcal{C}(1,1)=\{c_3,c_{12}\}$	1100, 1101
	S_1^8	$\mathcal{C}(0,4)=\{c_{16},c_{31}\}$	1110, 1111		S_2^8	$\mathcal{C}(1,0)=\{c_0,c_{15}\}$	1110, 1111
S_1^3	S_1^1	$\mathcal{C}(0,4)=\{c_{16},c_{31}\}$	0000, 0001	S_2^3	S_2^1	$\mathcal{C}(1,0)=\{c_0,c_{15}\}$	0000, 0001
	S_1^2	$\mathcal{C}(0,5)=\{c_{19},c_{28}\}$	0010, 0011		S_2^2	$\mathcal{C}(1,1)=\{c_3,c_{12}\}$	0010, 0011
	S_1^3	$\mathcal{C}(0,6)=\{c_{21},c_{26}\}$	0100, 0101		S_2^3	$\mathcal{C}(1,2)=\{c_5,c_{10}\}$	0100, 0101
	S_1^4	$\mathcal{C}(0,7)=\{c_{22},c_{25}\}$	0110, 0111		S_2^4	$\mathcal{C}(1,3)=\{c_6,c_9\}$	0110, 0111
	S_1^5	$\mathcal{C}(0,0)=\{c_1,c_{14}\}$	1000, 1001		S_2^5	$\mathcal{C}(1,4)=\{c_{17},c_{30}\}$	1000, 1001
	S_1^6	$\mathcal{C}(0,1)=\{c_2,c_{13}\}$	1010, 1011		S_2^6	$\mathcal{C}(1,5)=\{c_{18},c_{29}\}$	1010, 1011
	S_1^7	$\mathcal{C}(0,2)=\{c_4,c_{11}\}$	1100, 1101		S_2^7	$\mathcal{C}(1,6)=\{c_{20},c_{27}\}$	1100, 1101
	S_1^8	$\mathcal{C}(0,3)=\{c_7,c_8\}$	1110, 1111		S_2^8	$\mathcal{C}(1,7)=\{c_{23},c_{24}\}$	1110, 1111
S_1^4	S_1^1	$\mathcal{C}(0,7)=\{c_{22},c_{25}\}$	0000, 0001	S_2^4	S_2^1	$\mathcal{C}(1,3)=\{c_6,c_9\}$	0000, 0001
	S_1^2	$\mathcal{C}(0,6)=\{c_{21},c_{26}\}$	0010, 0011		S_2^2	$\mathcal{C}(1,2)=\{c_5,c_{10}\}$	0010, 0011
	S_1^3	$\mathcal{C}(0,5)=\{c_{19},c_{28}\}$	0100, 0101		S_2^3	$\mathcal{C}(1,1)=\{c_3,c_{12}\}$	0100, 0101
	S_1^4	$\mathcal{C}(0,4)=\{c_{16},c_{31}\}$	0110, 0111		S_2^4	$\mathcal{C}(1,0)=\{c_0,c_{15}\}$	0110, 0111
	S_1^5	$\mathcal{C}(0,3)=\{c_7,c_8\}$	1000, 1001		S_2^5	$\mathcal{C}(1,7)=\{c_{23},c_{24}\}$	1000, 1001
	S_1^6	$\mathcal{C}(0,2)=\{c_4,c_{11}\}$	1010, 1011		S_2^6	$\mathcal{C}(1,6)=\{c_{20},c_{27}\}$	1010, 1011
	S_1^7	$\mathcal{C}(0,1)=\{c_2,c_{13}\}$	1100, 1101		S_2^7	$\mathcal{C}(1,5)=\{c_{18},c_{29}\}$	1100, 1101
	S_1^8	$\mathcal{C}(0,0)=\{c_1,c_{14}\}$	1110, 1111		S_2^8	$\mathcal{C}(1,4)=\{c_{17},c_{30}\}$	1110, 1111

Appendix B: Brightness level for referenced RLL codes

Histograms of the brightness level for referenced RLL codes are shown in Figs. B1–B4.

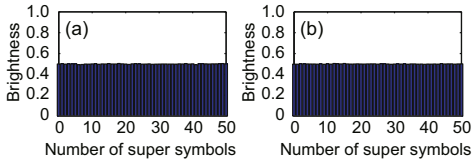

Fig. B1 Brightness in rate 1/2 Miller codes: (a) M=5000; (b) M=7500

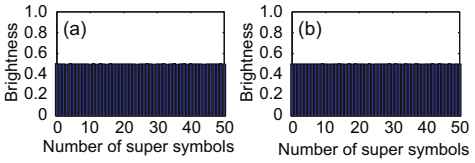

Fig. B2 Brightness in rate 3/4 RLL codes with N=2 and $d_{\min}$=2 (Mejia et al., 2017): (a) M=100; (b) M=200

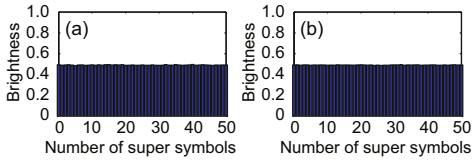

Fig. B3 Brightness in rate 4/6 RLL codes with N=8 and $d_{\min}$=4 (Mejia et al., 2017): (a) M=300; (b) M=500

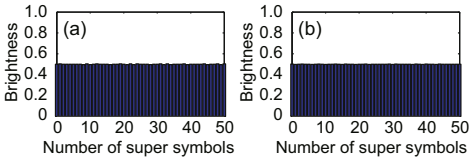

Fig. B4 Dimming value for rate 4/6 RLL codes with N=3 and $d_{\min}$=2 (Mejia et al., 2017): (a) M=100; (b) M=200

Table A2 Trellis diagram and mapping relationship between the output codeword set and input bits of rate 5/6 codes with $d_{\min} = 3$

Current state	Next state	Output codeword set	Input bits	Current state	Next state	Output codeword set	Input bits
S_1^1	S_1^1	$\mathcal{C}(0,0)=\{c_1,c_{14},c_{50},c_{61}\}$	0–3	S_2^1	S_2^1	$\mathcal{C}(1,4)=\{c_{17},c_{30},c_{34},c_{45}\}$	0–3
	S_1^2	$\mathcal{C}(0,1)=\{c_2,c_{13},c_{52},c_{59}\}$	4–7		S_2^2	$\mathcal{C}(1,5)=\{c_{18},c_{29},c_{36},c_{43}\}$	4–7
	S_1^3	$\mathcal{C}(0,2)=\{c_4,c_{11},c_{55},c_{56}\}$	8–11		S_2^3	$\mathcal{C}(1,6)=\{c_{20},c_{27},c_{39},c_{40}\}$	8–11
	S_1^4	$\mathcal{C}(0,3)=\{c_7,c_8,c_{49},c_{62}\}$	12–15		S_2^4	$\mathcal{C}(1,7)=\{c_{23},c_{24},c_{33},c_{46}\}$	12–15
	S_1^5	$\mathcal{C}(0,4)=\{c_{16},c_{31},c_{32},c_{47}\}$	16–19		S_2^5	$\mathcal{C}(1,0)=\{c_0,c_{15},c_{51},c_{60}\}$	16–19
	S_1^6	$\mathcal{C}(0,5)=\{c_{19},c_{28},c_{35},c_{44}\}$	20–23		S_2^6	$\mathcal{C}(1,1)=\{c_3,c_{12},c_{48},c_{63}\}$	20–23
	S_1^7	$\mathcal{C}(0,6)=\{c_{21},c_{26},c_{37},c_{42}\}$	24–27		S_2^7	$\mathcal{C}(1,2)=\{c_5,c_{10},c_{54},c_{57}\}$	24–27
	S_1^8	$\mathcal{C}(0,7)=\{c_{22},c_{25},c_{38},c_{41}\}$	28–31		S_2^8	$\mathcal{C}(1,3)=\{c_6,c_9,c_{53},c_{58}\}$	28–31
S_1^2	S_1^1	$\mathcal{C}(0,3)=\{c_7,c_8,c_{49},c_{62}\}$	0–3	S_2^2	S_2^1	$\mathcal{C}(1,7)=\{c_{23},c_{24},c_{33},c_{46}\}$	0–3
	S_1^2	$\mathcal{C}(0,2)=\{c_4,c_{11},c_{55},c_{56}\}$	4–7		S_2^2	$\mathcal{C}(1,6)=\{c_{20},c_{27},c_{39},c_{40}\}$	4–7
	S_1^3	$\mathcal{C}(0,1)=\{c_2,c_{13},c_{52},c_{59}\}$	8–11		S_2^3	$\mathcal{C}(1,5)=\{c_{18},c_{29},c_{36},c_{43}\}$	8–11
	S_1^4	$\mathcal{C}(0,0)=\{c_1,c_{14},c_{50},c_{61}\}$	12–15		S_2^4	$\mathcal{C}(1,4)=\{c_{17},c_{30},c_{34},c_{45}\}$	12–15
	S_1^5	$\mathcal{C}(0,7)=\{c_{22},c_{25},c_{38},c_{41}\}$	16–19		S_2^5	$\mathcal{C}(1,3)=\{c_6,c_9,c_{53},c_{58}\}$	16–19
	S_1^6	$\mathcal{C}(0,6)=\{c_{21},c_{26},c_{37},c_{42}\}$	20–23		S_2^6	$\mathcal{C}(1,2)=\{c_5,c_{10},c_{54},c_{57}\}$	20–23
	S_1^7	$\mathcal{C}(0,5)=\{c_{19},c_{28},c_{35},c_{44}\}$	24–27		S_2^7	$\mathcal{C}(1,1)=\{c_3,c_{12},c_{48},c_{63}\}$	24–27
	S_1^8	$\mathcal{C}(0,4)=\{c_{16},c_{31},c_{32},c_{47}\}$	28–31		S_2^8	$\mathcal{C}(1,0)=\{c_0,c_{15},c_{51},c_{60}\}$	28–31
S_1^3	S_1^1	$\mathcal{C}(0,4)=\{c_{16},c_{31},c_{32},c_{47}\}$	0–3	S_2^3	S_2^1	$\mathcal{C}(1,0)=\{c_0,c_{15},c_{51},c_{60}\}$	0–3
	S_1^2	$\mathcal{C}(0,5)=\{c_{19},c_{28},c_{35},c_{44}\}$	4–7		S_2^2	$\mathcal{C}(1,1)=\{c_3,c_{12},c_{48},c_{63}\}$	4–7
	S_1^3	$\mathcal{C}(0,6)=\{c_{21},c_{26},c_{37},c_{42}\}$	8–11		S_2^3	$\mathcal{C}(1,2)=\{c_5,c_{10},c_{54},c_{57}\}$	8–11
	S_1^4	$\mathcal{C}(0,7)=\{c_{22},c_{25},c_{38},c_{41}\}$	12–15		S_2^4	$\mathcal{C}(1,3)=\{c_6,c_9,c_{53},c_{58}\}$	12–15
	S_1^5	$\mathcal{C}(0,0)=\{c_1,c_{14},c_{50},c_{61}\}$	16–19		S_2^5	$\mathcal{C}(1,4)=\{c_{17},c_{30},c_{34},c_{45}\}$	16–19
	S_1^6	$\mathcal{C}(0,1)=\{c_2,c_{13},c_{52},c_{59}\}$	20–23		S_2^6	$\mathcal{C}(1,5)=\{c_{18},c_{29},c_{36},c_{43}\}$	20–23
	S_1^7	$\mathcal{C}(0,2)=\{c_4,c_{11},c_{55},c_{56}\}$	24–27		S_2^7	$\mathcal{C}(1,6)=\{c_{20},c_{27},c_{39},c_{40}\}$	24–27
	S_1^8	$\mathcal{C}(0,3)=\{c_7,c_8,c_{49},c_{62}\}$	28–31		S_2^8	$\mathcal{C}(1,7)=\{c_{23},c_{24},c_{33},c_{46}\}$	28–31
S_1^4	S_1^1	$\mathcal{C}(0,7)=\{c_{22},c_{25},c_{38},c_{41}\}$	0–3	S_2^4	S_2^1	$\mathcal{C}(1,3)=\{c_6,c_9,c_{53},c_{58}\}$	0–3
	S_1^2	$\mathcal{C}(0,6)=\{c_{21},c_{26},c_{37},c_{42}\}$	4–7		S_2^2	$\mathcal{C}(1,2)=\{c_5,c_{10},c_{54},c_{57}\}$	4–7
	S_1^3	$\mathcal{C}(0,5)=\{c_{19},c_{28},c_{35},c_{44}\}$	8–11		S_2^3	$\mathcal{C}(1,1)=\{c_3,c_{12},c_{48},c_{63}\}$	8–11
	S_1^4	$\mathcal{C}(0,4)=\{c_{16},c_{31},c_{32},c_{47}\}$	12–15		S_2^4	$\mathcal{C}(1,0)=\{c_0,c_{15},c_{51},c_{60}\}$	12–15
	S_1^5	$\mathcal{C}(0,3)=\{c_7,c_8,c_{49},c_{62}\}$	16–19		S_2^5	$\mathcal{C}(1,7)=\{c_{23},c_{24},c_{33},c_{46}\}$	16–19
	S_1^6	$\mathcal{C}(0,2)=\{c_4,c_{11},c_{55},c_{56}\}$	20–23		S_2^6	$\mathcal{C}(1,6)=\{c_{20},c_{27},c_{39},c_{40}\}$	20–23
	S_1^7	$\mathcal{C}(0,1)=\{c_2,c_{13},c_{52},c_{59}\}$	24–27		S_2^7	$\mathcal{C}(1,5)=\{c_{18},c_{29},c_{36},c_{43}\}$	24–27
	S_1^8	$\mathcal{C}(0,0)=\{c_1,c_{14},c_{50},c_{61}\}$	28–31		S_2^8	$\mathcal{C}(1,4)=\{c_{17},c_{30},c_{34},c_{45}\}$	28–31