IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 4, 2018, accepted January 26, 2018, date of publication January 30, 2018, date of current version November 14, 2019.

Digital Object Identifier 10.1109/ACCESS.2018.2799974

On the Concatenated Transmission Scheme With the Low-Complexity Symbol-Level Watermark Decoder for Recovering the Synchronization

YUAN LIU
Tianjin Key Laboratory of Wireless Mobile Communications and Power Transmission, College of Electronic and Communication Engineering, Tianjin Normal University, Tianjin 300387, China

Corresponding author: liuyuan@tjnu.edu.cn

This work was supported by the Doctor Fund of Tianjin Normal University under Grant 043135202-XB1711.

ABSTRACT The Davey–MacKay (DM) concatenated code employing the symbol-level watermark decoding algorithm is able to correct a large number of binary insertions, deletions, and substitutions, while it has the high computational complexity for recovering the synchronization. In the DM concatenated scheme, at the large insertion/deletion probability, in order to achieve the reliable output, the watermark decoder needs to perform the forward and backward passes on a very large trellis. In this paper, a threshold is selected to prevent the paths having very low forward and backward quantities from participating in the calculation of log-likelihood ratios from the watermark decoder. Simulation results show that great reduction in the complexity of the decoding algorithm is achieved at a very slight expense of accuracy.

INDEX TERMS DM concatenated scheme, forward-backward algorithm, binary insertions/deletions, low complexity.

I. INTRODUCTION

Insertions and deletions, which could induce the loss of synchronization, occur in many practical systems such as the imperfect synchronization in the communication systems [1], the file updates under general edits [2], [3], the bit-patterned media recording channel [4], [5], the multimedia watermarking methods in speech watermarking [6], and the differential pulse position modulation system [7]. Once insertions or deletions exist in the received sequences, symbol and block boundaries are unknown [8]–[11]. Considering that modern communication systems require more and more stringent synchronization constraints, alternative error-correcting codes for correcting insertions/ deletions and recovering the synchronization must be used for complement existing synchronization techniques [12]–[14].

The Davey-MacKay (DM) concatenated code was proposed to correct multiple binary insertions and deletions, and regarded as the most promising approach [15]. In this scheme, the watermark and non-binary low-density parity-check (NB-LDPC) codes were employed as inner and outer codes respectively. The watermark decoder called the inner decoder compared the received bits with the known watermark bits to identify the symbol and block boundaries [16]–[19]. For the watermark decoder, the unknown transmitted bits were treated as the additive noise. Since there were two types of the noise in the received sequences, the capability of the watermark decoder recovering the synchronization from the damaged sequences was limited.

In [20], using the symbol-level watermark decoding algorithm and considering more information about the transmitted code, the log-likelihood ratios (LLRs) provided by the watermark decoder were more reliable and the estimations outputted by the NB-LDPC decoder were more accurate. Therefore, the performance of the DM concatenated code was improved significantly. Since the width of the decoding trellis is proportional to the number of insertions and deletions in one block, at the large insertion/deletion probability, the scale of the decoding trellis is vast and the computations needed in the forward and backward passes over the trellis are very high [21]. For the large alphabet of the NB-LDPC code, the situation is worse.

In order to reduce the computational complexity, states with small values are removed in the trellis considering that small forward and backward values have little effect on the reliability of LLRs. The operations among all states is replaced by the operations among the $\delta < X$ most reliable states for the one symbol in the NB-LDPC code, where X is the number of states at each time in the trellis. Synchronization could still be recovered by decoding on the pruning trellis, because of the removed passes represent weak transitions due to the small drift at each step. Therefore, computations are reduced at a very slight expense of accuracy.

The rest of this paper is organized as follows. Section II introduces the original DM construction using the symbol-level watermark decoding algorithm. Section III describes the proposed watermark decoding scheme with low complexity and gives the comparison between the original and the proposed decoding schemes. Section IV illustrates the simulation results. Finally, some conclusions are drawn in Section V.

II. BACKGROUND

Following notations will be used throughout the paper.

- NB-LDPC code (N_L, K_L) over GF(q): N_L denotes the number of symbols in a codeword, K_L is the number of information symbols.
- P_i, P_d, P_t, P_s: insertion, deletion, transmission and substitution probabilities (respectively).
- x_i: the drift of the i-th position.
- Forward quantity $F_{ni}(x_{ni} = y)$: the probability that the drift at (ni)-th position is y and that the first $(ni-1+y)$ bits agree with $\mathbf{r}$.
- Backward quantity $B_{ni}(x_{ni} = y)$: the probability of outputting the tail of $\mathbf{r}$ given a drift of y at (ni)-th position.

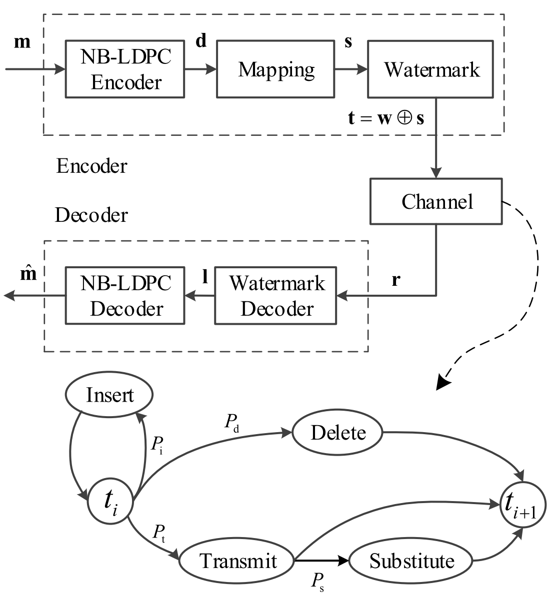

FIGURE 1. The schematic model of the original system.

A. ORIGINAL DM TRANSMISSION SCHEME

The schematic model of the original system [20] is depicted in Fig. 1. The information sequence $\mathbf{m}$ with the length of K_L symbols is encoded into a q-ary NB-LDPC code [22], [23] $\mathbf{d}$ with the length of N_L symbols, which is then mapped into a sparse code $\underline{\mathbf{s}}$ using a nonlinear mapping law μ. $q = 2^k$, where k is the number of bits associated with a symbol d_i. After transforming, each symbol d_i is associated with a vector $\mathbf{s}_i$ of length n bits, where $0 \leq i < N_L$. A binary pseudorandom sequence is selected as a watermark code $\mathbf{w}$, i.e., the inner code. The concatenation of $\underline{\mathbf{s}}$ and $\mathbf{w}$ is the modulo-2 addition. The whole length of the concatenated code $\mathbf{t}$ is $N = nN_L$, and the code rate is $R = kK_L/(nN_L)$.

The transmitted code $\mathbf{t}$ is sent over a channel involving bit insertions and deletions, which can model the imperfect sampling. As illustrated in Fig. 1, t_i enters a queue to be transmitted over the channel, and experiences the following three disturbances: a random symbol is inserted with a probability P_i, t_i is deleted with a probability P_d, or t_i is transmitted with a probability $P_\mathrm{t} = 1 - P_\mathrm{i} - P_\mathrm{d}$, the transmitted bit suffers a substitution with a probability P_s.

B. ENCODER

Take $q = 16$ for example, the nonlinear mapping law $\mu : d \rightarrow \underline{\mathbf{s}}$ is shown in Table 1, where $n = 5$. The frame structure of the concatenated code is illustrated in Fig. 2.

TABLE 1. The nonlinear mapping law for $q = 16$ and $n = 5$.

d_i	$\mathbf{s}_i$	d_i	$\mathbf{s}_i$
0	00000	8	01000
1	00001	9	01001
2	00010	10	01010
3	00011	11	10001
4	00100	12	01100
5	00101	13	10010
6	00110	14	10100
7	10000	15	11000

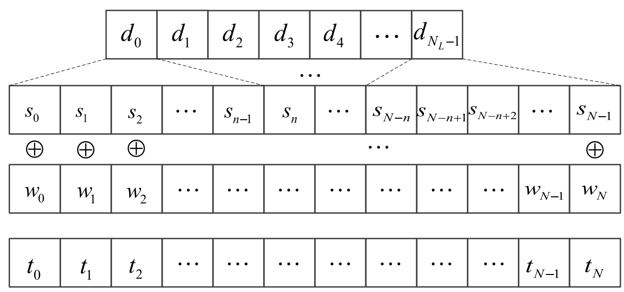

FIGURE 2. The frame structure of the concatenated code

C. DECODER USING THE SYMBOL-LEVEL WATERMARK DECODING ALGORITHM

The channel outputs the received sequence $\mathbf{r}$ which is corrupted by insertions, deletions and substitutions. In order to provide LLRs $\mathbf{l} = P(\mathbf{d}|\mathbf{r})$ for the NB-LDPC code, the watermark decoder recovers the symbol boundaries and compares $\mathbf{r}$ to all codewords in the codebook of the NB-LDPC code by employing the symbol-level forward-backward algorithm based on a decoding trellis.

Considering that insertions and deletions have effect on the positions of symbols, the drift of the i-th position x_i is defined, which is computed as follows.

$$x_i = D_I - D_D \tag{1}$$

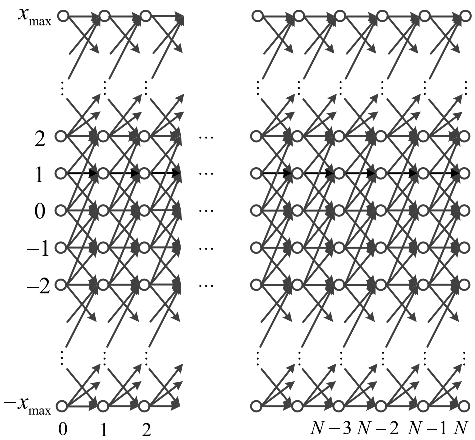

FIGURE 3. The decoding trellis for a block used in the watermark decoding algorithm. The maximum insertion length is 2 at each time.

where D_I is the number of insertions and D_D is the number of deletions in the channel from the first transmitted symbol t_0 to the point where the symbol t_i is ready to be transmitted. The drift is denoted as the state in the trellis. The size of the trellis is $X \times N = (2x_{\max} + 1) \times N$, where $x_{\max}$ is the maximum number of the drift and $x_{\max} = 5\sqrt{NP_{\mathrm{d}}/(1-P_{\mathrm{d}})}$ [15]. $P_{\mathrm{d}} = P_{\mathrm{i}}$. Fig. 3 gives the trellis used in the symbol-level watermark decoding algorithm of the DM construction. In this paper, the maximum insertion length I is assumed to be 2.

The LLR at the i-th position [20] is

$$\frac{P(d_i = a|\mathbf{r})}{P(d_i = 0|r)} = \frac{P(\mathbf{r}|d_i = a)}{P(\mathbf{r}|d_i = 0)} = \frac{\sum\limits_{x_{ni}, x_{n(i+1)}} F_{ni}(x_{ni}) M(x_{n(i+1)}|x_{ni}, d_i = a) B_{n(i+1)}(x_{n(i+1)})}{\sum\limits_{x_{ni}, x_{n(i+1)}} F_{ni}(x_{ni}) M(x_{n(i+1)}|x_{ni}, d_i = 0) B_{n(i+1)}(x_{n(i+1)})}, \tag{2}$$

where $0 < a \leq q-1$, $F_{ni}(x_{ni})$ and $B_{n(i+1)}(x_{n(i+1)})$ are forward and backward quantities respectively and are calculated as follows. The forward quantity [20]

$$\begin{aligned} F_{ni}(x_{ni} = y) &= P(r_0, \cdots, r_{ni-1+y}, x_{ni} = y) \\ &= \sum_{c, d_{i-1}} F_{n(i-1)}(x_{n(i-1)} = c) \\ &\quad \times P(d_{i-1}) P(r', x_{ni} = y | x_{n(i-1)} = c, d_{i-1}), \end{aligned} \tag{3}$$

where $\mathbf{r}' = (r_{n(i-1)+a}, \cdots, r_{ni+y-1})$, $d_{i-1} \rightarrow (s_{n(i-1)}, s_{n(i-1)+1}, s_{n(i-1)+2}, \cdots, s_{ni-1}, s_{ni})$, $y \in \{-x_{\max}, \cdots, 0, \cdots, x_{\max}\}$. The conditional probability $P(\mathbf{r}', x_{ni} = y|x_{n(i-1)} = c, d_{i-1})$ is the middle quantity which is calculated in (4).

The backward quantity [20]

$$\begin{aligned} B_{ni}(x_{ni} = y) &= P(r_{ni+y}, \cdots | x_{ni} = y) \\ &= \sum_{b, d_i} B_{n(i+1)}(b) P(d_i) P(r'', x_{n(i+1)} = b | x_{ni} = y, d_i), \end{aligned} \tag{4}$$

where $\mathbf{r}'' = (r_{ni+y}, \cdots, r_{n(i+1)+b-1})$, $d_i \rightarrow (s_{ni}, \cdots, s_{ni+n})$.

The middle quantity is calculated as follows.

$$\begin{aligned} &P(\mathbf{r}^0, x_{ni+\theta} = y | x_{ni} = z, d_i = a) \\ &\quad \times M(x_{ni+\theta} = y | x_{ni} = z, d_i = a) \\ &= \sum_{\varepsilon = y-I}^{y+1} M(x_{ni+\theta-1} = \varepsilon | x_{ni} = z, d_i = a) P_{\varepsilon y} Q_{\varepsilon y}^{ni+\theta-1}, \end{aligned} \tag{5}$$

where $0 \leq \theta < n$, $M(x_{ni} = z) = 1$ and is set to 0 at other drifts, $\mathbf{r}^0 = (r_{ni+x_{ni}}, \cdots, r_{ni+l+x_{ni+l}})$, the branch quantity $P_{zy} \cdot Q_{zy}^{ni+\theta-1}$ is the output probability [20].

Finally, the NB-LDPC code receives the LLRs and generates the estimation of $\hat{\mathbf{m}}$.

III. PROPOSED DECODING SCHEME

Parameters q, N_L and X affect the number of computations needed to perform the forward, middle and backward passes. With the alphabet q and the code length N_L increasing, the trellis becomes larger and the complexity becomes higher. It is known that each bit t_i enters the channel to generate strings with the length between 0 and $I+1$. Thus, The relative drift for one symbol takes value in $\{-1, 0, 1, 2, \cdots, I\}$. Since symbols only drifts a small amount at each time, most transitions among states in the trellis are weak or disallowed.

In this paper, the decoding path is constrained along a narrow corridor around the most likely path in the trellis of the modified watermark decoding scheme. Therefore, the irregular decoding trellis is used in the calculations of forward/backward quantities through a block. The procedure of passing information from the watermark decoder to the NB-LDPC decoder is illustrated in Fig. 4.

A. TRELLIS PRUNING RULE

The pruning rule is used for classifying the states involved in forward/backward vectors into *dominant* set and *non-dominant* set. The pruning threshold for the classification is related to the forward/backward quantities and independent of the middle quantities. Only the dominant set is used for watermark decoding. In this algorithm, the key idea is to choose the δ most reliable entries from each X-sized forward/backward vector, which is the ni-th column of the trellis in the watermark decoder shown in Fig. 4, $0 \leq i < N_L$.

In order to reduce the computational complexity of the watermark decoder, the size of the dominant set is fixed to $\delta \times N_L$ and should be as small as possible. Since N_L is the length of symbols in the NB-LDPC code, the scaling factor δ is expected to be small.

B. CALCULATION OF THE FORWARD/ BACKWARD QUANTITY

Define the following index set

$$\mathbf{\Omega} = \{\Omega_0, \Omega_1, \cdots, \Omega_i, \cdots, \Omega_{N_L-1}\} \tag{6}$$

as the dominant set. $\{x_i | x_i \in \Omega_i\}$ takes value in the alphabet $\{y_0, y_1, \cdots, y_j, \cdots, y_{\delta-1}\}$, where $0 \leq j < \delta$.

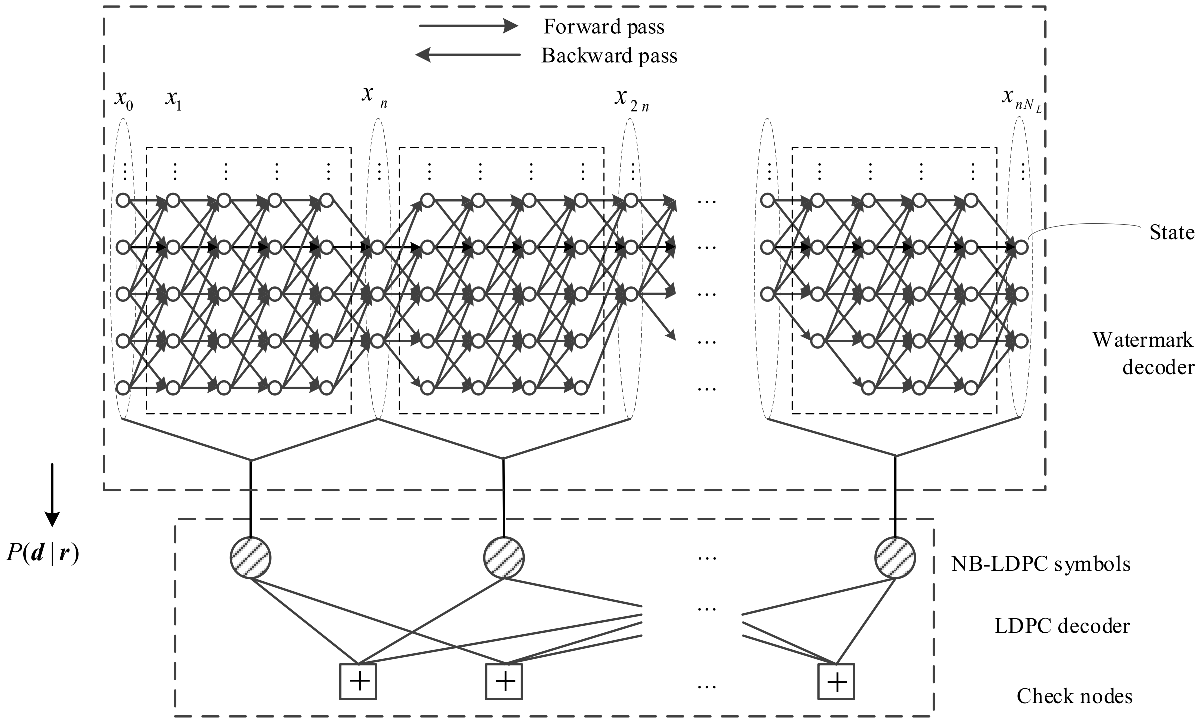

FIGURE 4. The procedure of passing information from the watermark decoder to the NB-LDPC decoder, where $n = 5$, $I = 2$.

The forward quantity at ni-th position is calculated as follows.

$$F_{ni}(x_{ni} = y) = \sum_{c \in \Omega_{i-1}, d_{i-1}} F_{n(i-1)}(x_{n(i-1)} = c) \times P(d_{i-1}) P(\mathbf{r}', x_{ni} = y | x_{n(i-1)} = c, d_{i-1}) \quad (7)$$

where, $P(\mathbf{r}', x_{ni} = y|x_{n(i-1)} = c, d_{i-1})$ is computed by performing the forward pass from Ω_{i-1} to $\{-x_{\max}, \cdots, 0, x_{\max}\}$ on the irregular trellis, and is stored in $M(x_{ni} = y|x_{n(i-1)} = c, d_{i-1})$. Each of the $q$possible values of d_i is stored. At ni-th position, states having the most δ largest forward quantities build the set Ω_i.

The lookup-table approach is used in the computation of the middle quantity $M(x_{n(i+1)} = y|x_{ni} = z, d_i = a)$, where y and z take the value in Ω_{i+1} and Ω_i respectively.

For the calculation of the backward quantity,

$$B_{ni}(x_{ni} = y) = \sum_{b \in \Omega_{i+1}, d_i} B_{n(i+1)}(b) P(d_i) \times P(\mathbf{r}'', x_{n(i+1)} = b|x_{ni} = y, d_i), \quad (8)$$

C. COMPLEXITY ANALYSIS

In this section, the computational complexity of the original and the proposed watermark decoding algorithms is analyzed. Computations are determined by counting the number of multiplications, additions involved in the calculations of forward, middle, and backward quantities. Let's take $I = 2$ as an example. Considering that the complexity of the branch quantity is independent of the time, a forward pass from ni-th position to $(ni+1)$-th position in the watermark decoding trellis is illustrated in Fig. 5, the calculations of branch quantities are given in eq. (8), and the computations are shown in Table 2. α_I is the normalizing constant. Finally, the number of multiplications and additions needed in the original and proposed passes is analyzed in Table 3. Since the computations involved in the calculations of the forward and backward passes are the same, only computations for performing forward passes are shown as follows.

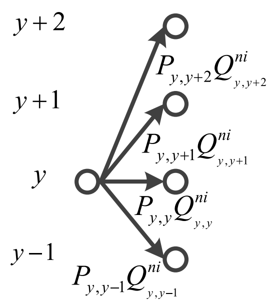

FIGURE 5. A forward pass for one bit in the watermark decoding trellis, $I = 2$, $y \in \{-x_{\max}, \cdots, 0, \cdots, x_{\max}\}$.

As observed from Table 3, the proposed decoding scheme attains a reduction in the computations. The complexity ratio ρ is denoted as

$$\rho = \frac{\{(I+2)(10I+15)+2\delta\} \cdot \delta}{\{(I+2)(10I+15)+2X\} \cdot X}. \quad (9)$$

Furthermore, the proposed decoding scheme becomes very efficient as P_{d} increases because X becomes larger

TABLE 2. The computations needed in a forward pass for one bit.

Operation	Multiplication	Addition
$P_{y,y+I}Q^{ni}_{y,y+I}$	6	5
$P_{y,y+I-1}Q^{ni}_{y,y+I-1}$	8	6
⋮	⋮	⋮
$P_{y,y+1}Q^{ni}_{y,y+1}$	8	6
$P_{y,y}Q^{ni}_{y,y}$	6	6
$P_{y,y-1}Q^{ni}_{y,y-1}$	0	0
Total	4+8I	5+6I

TABLE 3. The computations needed in the original and proposed watermark decoding schemes at $n = 5$.

Operation	Scheme	Multiplication	Addition
Middle pass	Original	$(I+2)(10I+15)XqN_{\mathrm{L}}$	$10(I+1)^2XqN_{\mathrm{L}}$
	Proposed	$(I+2)(10I+15)\,\delta\, qN_{\mathrm{L}}$	$10(I+1)^2\,\delta\, qN_{\mathrm{L}}$
Forward Pass	Original	$\{(I+2)(10I+15)X +2X^2\}qN_{\mathrm{L}}$	$\{10(I+1)^2q +Xq-1\}XN_{\mathrm{L}}$
	Proposed	$\{(I+2)(10I+15)\,\delta +2\,\delta^2\}qN_{\mathrm{L}}$	$\{10(I+1)^2q +\delta\, q-1\}\,\delta\, N_{\mathrm{L}}$

with the increasing of P_{d}. $X = 2x_{\max}+1 = 10\sqrt{NP_{\mathrm{d}}/(1-P_{\mathrm{d}})}+1$.

The selection of the width of the narrow corridor δ is important to the accuracy of LLRs and the computational complexity of the watermark decoding algorithm. Large width will improve the accuracy of LLRs but lead to the little reduction in the complexity. Conversely, small width will provide significant reduction in the computations but cause the performance lose. In the simulation results, the analysis of the selection of X will be described.

$$P_{x_{ni},x_{ni+1}}Q^{ni}_{x_{ni},x_{ni+1}} = \begin{cases} 2^{-I}\alpha_I P_{\mathrm{i}}^I P_{\mathrm{t}}[(s_{ni}\oplus s^*)P_{\mathrm{s}} \\ +(s_{ni}\oplus s^*\oplus 1)(1-P_{\mathrm{s}})], & x_{ni}=x_{ni+1}+I \\ a_I\{2^{-(x_{ni+1}-x_{ni}+1)}P_{\mathrm{i}}^{(x_{ni+1}-x_{ni}+1)}P_{\mathrm{d}} \\ +2^{-(x_{ni+1}-x_{ni})}\cdot P_{\mathrm{i}}^{(x_{ni+1}-x_{ni})} \\ P_{\mathrm{t}}[(s_{ni}\oplus s^*)P_{\mathrm{s}}+(s_{ni}\oplus s^*\oplus 1)(1-P_{\mathrm{s}})]\}, \\ & x_{ni+1}<x_{ni}<x_{ni+1}+I \\ 2^{-1}a_I P_{\mathrm{i}}P_{\mathrm{d}}+P_{\mathrm{t}}[(s_{ni}\oplus s^*)P_{\mathrm{s}} \\ +(s_{ni}\oplus s^*\oplus 1)(1-P_{\mathrm{s}})], \\ & x_{ni+1}=x_{ni} \\ P_{\mathrm{d}}, & x_{ni+1}-x_{ni}=-1 \end{cases} \quad (10)$$

IV. SIMULATION RESULTS

In this section, the efficiency of the proposed scheme is demonstrated as follows. The NB-LDPC code of length 999 symbols over GF($q = 16$) and the mapping parameter $n = 5$ are chosen. A binary pseudorandom sequence of length 4995 bits is generated as the watermark code $I = 2$. $P_{\mathrm{i}} = P_{\mathrm{d}}$. $P_{\mathrm{s}} = 0$. The forward quantity of the first position in the first block are initialized as follows.

$$F_0(x) = \begin{cases} 1, & x = 0, \\ 0, & \text{otherwise.} \end{cases} \quad (11)$$

At the beginning of the next block, the quantities are computed as follows.

$$F_0(x) = \begin{cases} F_N(x+\hat{x}_N), & |x+\hat{x}_N| \le x_{\max}, \\ 0, & \text{otherwise.} \end{cases} \quad (12)$$

The backward quantities are usually set to the equal probability as follows:

$$B_{N+5x_{\max}}(t) = \begin{cases} 1/X, & |x| \le x_{\max}, \\ 0, & \text{otherwise.} \end{cases} \quad (13)$$

A belief-propagation algorithm in log-domain is used in the NB-LDPC decoder and the maximum number of iterations is set to 20.

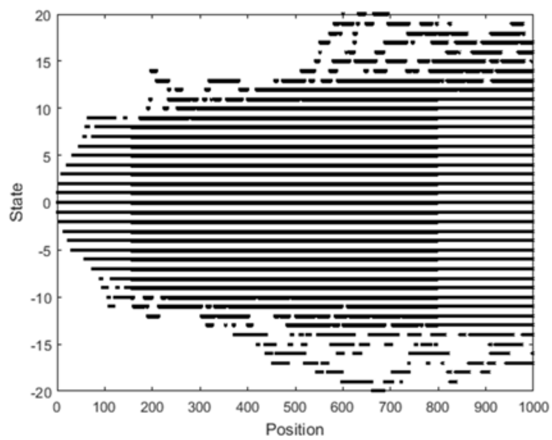

FIGURE 6. The true drift paths within the received blocks. The number of samples is 1000.

The true drift paths within the received blocks are analyzed and plotted in Fig. 6, where 1000 samples are employed in the simulation. As shown in Fig. 6, at the front of the block, the synchronization only drifts a small amount, and many states have not been accessed. As the position increases, the drift becomes large. This phenomenon reveals that the original decoding scheme is wasteful.

Furthermore, a true drift path within a received block is selected from Fig. 6 and shown in Fig. 7. Moreover, the forward and backward quantities in a block are illustrated in the Fig. 8 and Fig. 9 respectively. From Fig. 7, it is obviously shown that most transitions among states in the trellis are weak or disallowed. Symbols only drifts a small amount at each time. Furthermore, as shown in Figs. 8 and 9, the values of forward/backward quantities decrease with the increasing of the difference between the true drift and the estimated drift at each position, where the drift is denoted as the state in the trellis. Very small forward and backward quantities have little effect on the output from the watermark decoder. Therefore, decoding along a narrow corridor around the most likely path

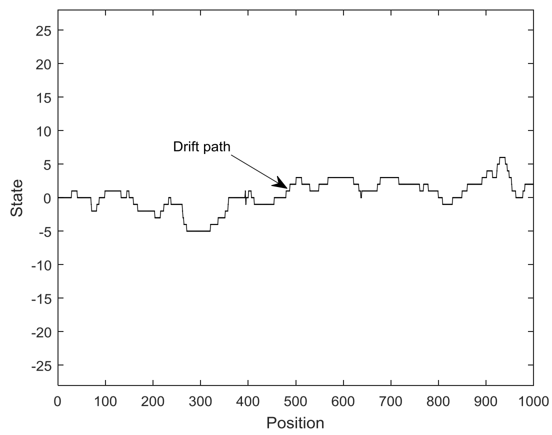

FIGURE 7. The true drift path within a received block.

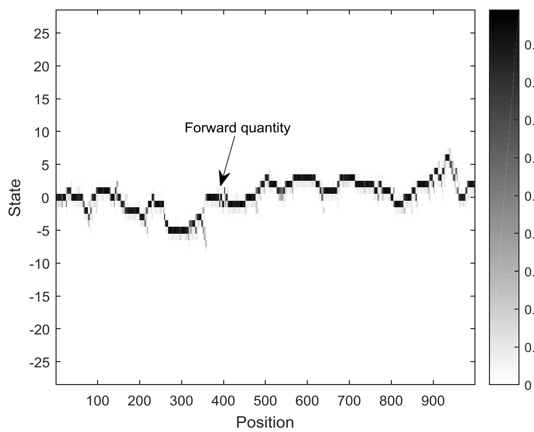

FIGURE 8. Forward quantities in a block.

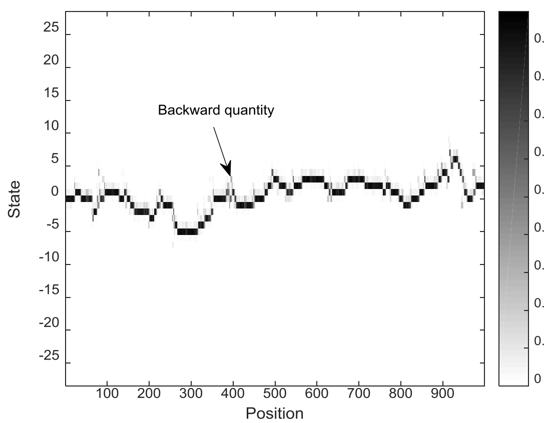

FIGURE 9. Backward quantities in a block.

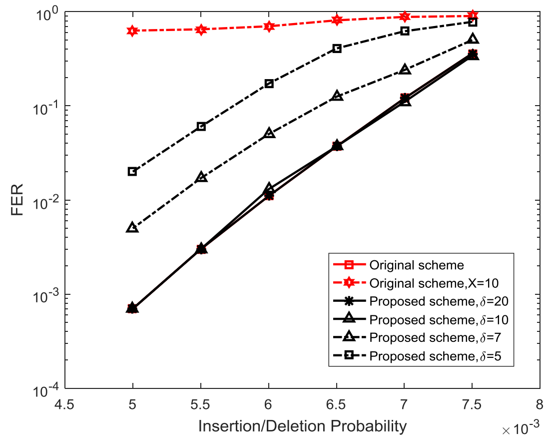

FIGURE 10. FERs of the proposed scheme. $P_{s.} = 0$

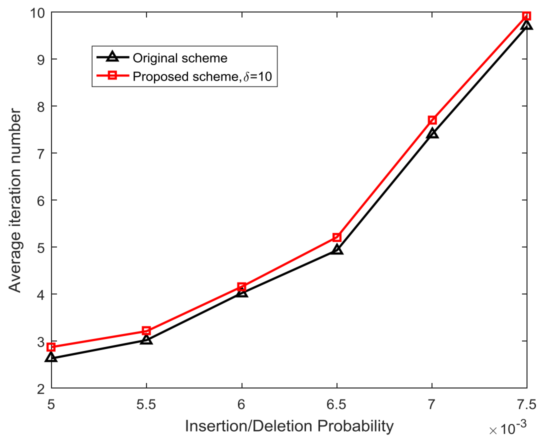

FIGURE 11. The average iteration number of the NB-LDPC decoders employed in the proposed and original schemes.

is generally good enough for recovering the synchronization and is a cheaper alternative approach.

In order to demonstrate the efficiency of the proposed decoding scheme, the proposed decoding scheme with different δ is compared to the original scheme. Frame error rates (FERs) of the proposed scheme with $\delta = 5, 7, 10, 20$ are simulated in Fig. 10. If $P_d = 0.0065, 0.006, 0.0055, 0.005$, then $X = 59, 57, 55, 51$ respectively. In Fig. 10, it is shown that FERs of the proposed scheme closely matches with the original result for $\delta \geq 10$. In order to achieve the low complexity at a very slight performance loss, $\delta = 10$ is selected in this paper. For $\delta = 10$, the average complexity ratio $\rho \approx 11\%$. Furthermore, X of the original scheme is set to δ, and then two approaches with the same complexity are compared. As illustrated from the figure, the proposed scheme for $\delta = 10$ performs substantially better than the original scheme for $X = 10$. Furthermore, the average iteration number of the NB-LDPC decoders employed in the proposed and original schemes is analyzed in Fig. 11. The computations of NB-LDPC decoders employed in the proposed and the original schemes are very close. It is clear that the reduction in the complexity of the watermark decoder does not lead to the increasing of the complexity of the NB-LDPC decoder. Therefore, the proposed decoding scheme is able to reduce the computations with little performance loss compared to the original scheme.

V. CONCLUSIONS

An irregular decoding trellis is used in the watermark decoding scheme of the concatenated code for correcting binary insertions and deletions. In this proposed decoding scheme, the decoding path is constrained along a narrow corridor around the most likely path in the trellis. Since most of reliable states are involved in the narrow corridor, the LLRs outputted by the watermark decoder are accurate enough. As a result, significant reduction in decoding complexity is achieved by pruning states with very small forward/backward quantities.

REFERENCES

[1] H. Mercier, V. K. Bhargava, and V. Tarokh, "A survey of error-correcting codes for channels with symbol synchronization errors," *IEEE Commun. Surveys Tuts.*, vol. 12, no. 1, pp. 87–94, 1st Quart., 2010.

[2] Q. Wang, S. Jaggi, M. Médard, V. R. Cadambe, and M. Schwartz, "File updates under random/arbitrary insertions and deletions," *IEEE Trans. Inf. Theory*, vol. 63, no. 10, pp. 6487–6513, Oct. 2017.

[3] Q. Wang, M. Médard, and M. Skoglund, "Efficient compression algorithm for file updates under random insertions and deletions," in *Proc. IEEE Inf. Theory Workshop*, Cambridge, U.K., Sep. 2016, pp. 434–438.

[4] X. Jiao and M. A. Armand, "Soft-input inner decoder for the Davey-MacKay construction," *IEEE Commun. Lett.*, vol. 16, no. 5, pp. 722–725, May 2012.

[5] T. Wu and M. A. Armand, "Marker codes on BPMR write channel with data-dependent written-in errors," *IEEE Trans. Magn.*, vol. 51, no. 8, Aug. 2015, Art. no. 3101007.

[6] D. J. Coumou and G. Sharma, "Insertion, deletion codes with feature-based embedding: A new paradigm for watermark synchronization with applications to speech watermarking," *IEEE Trans. Inf. Forensics Security*, vol. 3, no. 2, pp. 153–165, Jun. 2008.

[7] W. Chen and Y. Liu, "Efficient transmission schemes for correcting insertions/deletions in DPPM," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Kuala Lumpur, Malaysia, May 2016, pp. 1–6.

[8] C. Schoeny, A. Wachter-Zeh, R. Gabrys, and E. Yaakobi, "Codes correcting a burst of deletions or insertions," *IEEE Trans. Inf. Theory*, vol. 63, no. 4, pp. 1971–1985, Apr. 2017.

[9] J. Ullman, "Near-optimal, single-synchronization-error-correcting code," *IEEE Trans. Inf. Theory*, vol. 12, no. 4, pp. 418–424, Oct. 1966.

[10] A. S. J. Helberg and H. C. Ferreira, "On multiple insertion/deletion correcting codes," *IEEE Trans. Inf. Theory*, vol. 48, no. 1, pp. 305–308, Jan. 2002.

[11] T. G. Swart and H. C. Ferreira, "A note on double insertion/deletion correcting codes," *IEEE Trans. Inf. Theory*, vol. 49, no. 1, pp. 269–273, Jan. 2003.

[12] M. F. Mansour and A. H. Tewfik, "Convolutional decoding in the presence of synchronization errors," *IEEE J. Sel. Areas Commun.*, vol. 28, no. 2, pp. 218–227, Feb. 2010.

[13] V. Buttigieg and N. Farrugia, "Improved bit error rate performance of convolutional codes with synchronization errors," in *Proc. IEEE Int. Conf. Commun.*, London, U.K., Jun. 2015, pp. 4077–4082.

[14] M. F. Mansour and A. H. Tewfik, "A turbo coding scheme for channels with synchronization errors," *IEEE Trans. Commun.*, vol. 60, no. 8, pp. 2091–2100, Aug. 2012.

[15] M. C. Davey and D. J. C. MacKay, "Reliable communication over channels with insertions, deletions, and substitutions," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 687–698, Feb. 2001.

[16] R. Yazdani and M. Ardakani, "Reliable communication over non-binary insertion/deletion channels," *IEEE Trans. Commun.*, vol. 60, no. 12, pp. 3597–3608, Dec. 2012.

[17] E. A. Ratzer, "Marker codes for channels with insertions and deletions," *Ann. Telecommun.*, vol. 60, no. 1, pp. 29–44, Feb. 2005.

[18] T. Wu and M. A. Armand, "The Davey-MacKay coding scheme for channels with dependent insertion, deletion, and substitution errors," *IEEE Trans. Magn.*, vol. 49, no. 1, pp. 489–495, Jan. 2013.

[19] F. Wang, D. Fertonini, and T. M. Duman, "Symbol-level synchronization and LDPC code design for insertion/deletion channels," *IEEE Trans. Commun.*, vol. 59, no. 5, pp. 1287–1297, May 2011.

[20] J. A. Briffa, H. G. Schaathun, and S. Wesemeyer, "An improved decoding algorithm for the Davey-MacKay construction," in *Proc. IEEE Int. Conf. Commun.*, Cape Town, South Africa, May 2010, pp. 1–5.

[21] X. Jiao and M. A. Armand, "Interleaved LDPC codes, reduced-complexity inner decoder and an iterative decoder for the Davey-MacKay construction," in *Proc. IEEE Int. Symp. Inf. Theory*, St. Petersburg, Russia, Jul./Aug. 2011, pp. 742–746.

[22] S. Yeo and I.-C. Park, "Improved hard-reliability based majority-logic decoding for non-binary LDPC codes," *IEEE Commun. Lett.*, vol. 21, no. 2, pp. 230–233, Feb. 2017.

[23] K. Deka, A. Rajesh, and P. K. Bora, "EXIT chart analysis of puncturing for non-binary LDPC codes," *IEEE Commun. Lett.*, vol. 18, no. 12, pp. 2089–2092, Dec. 2014.

YUAN LIU received the B.E. degree from Tianjin Polytechnic University, Tianjin, China, in 2010, and the Ph.D. degree from Tianjin University, Tianjin, in 2017. She is currently a Lecturer with the School of Electronic and Communication Engineering, Tianjin Normal University. Her research interests include communication theory and synchronization error correcting scheme.