Coding for Short Messages in Multipath Underwater Acoustic Communication Channels

Mohammadhossein Behgam and Y. Rosa Zheng
Department of Electrical and Computer Engineering
Missouri University of Science and Technology
Rolla, Missouri 65409, USA
Email: {mb279, zhengyr}@mst.edu

Zhiqiang Liu
Acoustics Signal Processing and System Branch
United States Naval Research Laboratory
Washington DC 20375, USA
Email: zhiqiang.liu@nrl.navy.mil

***Abstract*—This paper applies the full tail-biting (FTB) convolutional codes to short data packets and evaluates their performance in underwater acoustic communication by computer simulation and an ocean experiment. The simulation results for AWGN channels show that the FTB codes achieve the similar bit error rate (BER) performance as the zero-tailing convolutional (ZTC) codes regardless of block lengths, while the direct-truncate convolutional (DTC) codes suffer from BER degradation, specially with short block lengths. Both simulation and ocean experimental results demonstrate that the FTB codes are excellent candidates for underwater acoustic communication systems where short data blocks and strong error correction codes are needed.**

***Index Terms*—Tail-biting convolutional codes, underwater acoustic communications, circular Viterbi algorithm, Internet of Things, Internet of Underwater Things.**

I. Introduction

Wireless communication systems require strong error correction coding to combat time-varying fading channels. The multipath delay spread and Doppler-induced frequency spread lead to random channel impulse responses that causes severe inter-symbol interference and burst bit errors. Channel impairments in underwater acoustic (UWA) communications are more severe than terrestrial RF communications, in that both multipath delay spread and Doppler spread are extremely large. In addition, the available bandwidth in UWA is very small (on the order of kilo-Hertz) and the propagation velocity of sound wave in water is very slow (on the order of 1500 m/s). These factors call for extra strong coding schemes [1], [2] and complex channel equalization techniques [3], [4] to improve the reliability of UWA communication.

The recent development in massive machine-to-machine (M2M) communications or Internet of Things (IoT) will support thousands of sensors and smart devices that only transmit short packets sporadically [5]. Those messages are normally a few bytes in length. Internet of Underwater Things [6] also attracts great attention in recent years, which aims to connect underwater sensors, instruments, robots, gliders, and autonomous underwater vehicles (AUVs) to the Internet via underwater wireless links. These underwater devices mostly need short command and control or sensing messages in order to stay connected and communicate with each other. The classical iteratively-decodable codes developed for long packets tend to suffer from coding rate loss and performance loss when applied to short packets [5], [7] because of the asymptotic nature of these algorithms. Using unnecessary long packets for short data blocks also results in waste of bandwidth resources and increase of latency.

Code design for short data blocks is surveyed for many FEC codes [7], including LDPC codes, turbo codes, BCH codes, polar codes, and convolutional codes. It is reported that the LDPC codes exhibit advantages mostly for long data blocks with lengths on the order of over 1000 bits. In contrast, convolutional codes are known to perform well for short data blocks with lengths on the order of 100 bits [8], [9]. Convolutional codes are described by $(n, 1, K)$ where the code rate is $R = 1/n$ and constraint length is K. The higher the constraint length K or the lower the coding rate usually means the better error correction capability. Let the data block length be L. For short data blocks with $L < 100$ bits, convolutional codes with a large constraint length would suffer from rate loss. When L and K are on the same order, the rate loss is significant. To reduce the rate loss, direct truncation (DT) or tail-biting (TB) techniques are usually used [10], resulting in the Direct-Truncation Convolutional (DTC) code and Full Tail-Biting Convolutional (FTB) codes, respectively.

A. Convolutional Coding Techniques

1) Zero-Tailing: Using this technique, the encoder starts from state zero, shifts all the raw data bits through its memory to generate codewords, and after encoding the last bit in the data block forces itself to end at state zero by adding $(K - 1)$ zeros to the end of the data block. With this method, the effective coding rate becomes $L/((L + K - 1)n)$ instead of $1/n$.

2) Direct-Truncation: This technique has similar starting point like zero-tailing, but unlike the previous technique, the encoder simply stops when reached to the end of the data block, keeping the code rate at $1/n$. Due to stopping at an

The work of M. Behgam and Y.R. Zheng was supported in part by the U.S. National Science Foundation under Grant ECCS-1646548 and the Wilkens Missouri Telecommunications Endowment at Missouri University of Science and Technology. The work of Z. Liu was supported by the US Office of Naval Research.

978-1-5386-6442-1/18/$31.00 © 2018 IEEE

978-1-5386-4814-8/18/$31.00 ©2018 IEEE

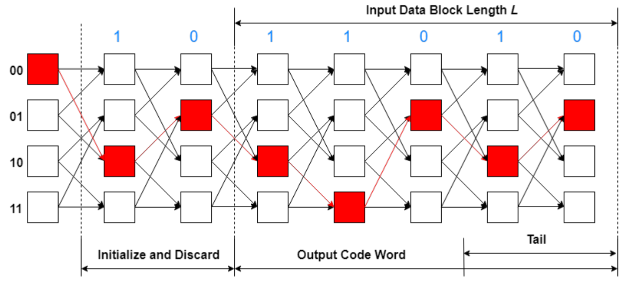


Fig. 1. Encoder trellis diagram for tail-biting codes with $K = 3$ and rate 1/2, for input 11010.

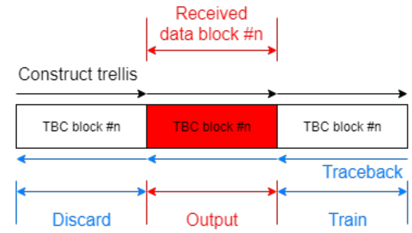


Fig. 2. Low-complexity CVA for FTB decoding. The fixed stopping rule is to use three copies of the received data block and traverses the trellis through them.

unknown state, this techniques suffers form BER degradation, because of lack of protection at the end of tail of the data.

3) Tail-Biting: The encoder, in this technique, uses the last $(K-1)$ bits of the data block to initialize its memory prior to encoding of the data block. This technique ensures that the encoder starts and ends in an identical state for each data block, and attempts to overcome the problem of code rate loss while maintaining equal amount of protection for every bit in the block. However, this goal is achieved at cost of more complex decoding techniques. This is illustrated in the Trellis diagram in Fig. 1 for the $K = 3$ rate 1/2 FTB code.

B. Decoding Techniques

The Viterbi algorithm (VA) [11] is the Maximal Likelihood (ML) decoder for convolutional codes. When applied to any convolutional codes, the branch metrics are calculated for each state at each stage of the Trellis diagram. The branch metric is the distance of the received code from a legitimate code. This metric can be calculated using hamming distance for hard decision VA, or using Euclidean distance for soft decision VA. At each stage, the path with the shortest distance entering each state is kept and other branches are discarded. After $3K - 5K$ stages, the survival paths converge to a single path corresponding to the decoded bit sequence.

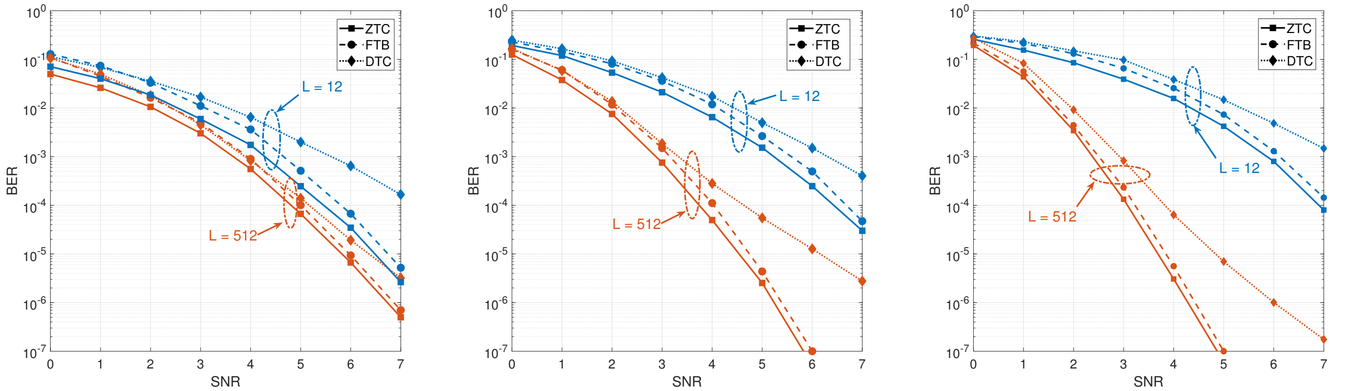
When VA is applied to ZTC codes, the receiver knows, the ML path starts and end at state zero. When it is applied to the DTC codes, the receiver is unaware of the last state of the encoder, therefore it looks for a path that starts at state zero and ends in any state with the minimum path metric. The computational complexity for the DTC and ZTC codes is the same, since VA is applied once through the received code.

However, the situation is different when the VA is applied to FTB codes. The algorithm must find the best path with the constraint that the maximum likelihood path starts and ends in an identical state which can be any one of the 2^{K-1} possible states. It simply appears that a brute force method to decode a FTB code is to run VA once for each possible starting state and, then after constructing the path metrics, check if the ending state with the minimum path metric is equal to the starting state. The VA can start decoding from state zero and work it way up to the last state in an orderly fashion, or take a probabilistic approach [10] by choosing an arbitrary starting state.

Optimal decoding of FTB codes is rather complex. Since the FTB uses the last $(K-1)$ bits of the data block to initialize the encoder memory prior to encoding the data block and discards the $2(K-1)$ output bits corresponding to the initial bits, the start state of the code is unknown *a priori* to the decoder. The tail-biting technique constrains the encoder start and end states to be identical, which requires the decoder to search all possible start states to achieve maximal likelihood (ML) decoding. For codes with a large constraint length, the complexity becomes prohibitive since it requires 2^{K-1} runs of Viterbi decoding. Maximum *A Posteriori* (MAP) decoding of FTB codes is also available achieving superior performance with more computational complexity [12]. Many low-complexity suboptimal algorithms are available in the literature [12]–[17], including the circular Viterbi algorithm (CVA) [14], the Wrap-Around Viterbi Algorithm (WAVA) and Bi-directional VA (BVA) [16], the bounded-distance decoding (BDD) CVA [15], the reliability-output Viterbi algorithm (ROVA) [17], [18], and two-phase algorithms that acquire trellis metrics in a forward and backward manner in search of the ML path [19]. These algorithms are iterative in nature and have less complexity than the brute force ML algorithm.

A low-complexity sub-optimal algorithm for FTB decoding is known as the circular Viterbi algorithm (CVA), where VA traverses around the tail-biting circle more than once [14]–[16]. In this paper, a CVA with a fixed stopping rule was used to decode FTB, similar to the lowest-complexity case in [14]. The received data block is repeated twice, as shown in Fig 2. The decoder starts in an arbitrary state or all states with the same starting metric value, constructs the Trellis by calculating the branch metric. At the block boundary, the VA continues passing the received data block three times. The first time is to find the correct start state for the trellis construction for the second copy of the received block; the second time is to construct the output trellis, and the last time is to perform the correct training so that the traceback begins from the correct state. The decoded bit sequence corresponds to the output in the second copy of the received data block.

This paper evaluates the FTB codes for very short data blocks for underwater acoustic communications by both computer simulation and ocean experiment. Data block lengths were selected as $L = 12, 25, 32, 64$ in comparison to $L = 512$. The FTB codes used the soft-decision CVA decoding



(a) K = 3 (b) K = 6 (c) K = 9

Fig. 3. BER performance in AWGN channels with different constraint lengths: (a) K = 3, (b) K = 6, and (c) K = 9.

algorithm and the ZTC and DTC used the soft-decision VA algorithm.

II. Simulation Results

Computer simulations, in this paper, compare the $K = 3, 6, 9$ rate 1/2 codes, with generator polynomials of [7,5], [74, 64], and [753, 561] respectively, in Additive White Gaussian Noise (AWGN) channels and the results show that the FTB codes with CVA performed similarly with the ZTC codes, while the DTC codes suffered clear degradation in bit error rate (BER). The input data block lengths were $L = 12, 25, 32, 64, 512$. To avoid cluttering in the BER graphs, only the $L = 12$ and the $L = 512$ scenarios are plotted, as shown in Fig 3, where performance of other block lengths laid in between the two groups of curves.

It is clear that DT codes performed worst for all constraint lengths and all data block lengths, because the truncated part of the code suffer from least protection. The larger the constraint length K, the more degradation of the DTC. The FTB and ZTC performed very closely to each other, with the performance of FTB being slightly inferior to the ZTC in all cases. However, the performance gap was only 0.1 dB at BER of 10^{-5}. This demonstrated the advantages of using FTB as a coding technique for short packets.

As the constraint length increases, the performance gap between FTB and ZTC with longer data block lengths was smaller than those with shorter data block length. This is because the traceback length of the decoder is normally $3K$ to $5K$ to achieve good performance. If the data block length is very small, the circular Viterbi algorithm would not have enough length of the received data to cover the traceback length, thus resulting in performance loss.

III. Underwater Experiment Results

An ocean experiment, called SECOMM2017, was conducted in the Atlantic ocean during Oct 4-14, 2017 to test the proposed tail-biting convolutional codes. The experiment site was on the eastern US continental shelf in a 10×10 km^2 area centered at 39.33° N and 72.88° W. Teledyne Benthos ATM-885 MF (16-21 kHz) Subsea Modems were used for transmission and reception. The receiver was anchored about 100 meters above the sea bottom and 40 meters beneath the sea surface. The transmitter was slowly towed on a ship at a speed of a few nauts. The transmit modem was a few meters below the sea surface, and the sea state was calm. The Tx - Rx separation was approximately 1-2 km. meanwhile, another transmitter anchored 0.2 km away from the receiver was sending sporadic interference signals during the experiment.

The structure of the transmitted data packets are shown in Fig. 4, where each data block length frame was consisted of a head linear frequency modulation (LFM) chirp signal labeled as LFMB, followed by an equal number of data blocks coded with ZTC, DTC, and FTB separated by gaps. The number of blocks for $L = 12, 25, 32, 64, 512$ were $N = [80, 40, 32, 16, 2]$ so that the number of randomly generated bits between each chirp is close to 1000 bits. The gaps length was $N_{gap} = 120$ symbols to avoid inter-block interference under highly dispersive UWA channels. The receiver also use the gap before each block to estimate the noise spectrum and compute the SNR. The chirp LFM signals served multiple purposes, such as delimiter of frame start, coarse synchronization, and Doppler shift estimation.

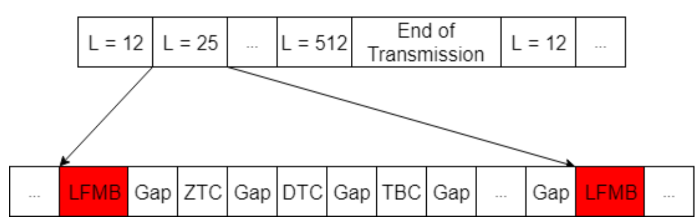


Fig. 4. Structure of Transmitted Data Packets

Different codes and modulation schemes were designed for the experiment. Unfortunately, due to various physical constraints, only two test cases recorded valid data. One was the frame with constraint length $K = 6$ and code rate of

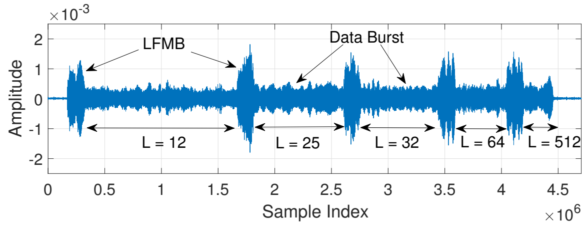


Fig. 5. Received passband signals after a bandpass filter.

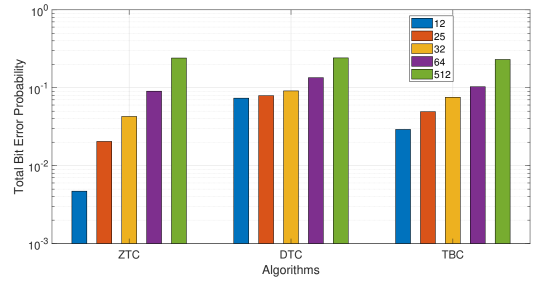
$R = 1/3$, another was $K = 9$ with $R = 1/2$. Both cases used the On-Off Keying (OOK) modulation scheme. The generator polynomials used in these two cases were $G_6 = [75, 53, 47]$, and $G_9 = [753, 561]$, respectively. The data block lengths that are fed to the encoders are the same as simulation. The coded data rate was $R_b = 2560$ bps, and the baseband signal was pulse-shaped using a square-root-raised-cosine (SRRC) filter with a sampling rate of $f_s = 10240$ Hz. Finally, the baseband waveform was modulated by the modem with a carrier frequency of $f_c = 18560$ Hz, and a bandwidth of $BW = 5120$ Hz.

An example of the received passband signal is shown in Fig. 5, after it is passed through a bandpass filter, with cutoff frequencies at $f_c \pm BW$. Received signal is then demodulated non-coherently, by passing through an envelope detector and an SRRC filter. Frame and symbol synchronization was achieved by correlating the LFM with the received signal to find the starting point of the data stream.

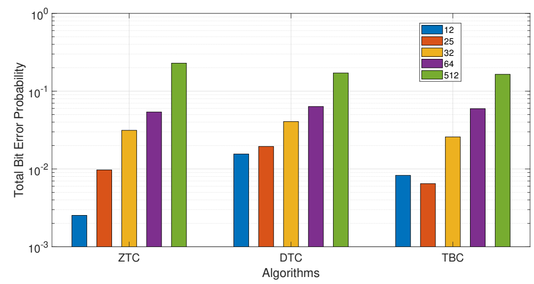
The down-sampled baseband signals were decoded by soft decision Viterbi decoders. Finally the decoded bits are compared to the transmitted data to estimate the BER. The numbers of successfully decoded bits are listed in Table I for the (2,1,9) code and (3,1,6) code. The error probabilities are plotted in Fig. 6. An example of the estimated baseband time-varying fading channel impulse response is shown in Fig. 7. Despite high ISI in received signal, Without a channel equalizer, the short data blocks performed very well in both sub-figures in Fig. 6. With the (3,1,6) codes, the ZTC achieved $2.5 \times 10^{-3}, 1 \times 10^{-2}, 3.5 \times 10^{-2}$ for $L = 12, 25, 32$, respectively. The FTB short data blocks also achieved comparable error rate with $8 \times 10^{-3}, 6.5 \times 10^{-3}, 2.5 \times 10^{-2}$ for $L = 12, 25, 32$, respectively. The DTC clearly suffered from performance loss with the short data blocks with $2 \times 10^{-2}, 2.4 \times 10^{-2}, 4.2 \times 10^{-2}$ for $L = 12, 25, 32$, respectively. For longer data blocks such as $L = 64$ and $L = 512$, the three convolutional coding schemes performed similarly with high errors due to ISI caused by the severe multipath channels. Similar results are shown for the (2,1,9) codes.

TABLE I
TOTAL NUMBER OF PROCESSED BITS FOR EACH SCENARIO

	12	25	32	64	512
$K = 9, R = 1/2$	26880	35000	37888	47104	44032
$K = 6, R = 1/3$	19200	19000	29696	27648	27648



(a)



(b)

Fig. 6. Bit error performance comparison. (a) $K = 9$, rate 1/2 codes, (b) $K = 6$, rate 1/3 codes.

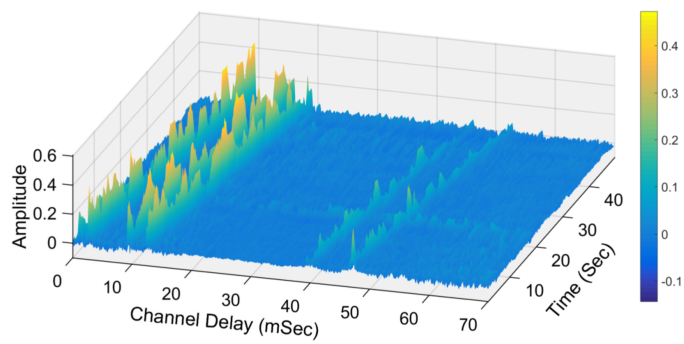


Fig. 7. Bi-time representation of the estimated fading channel impulse response

IV. Conclusion

In this paper, we have evaluated the performance of tail-biting convolutional codes for very short data blocks in underwater acoustic communications. Simulation results show similar performance between ZTC and FTB codes, suggesting that FTB codes for short data blocks can be used in applications where high bandwidth efficiency is required. The lengths of the data blocks are 12, 25, 32, 64 and 512, and the (2,1,9) and (3,1,6) FTB codes are compared with the ZTC and DTC codes. The ocean experimental results show that without channel equalization, the (3,1,6) FTB codes with lengths 25 and 32 perform better than ZTC and DTC codes and the length 12 ZTC perform the best among all codes. These results provide interesting suggestions that although the short data blocks suffered less inter-symbol interference induced by the multipath fading channels the bit error rate performance of the short data block can be further improved with utilizing a simple equalizer.

REFERENCES

[1] M. Chitre, S. Shahabudeen, and M. Stojanovic, "Underwater acoustic communications and networking: Recent advances and future challenges," *Marine Technology Society Journal*, vol. 42, no. 1, pp. 103–116, 2008.

[2] M. Stojanovic and J. Preisig, "Underwater acoustic communication channels: Propagation models and statistical characterization," *IEEE Communications Magazine*, vol. 47, no. 1, pp. 84–89, 2009.

[3] M. Stojanovic, "Recent advances in high rate underwater acoustic communication," *IEEE Journal of Oceanic Engineering*, vol. 21, no. 2, pp. 125–136, 1996.

[4] Y. R. Zheng, J. Wu, and C. Xiao, "Turbo equalization for single-carrier underwater acoustic communications," *IEEE Communications Magazine*, vol. 53, no. 11, pp. 79–87, November 2015.

[5] F. Boccardi, R. W. Heath, A. Lozano, T. L. Marzetta, and P. Popovski, "Five disruptive technology directions for 5g," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 74–80, 2014.

[6] M. C. Domingo, "An overview of the internet of underwater things," *Journal of Network and Computer Applications*, vol. 35, no. 6, pp. 1879 – 1890, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1084804512001646

[7] G. Liva, L. Gaudio, T. Ninacs, and T. Jerkovits, "Code design for short blocks: A survey," *arXiv preprint arXiv:1610.00873*, 2016.

[8] G. C. Clark Jr and J. B. Cain, *Error-correction coding for digital communications*. Springer Science & Business Media, 2013.

[9] S. Lin and D. J. Costello, *Error control coding*. Pearson Education India, 2004.

[10] H. Ma and J. Wolf, "On tail biting convolutional codes," *IEEE Transactions on Communications*, vol. 34, no. 2, pp. 104–111, 1986.

[11] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Inform. Theory*, vol. 13, no. 2, pp. 260 – 269, 1967.

[12] J. B. Anderson and S. M. Hladik, "Tailbiting map decoders," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 2, pp. 297–302, 1998.

[13] Q. Wang and V. K. Bhargava, "An efficient maximum likelihood decoding algorithm for generalized tail biting convolutional codes including quasicyclic codes," *IEEE transactions on communications*, vol. 37, no. 8, pp. 875–879, 1989.

[14] R. V. Cox and C.-E. W. Sundberg, "An efficient adaptive circular viterbi algorithm for decoding generalized tailbiting convolutional codes," *IEEE transactions on vehicular technology*, vol. 43, no. 1, pp. 57–68, 1994.

[15] J. B. Anderson and S. M. Hladik, "An optimal circular viterbi decoder for the bounded distance criterion," *IEEE Transactions on Communications*, vol. 50, no. 11, pp. 1736–1742, 2002.

[16] R. Y. Shao, S. Lin, and M. P. Fossorier, "Two decoding algorithms for tailbiting codes," *IEEE transactions on communications*, vol. 51, no. 10, pp. 1658–1665, 2003.

[17] A. R. Williamson, M. J. Marshall, and R. D. Wesel, "Reliability-output decoding of tail-biting convolutional codes," *IEEE Transactions on Communications*, vol. 62, no. 6, pp. 1768–1778, 2014.

[18] A. R. Raghavan and C. W. Baum, "A reliability output viterbi algorithm with applications to hybrid arq," *IEEE Transactions on Information Theory*, vol. 44, no. 3, pp. 1214–1216, 1998.

[19] Y. Han, T.-Y. Wu, P.-N. Chen, and P. Varshney, "A low-complexity maximum-likelihood decoder for tail-biting convolutional codes," *IEEE Trans. Communications*, vol. PP, no. 99, pp. 1 – 1, 2018.