Ternary LDPC Error Correction for Arrhythmia Classification in Wireless Wearable Electrocardiogram Sensors

Yanfang Liu, *Member, IEEE*, Xiaochen Tang, *Member, IEEE*, David G. M. Mitchell, *Senior Member, IEEE*, and Wei Tang, *Member, IEEE*

***Abstract*— This paper presents a ternary low-density parity-check (LDPC) error correction system for wireless electrocardiogram sensors to improve the accuracy of arrhythmia classification. The classification system is based on ternary Delta-modulated bitstreams and rotation linear kernel support vector machines, which identifies the supraventricular ectopic beat (SVEB) and the ventricular ectopic beat (VEB) over the normal heartbeats. We model errors using a ternary symmetric channel with probability parameter p and construct a variety of ternary LDPC codes with different coding rates by concatenating two-component sub-matrices to form a parity-check matrix with a quasi-cyclic structure that facilitates the hardware design. In particular, a hardware-friendly LDPC encoder circuit is proposed that leverages the highly structured parity-check matrix to perform serial generation of the parity symbols using an accumulator and a look-up table. The encoder circuits are implemented on FPGA and synthesized on ASIC using a 32 nm CMOS process. Simulation results show that the ternary LDPC codes can significantly improve classification accuracy in the presence of errors. For example, with an error probability of up to 21% in the sensor output bitstreams, the classification accuracy remains above 99% with the proposed error correction system.**

***Index Terms*— Ternary error correction, low density parity check, LDPC, delta modulation, wireless wearable sensors, electrocardiogram, arrhythmia classification.**

I. Introduction

WIRELESS wearable biosensors play important roles in health monitoring applications [1]. Such a system is expected to perform biomedical signal sensing, processing, and wireless communication with low power, high accuracy, and low error rate. For example, wearable electrocardiogram (ECG) sensors are expected to continuously monitor the variation of ECG signals in real-time and identify the abnormality, i.e., arrhythmia. Since most wearable systems [2]–[7] are powered by batteries, a common problem in these systems is that the signal processing usually involves complicated machine-learning algorithms, which may cause a high computing overhead and reduces the battery lifetime. For example, [8] proposed a wearable arrhythmia classification system using a deep neural network, which relies on a cloud computing system. However, transmitting the raw ECG signal to the cloud requires a considerable amount of power and the finite battery capacity limits the operation time of the system. The trade-off between the circuit power consumption and real-time signal processing/communication is one of the primary challenges in wearable biomedical sensors.

Recently, pulse-based sensing and direct feature extraction have become promising solutions in wireless biosensors. For example, the signal slope and slope variation can be directly obtained from a first-order Delta modulated bit-stream [9] and a second-order Delta modulated bit-stream [10]. Such an analog-to-feature converter provides advantages of simple circuit structure, lower power consumption, and efficient interface for signal processing. For example, [11] proposed an arrhythmia classification method that uses the ternary bit-stream output, i.e., positive, zero, and negative. In wireless sensors, such a ternary data format would need novel error correction circuits and systems for data communication. Although a few bit errors in the non-weighted bit-stream may not affect the signal processing [12], in biomedical applications, the system usually requires very high accuracy. For example, ECG arrhythmia classification usually achieves an accuracy higher than 95% [11]. Such requirements raise challenges for the ternary bit-stream processing and communication systems. This is because the bit-streams are usually generated using oversampling methods and the sampling rate is much higher than the signal bandwidth and the Nyquist rate, i.e., at least 8 times. The high sampling rate and the non-ideality of the comparator may create bit-errors from the sensor readout circuits. Moreover, errors may occur during wireless communication between the wearable sensor and the remote receiver for telemedicine. Therefore, a ternary bit error detection and correction system is necessary for such a wireless biomedical application.

To perform error-correction, we consider ternary low-density parity-check (LDPC) codes. LDPC codes were first proposed by Gallager in the 1960s and rediscovered in the late 90s [13], [14]. Since then, an intense research effort

Manuscript received March 7, 2021; revised June 16, 2021; accepted July 5, 2021. This work was supported by the United States National Science Foundation under Grant ECCS-1652944, Grant OIA-1757207, Grant HRD-1914635, and Grant ECCS-2015573. This article was recommended by Associate Editor T. Siriburanon. *(Yanfang Liu and Xiaochen Tang contributed equally to this work.) (Corresponding author: Wei Tang.)*

Yanfang Liu, David G. M. Mitchell, and Wei Tang are with the Klipsch School of Electrical and Computer Engineering, New Mexico State University, Las Cruces, NM 88003 USA (e-mail: wtang@nmsu.edu).

Xiaochen Tang is with the Department of Biomedical Engineering, Texas A&M University, College Station, TX 77843 USA.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCSI.2021.3096755.

Digital Object Identifier 10.1109/TCSI.2021.3096755

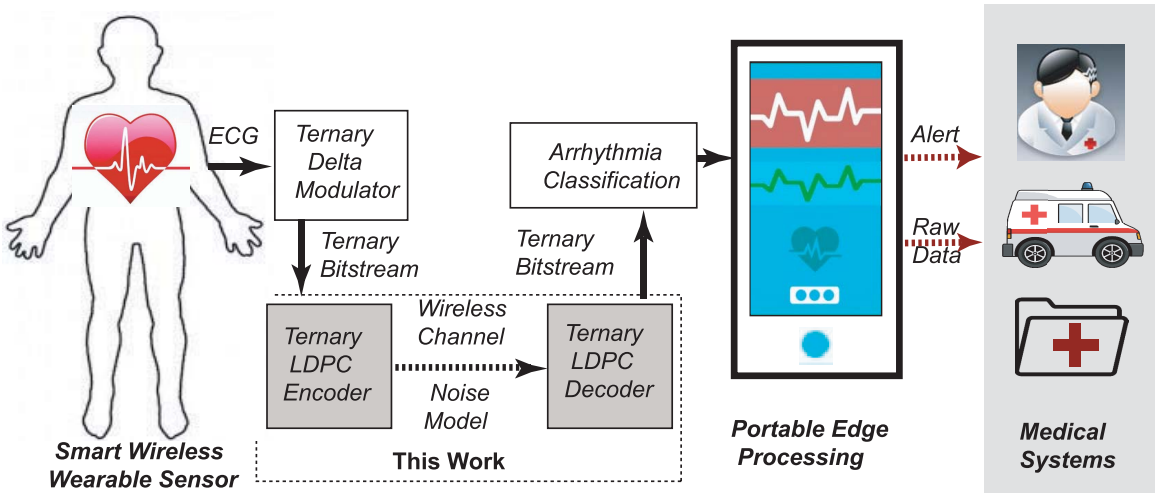


Fig. 1. Proposed ternary LDPC error correction encoder and decoder for a wireless wearable ECG sensor performing arrhythmia classification.

on this class of error-correction codes has been performed, including, among many other aspects, achieving channel capacity [15], [16], weight enumerator and minimum distance analysis [17], structured codes with efficient hardware implementations [18], and finite-length performance analysis [19]. Due to their outstanding decoding performance, LDPC codes have been adopted in multiple communication standards, including 5G new radio, wireless LANs (IEEE 802.11n), WiMax (IEEE 802.16e), digital video broadcasting (DMB-T/H, DVB-S2, DVB-T2, and DVB-C2), and China Multimedia Mobile Broadcasting (CMMB). Asymptotically, LDPC codes are known to achieve capacity, but at finite lengths there remain a number of open issues and challenges. These include designs and mitigation techniques to avoid error floors that are critical for ultra reliable low latency communication (URLLC) in IoT applications and, as addressed in this paper, code and hardware co-design to provide robust error control performance and reduce power consumption on highly power-constrained devices.

Non-binary LDPC codes, constructed over finite fields GF(q) of size $q > 2$, have been shown to outperform comparable binary LDPC-BCs, in particular when the block length is short to moderate [20]. Although non-binary LDPC codes have received significant attention in the literature (see, e.g., [20]–[22]), almost all of these code and hardware designs use an alphabet size that is a power of 2, i.e., $q = 2^m$. Since there have been few applications that require ternary codes, they have received relatively little attention, with few exceptions. For example, different types of ternary channel model were discussed in [23], [24] and constructions of ternary codes have been proposed and analyzed in [25]–[28]. However, these designs do not meet the performance and implementation complexity requirements of wearable biosensors.

The main contribution of this paper is a novel ternary LDPC coding system for error detection and correction of the Delta modulated bit-streams for arrhythmia classification as presented in Fig. 1. We consider a ternary symmetric channel and propose a class of ternary LDPC codes with varying coding rates that achieve different levels of protection and possess a quasi-cyclic structure for efficient implementation. One particular contribution of this work is a low-complexity ternary LDPC encoder circuit that performs serial generation of the parity symbols using only an accumulator and a look-up table. This is achieved by careful design of the code parity-check matrix with hardware-friendly parameter optimization. Here 'hardware-friendly' refers to the saving of the hardware cost in terms of area and power consumption.

The proposed ternary LDPC encoder has been implemented on an FPGA and synthesised using a 32 nm CMOS process. We also outline an efficient sum-product algorithm (SPA) for decoding ternary LDPC codes that is shown to improve the accuracy of arrhythmia classification under communication and/or sensing errors. We evaluate the coding performance in terms of the symbol error rate (SER) and its protection of the classification accuracy under different channel noise levels. The simulation results show that with a coding rate of 1/4, the LDPC coded scheme can maintain a classification accuracy of 99% up to a symbol error probability of 21%. The synthesis results show that the proposed error correction system is hardware-efficient for the implementation of a ternary sensing system, which makes it attractive for low-power wireless wearable sensors.

The remainder of the paper is organized as follows. Section II describes the circuits and systems of the ternary Delta modulator as well as the feature extraction and classification of arrhythmia. Section III presents the channel model and the ternary error correction system, including both the encoding and decoding approaches, as well as the simulation results of the ternary LDPC code. Section IV details the hardware implementation of the proposed ternary LDPC encoder along with the FPGA and ASIC synthesis results. Section V presents the system performance analysis along with an estimation of the power cost. Section VI discusses the advantages and shortcomings of the proposed methods with future research directions. Section VII concludes the paper.

II. Ternary ECG Sensors

In this section, we summarize the necessary background of the overall system of a wearable wireless ECG sensor. The system includes the ternary Delta modulator, feature selection, and classification, which is based on our prior work [11]. In such a system, the wearable ECG sensor amplifies the analog ECG signal and converts the analog signal into digital bit-streams using the first order delta modulator. Then the important features are directly extracted from the bit-stream using a window counting method, which includes the peak and slope information. Then a support vector machine performs classification based on the values of the features extracted from the bitstreams. In the preliminary design, the classification focuses on normal heartbeat, the supraventricular ectopic beat (SVEB), and the ventricular ectopic beat (VEB). The classifier is trained offline using the MIT-BIH database with AAMI standard [29] with a rotation hyperplane, which combines the global and local classifier to achieve patient-specific classification. Here, offline means that the data was trained without manually added error. We first collect data from the user for a certain time and train the local classifier with the collected data. Then we use the proposed method to generate the rotated patient-dependent classifier and upload the classifier to the device. Finally, we can process on-sensor or offline classification depending on the user's choice. For a low power mode, i.e., the on-sensor classification mode, the classification is done without transmitting the data. However, if we need to adjust the classifier or to analyze the ECG data, we have to transmit the Delta modulated ternary bitstream to a remote station for more complicated analysis. Thus error may influence the outcome, and this is the practical scenario of applying the proposed method. Based on the simulation results, a reliable VEB/SVEB classifier is expected to achieve an accuracy higher than 95% without considering bit-errors from the sensor circuits and/or communication channels.

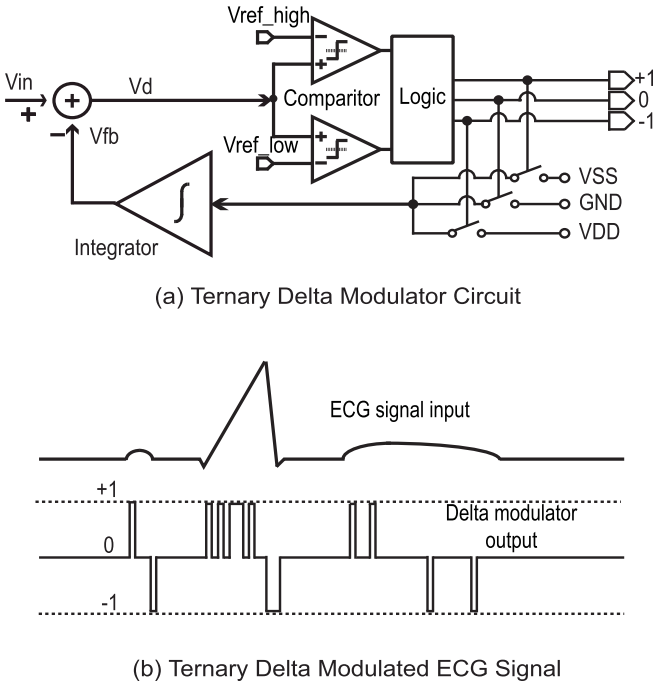


Fig. 2. The signals and systems of the proposed application: The ternary Delta modulator (part (a)) converts ECG signals into ternary bitstreams (part (b)).

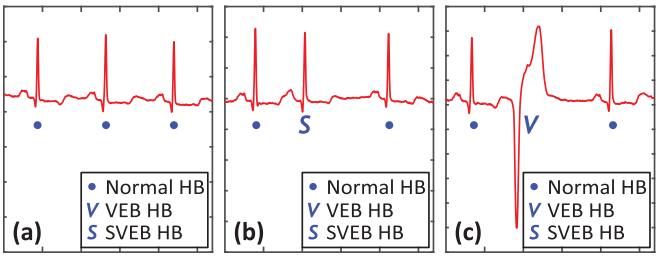


Fig. 3. The goal of the arrhythmia heart beat classification is to identify (a) normal heart beat, (b) SVEB heart beat, and (c) VEB heart beat.

A. Ternary Delta Modulation

A ternary Delta modulator converts an input analog signal into a ternary bit-stream. The schematic of the ternary Delta modulator is shown in Fig. 2(a). The circuit consists of an integrator, two comparators, and a feedback subtractor. The input signal is subtracted by the feedback signals. The feedback signals are determined by the comparator outputs. The subtraction results are then integrated and compared with two pre-defined thresholds. A $+1$ result means that the integrator output is higher than the upper threshold while a -1 result means the integrator output is lower than the lower threshold. In the case that the integrator output is between the window formed by the upper and lower thresholds, the comparison result is set to 0. Thus, $+1$, 0, and -1 are the three symbols in the ternary bit-streams. The ternary bit-stream from the first-order Delta modulator is a pulse density modulation (PDM) of the input signal slope. In such a system, a steep increasing slope of the input signal has a higher pulse density of $+1$ in the output bit-stream or vice versa. If the input signal is stable without variation, the output bit-stream would stay at 0. An example of the ternary Delta modulation is shown in Fig. 2(b). The output bitstream's pulse density is actually proportional to the input signal's slope: the steeper the input signal's slope, the denser the output pulse density. It may look wider in some figures but it is actually the result of consecutive '1' bits or '0' bits. Each pulse has the same width (1 ms), but sometimes some steeper input signal causes consecutive bits.

B. Feature Extraction and Classification

The ternary Delta modulated bit-stream is applied in ECG classification, which including normal heartbeat (sinus rhythm), VEB, and SVEB. VEB is initiated by ectopic focuses in ventricles instead of the sinoatrial node, which results in a bizarre QRS morphology. In contrast, SVEB usually has a normal QRS morphology but the location of the QRS complex is abnormal, which results in a variation of the R-R interval. The difference between normal sinus rhythm, VEB, and SVEB is illustrated in Fig. 3. To perform such classification, the most important features of the ECG waveform are the R-R interval and the morphology of the QRS complex. Since the ECG waveforms have been converted into ternary

bit-streams, we have proposed several features to obtain the key information of the ECG waveform. These features are calculated by counting the number of $+1$ and -1 symbols in the specific windows in the bit-stream and analyzing the distribution of the symbols. In this work, we apply a feature set SkP-32 [11] which includes the number of $+1/-1$ symbols, the skewness weight, the R wave polarity, and the R-R interval.

The classification is based on the feature values obtained from feature extraction, which were tested using the MIT-BIH database. The detailed feature selection analysis was reported in [11]. The features are calculated in a 215ms window around the R peak. Considering the hardware cost and computing overhead, we choose a simple linear kernel support vector machine as the classifier since it has a simple structure. The classifier is evaluated by its accuracy, sensitivity, and specificity. The primary challenge of improving the classification performance of a linear kernel SVM is to train the model and find the parameter of the hyperplane. In our platform, we proposed a rotation hyperplane that combines a global classifier, which was generated using all the data in the database, and a local classifier, which was optimized from data for a specific record. The combination of the two hyperplane balances the generalization and specification while providing reasonable accuracy.

III. Ternary LDPC Error Correction

Wireless communication is one of the key components in low-power wearable biosensors. The primary reason is that the capability of on-chip data processing is limited due to power constraints. Therefore, in most current devices, the raw data is required to be transmitted from the wearable sensor to a remote station for advanced processing [8] or checked by a human expert, i.e., a cardiologist [36]. Unfortunately, the error introduced during communication may affect the decision-making process. Indeed, a few critical bits in error may lead to wrong conclusion and/or treatment, which is not acceptable in medical applications. Although wearable biosensors have been actively studied [37], [38], most of these works did not focus on error correction. In particular, for sensing and processing with ternary data formats, such as [39], there are no references considering error correction. Thus, the effects of error and related error correction in communications become a necessary research topic for wearable biosensors, especially for the emerging ternary sensors. The proposed work focuses on the performance of LDPC error correction of ternary data from wireless transmission as well as its effect on the classification algorithm. The accuracy improvement from the sensor device and classification algorithms, interesting in their own right, goes beyond the scope of this paper.

A. Channel Model and Effect on Classification Accuracy

We consider the case where the ternary ECG symbols can be corrupted by noise, which we will model by the ternary symmetric channel with parameter (probability) $0 \leq p \leq 1$, as shown in Fig. 4. In the remainder of the paper we represent the "-1" symbol by a "2", such that the alphabet is GF(3) $= \{0, 1, 2\}$. The receiver correctly receives the transmitted symbol with probability $1 - p$, but an error occurs to corrupt the symbol to one of the other two symbols with transition probability $p/2$, respectively.[1] The operations on GF(3) are shown in Table I.

TABLE I
Modulo-3 Addition and Multiplication

+	0	1	2	×	0	1	2
0	0	1	2	0	0	0	0
1	1	2	0	1	0	1	2
2	2	0	1	2	0	2	1

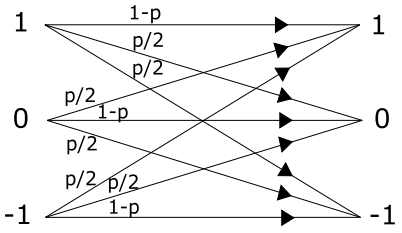


Fig. 4. Channel model: ternary symmetric channel with parameter $0 \leq p \leq 1$.

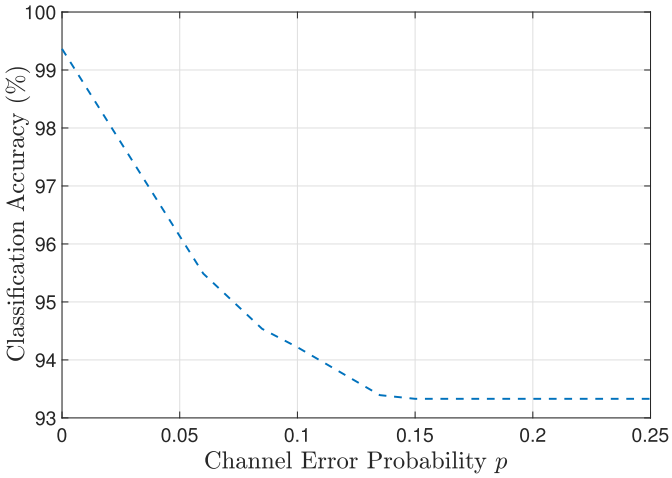


Fig. 5. Classification accuracy as a function of channel error probability p.

Simulation results of classification corresponding to the proposed ternary symmetric channel as a function of p are shown as the dashed curve in Fig. 5. It is observed that the classification accuracy decreases sharply with p until approximately $p = 0.15$, at which point the accuracy levels off at approximately 93.3%. The performance is measured in terms of a binary classification, which is SVEB (S) versus the other four types of heartbeats: Normal (N), VEB (V), Fusion Beats (F), and unknown beats (Q), i.e., the S beat versus the other four classes defined in the AAMI standard. We proposed two rotated patient-dependent SVM classifiers for two binary classifications, one is (S) v.s. (N, V, F, Q), and the other is (V) v.s. (N, S, F, Q). For explanatory demonstration, we just presented the result of the first classifier which is

[1] In this work, we restrict our attention to the symmetric case with uniform error probabilities to facilitate analysis; however, the channel model and decoder could be generalized to accommodate non-uniform probabilities.

(S) v.s. (N, V, F, Q). Crucially, we note that the accuracy drops below 96% when p is larger than 0.05, indicating the necessity of an error correction code to reduce the noise influence and maintain satisfactory classification accuracy.

B. Ternary LDPC Code Construction and Encoding/Decoding Algorithms

In this section, we present the proposed ternary LDPC code construction and decoding algorithm. We also confirm the effectiveness of the approach via simulation results.

1) Code Construction: To design a ternary LDPC code, we adapt the method proposed in [30], where the parity-check matrix of a non-binary low-density parity-check (NB-LDPC) code is constructed by concatenating two component submatrices as $\mathbf{H}_{\mathrm{NB}} = [\mathbf{H}_1\ \mathbf{H}_2]$, where

$$\mathbf{H}_1 = \begin{bmatrix} \delta_{0,0}\mathbf{P}_s^{b_{0,0}} & \delta_{0,1}\mathbf{P}_s^{b_{0,1}} & \dots & \delta_{0,m-1}\mathbf{P}_s^{b_{0,m-1}} \\ \delta_{1,0}\mathbf{P}_s^{b_{1,0}} & \delta_{1,1}\mathbf{P}_s^{b_{1,1}} & \dots & \delta_{1,m-1}\mathbf{P}_s^{b_{1,m-1}} \\ \vdots & \vdots & & \vdots \\ \delta_{k-1,0}\mathbf{P}_s^{b_{k-1,0}} & \delta_{k-1,1}\mathbf{P}_s^{b_{k-1,1}} & \dots & \delta_{k-1,m-1}\mathbf{P}_s^{b_{k-1,m-1}} \end{bmatrix},$$

is of size $K \times M$, with $K = ks$, $M = ms$, coefficients $\delta_{j,i} \in \mathrm{GF}(q)$, and submatrices $\mathbf{P}_s^{b_{j,i}}$ are $s \times s$ circulant permutation matrices constructed by circulantly shifting the rows of the $s \times s$ identity matrix $\mathbf{I}_s$ by $b_{j,i} \in \{1, 2, \dots, s\}$ positions to the left. Moreover,

$$\mathbf{H}_2 = \begin{bmatrix} \mathbf{I}_s & \mathbf{0}_s & \dots & \dots & \gamma_{k-1}\hat{\mathbf{P}}_s^1 \\ \gamma_0\mathbf{I}_s & \mathbf{I}_s & \mathbf{0}_s & \dots & \mathbf{0}_s \\ \mathbf{0}_s & \gamma_1\mathbf{I}_s & \mathbf{I}_s & \dots & \mathbf{0}_s \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{0}_s & \dots & \mathbf{0}_s & \gamma_{k-2}\mathbf{I}_s & \mathbf{I}_s \end{bmatrix}$$

is a $K \times K$ matrix, where $\gamma_i \in \mathrm{GF}(q)\backslash\{0\}$ and $\hat{\mathbf{P}}_s^1$ is constructed by deleting the right corner "1" in the circulant permutation matrix $\mathbf{P}_s^1$, i.e.,

$$\hat{\mathbf{P}}_s^1 = \begin{bmatrix} 0 & 0 & \dots & 0 & 0 \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & \dots & 0 & 1 & 0 \end{bmatrix}_{s\times s}.$$

This modification will allow efficient encoding, as explained below. The code length is $M + K$ and coding rate $R \geq M/(M + K)$.

This method to design the parity-check matrix of NB-LDPC via circulant permutation matrices promises parallel decoding, which can significantly increase the throughput and speed up the decoding process [30]. Moreover, the memory used to store the parity check matrix in the hardware design is decreased by using such a structure, which is desirable in practice.

2) Encoding Algorithms: We consider two approaches to form encode the ternary information symbols. The first method derives the generator matrix $\mathbf{G}$ used to encode the NB-LDPC code in a straightforward way by Gaussian elimination. First, we transform $\mathbf{H}_{\mathrm{NB}}$ to the following form via Gaussian elimination

$$\mathbf{H} = \begin{bmatrix} t_{0,0} & t_{0,1} & \dots & t_{0,M-1} & 1 & 0 & \dots & 0 \\ t_{1,0} & t_{1,1} & \dots & t_{1,M-1} & 0 & 1 & \dots & 0 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ t_{K-1,0} & t_{K-1,1} & \dots & t_{K-1,M-1} & 0 & 0 & \dots & 1 \end{bmatrix} = [\mathbf{T} \mid \mathbf{I}_K],$$

where the elements $t_{i,j} \in \mathrm{GF}(q)$, the size of $\mathbf{T}$ is $K \times M$, and $q = 3$ for ternary LDPC codes. Hence, $\mathbf{G}$ is of the form $\mathbf{G} = [\mathbf{I}_M \mid \mathbf{T}']$, where the size of $\mathbf{T}'$ is $M \times K$. The M ternary information symbols $\mathbf{u} = [u_0, u_1, \dots, u_{M-1}]$ are encoded to codeword $\mathbf{v}$ of length $M + K$ as $\mathbf{v} = \mathbf{uG}$. The generator matrix $\mathbf{G}$ generated by Gaussian elimination can be implemented in full parallel which speeds up the encoding process [31]. However, since the submatrix $\mathbf{T}'$ is dense and unstructured it is not favorable for implementation when compared to the second approach below because 1) a memory module will be required to store a large number of locations of symbols that are involved in computations, and 2) this method requires a large number of operations to encode and a significantly higher layout complexity [32].

To adapt the encoder design to a wearable device that has severe power constraints, we propose a way to encode the ternary information that makes use of the highly structured parity-check matrix $\mathbf{H}_{\mathrm{NB}}$. This results in a low-complexity encoder that can be implemented in small area (hardware considerations will be discussed in Section IV).

Assume a systematic codeword from the ternary LDPC code is $\mathbf{v} = [u_0, u_1, \dots, u_{M-1}, p_0, p_1, \dots, p_{K-1}]$, where the first M symbols are the ternary information symbols and such that $\mathbf{v}\mathbf{H}_{\mathrm{NB}}^T = \mathbf{0}$. By observing the special structure of $\hat{\mathbf{P}}_s^1$, we notice all elements in the first row are 0s, which means all the elements in the first row of the $\mathbf{H}_2$ are 0s except the first element. According to the parity-check rule $\mathbf{v}\mathbf{H}_{\mathrm{NB}}^T = \mathbf{0}$, the first row of $\mathbf{H}_{\mathrm{NB}}$ implies

$$\delta_{0,0}u_{s-b_{0,0}} + \delta_{0,1}u_{2s-b_{0,1}} + \cdots + \delta_{0,m-1}u_{ks-b_{0,m-1}} + p_0 \equiv 0 \pmod 3. \quad (1)$$

Hence, p_0 can be generated by (1). We then move forward to the $(s + 1)$th row of $\mathbf{H}_{\mathrm{NB}}$, where it follows that p_s can be determined (after obtaining p_0) from

$$\delta_{1,0}u_{s-b_{1,0}} + \delta_{1,1}u_{2s-b_{1,1}} + \cdots + \delta_{1,m-1}u_{ks-b_{1,m-1}} + \gamma_0 p_0 + p_s \equiv 0 \pmod 3. \quad (2)$$

By repeating this procedure, p_{is} can be serially generated from row $is + 1$ of $\mathbf{H}_{\mathrm{NB}}$, $i \in [0, k - 1]$. To compute p_1, we observe from the second row of $\mathbf{H}_{\mathrm{NB}}$ that

$$\delta_{0,0}u_1 + \delta_{0,1}u_{2s-b_{0,1}+1} + \cdots + \delta_{0,m-1}u_{ks-b_{0,m-1}+1} + p_1 + \gamma_{k-1}p_{(k-1)s} \equiv 0 \pmod 3. \quad (3)$$

Following a similar encoding procedure as described above, p_{is+1} is derived from row $is + 2$ of $\mathbf{H}_{\mathrm{NB}}$, $i \in [0, k - 1]$, and so on in blocks of k symbols until the entire codeword $\mathbf{v}$ is generated. This encoding algorithm is later summarized in Algorithm 1.

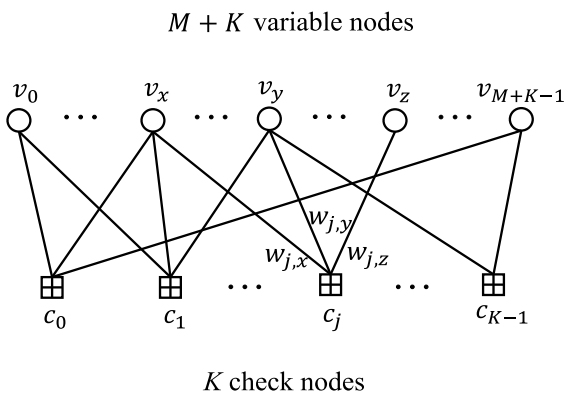


Fig. 6. Tanner graph of a NB-LDPC code.

As mentioned above, the circulant structure of the parity-check matrix allows this procedure to be implemented efficiently using a small area, as will be described in Section IV, but must be computed serially.[2]

3) Decoding via the Sum-Product Algorithm: To decode the designed ternary LDPC codes, we propose to use the sum-product algorithm (SPA), which iteratively passes messages (symbol likelihoods) back and forth on a factor graph representation of the sparse parity check matrix [33]. The decoding process of LDPC codes proceeds iteratively, updating the check nodes and variable nodes until convergence to a valid codeword, which corresponding to a decoding success, or failure to converge by some preset maximum number of iterations, which corresponds to a decoding failure. As a class of NB-LDPC codes over GF(3), ternary LDPC codes can be represented by Tanner graph as shown in Fig. 6, where the circles represent the variable nodes, which correspond to the columns in $\mathbf{H}_{\text{NB}}$ (code symbols), the squares represent the check nodes which correspond to the rows in $\mathbf{H}_{\text{NB}}$ (parity-check equations), and the lines represent the non-zero elements in $\mathbf{H}_{\text{NB}}$. Here, each edge connecting check node c_j to variable node v_x has a weight $w_{j,x}$ which corresponds to the entry in $\mathbf{H}_{\text{NB}}$ from GF$(q)\backslash\{0\}$. For example, in Fig. 6, we label check node c_j and its three connected edges with the corresponding weights $w_{j,x}$, $w_{j,y}$, and $w_{j,z}$.

The NB-LDPC decoding algorithm is executed by iteratively updating the variable nodes and check nodes. For the ternary SPA, the message from variable node v_x to check node c_j at iteration ℓ is a vector of probabilities $\mathbf{V}_{x,j}^{(\ell)} = (p_{v_{x,j,0}}^{(\ell)}, p_{v_{x,j,1}}^{(\ell)}, p_{v_{x,j,2}}^{(\ell)})$ (a probability mass function) that corresponds to the likelihood that v_x is a 0, 1, or 2, respectively, where $p_{v_{x,j,0}}^{(\ell)} + p_{v_{x,j,1}}^{(\ell)} + p_{v_{x,j,2}}^{(\ell)} = 1$. Similarly, the message from check node c_j to variable node v_x at iteration ℓ is also a vector of probabilities $\mathbf{C}_{j,x}^{(\ell)} = (p_{c_{j,x,0}}^{(\ell)}, p_{c_{j,x,1}}^{(\ell)}, p_{c_{j,x,2}}^{(\ell)})$ (a probability mass function) that corresponds to the likelihood that v_x is a 0, 1, or 2, respectively, where $p_{c_{j,x,0}}^{(\ell)} + p_{c_{j,x,1}}^{(\ell)} + p_{c_{j,x,2}}^{(\ell)} = 1$. Messages $\mathbf{V}_{z,j}^{(0)} = (p_{v_{x,j,0}}^{(0)}, p_{v_{x,j,1}}^{(0)}, p_{v_{x,j,2}}^{(0)})$ are initialized from the channel using Bayes' rule and assuming the inputs are equally likely.

To illustrate, we give a brief example of the update rules for a variable node of degree 3 and a check node of degree 3, but the approach suitably generalizes to arbitrary degrees with appropriate modifications.[3] In this case, the check node c_j has three connected variable nodes v_x, v_y, and v_z with corresponding weights $w_{j,x}$, $w_{j,y}$, and $w_{j,z}$.[4] According to parity-check rule, $w_{j,x}v_x + w_{j,y}v_y + w_{j,z}v_z = 0$. The probability mass function (message) to be passed to each ternary variable node can now be computed. For example, the message from c_j to v_z (illustrated in Fig. 7(a)) when $w_{j,z} = 1$ at iteration $\ell > 0$ is

$$\mathbf{C}_{j,z}^{(\ell)} = (p_{c_{j,z,0}}^{(\ell)}, p_{c_{j,z,1}}^{(\ell)}, p_{c_{j,z,2}}^{(\ell)}),$$

where

$$\begin{aligned}
p_{c_{j,z,0}}^{(\ell)} &= \sum_{\substack{a,b\in\text{GF(3)} \\ w_{j,x}a+w_{j,y}b\equiv 0 \pmod 3}} p_{v_{x,j,a}}^{(\ell-1)} p_{v_{y,j,b}}^{(\ell-1)}, \\
p_{c_{j,z,1}}^{(\ell)} &= \sum_{\substack{a,b\in\text{GF(3)} \\ w_{j,x}a+w_{j,y}b\equiv 2 \pmod 3}} p_{v_{x,j,a}}^{(\ell-1)} p_{v_{y,j,b}}^{(\ell-1)}, \\
p_{c_{j,z,2}}^{(\ell)} &= \sum_{\substack{a,b\in\text{GF(3)} \\ w_{j,x}a+w_{j,y}b\equiv 1 \pmod 3}} p_{v_{x,j,a}}^{(\ell-1)} p_{v_{y,j,b}}^{(\ell-1)}.
\end{aligned} \tag{4}$$

The other case when $w_{j,z} = 2$ is similar, where the modulo condition for $p_{c_{j,z,1}}$ and $p_{c_{j,z,2}}$ are instead congruent to 1 and 2, respectively. The other outgoing messages $\mathbf{C}_{j,x}^{(\ell)}$ and $\mathbf{C}_{j,y}^{(\ell)}$ would be computed similarly.

Suppose variable node v_z is connected to check nodes c_j, c_k, and c_l. Then the (normalized) message vector to pass from v_z to c_j at iteration $\ell > 0$ (illustrated in Fig. 7(b)) is

$$\mathbf{V}_{z,j}^{(\ell)} = (p_{v_{z,j,0}}^{(\ell)}, p_{v_{z,j,1}}^{(\ell)}, p_{v_{z,j,2}}^{(\ell)}),$$

where

$$\begin{aligned}
p_{v_{z,j,0}}^{(\ell)} &= p_{v_{z,j,0}}^{(0)} p_{c_{k,z,0}}^{(\ell)} p_{c_{l,z,0}}^{(\ell)} / \Lambda, \\
p_{v_{z,j,1}}^{(\ell)} &= p_{v_{z,j,1}}^{(0)} p_{c_{k,z,1}}^{(\ell)} p_{c_{l,z,1}}^{(\ell)} / \Lambda, \\
p_{v_{z,j,2}}^{(\ell)} &= p_{v_{z,j,2}}^{(0)} p_{c_{k,z,2}}^{(\ell)} p_{c_{l,z,2}}^{(\ell)} / \Lambda,
\end{aligned}$$

and

$$\begin{aligned}
\Lambda = p_{v_{z,j,0}}^{(0)} p_{c_{k,z,0}}^{(\ell)} p_{c_{l,z,0}}^{(\ell)} + p_{v_{z,j,1}}^{(0)} p_{c_{k,z,1}}^{(\ell)} p_{c_{l,z,1}}^{(\ell)} \\
+ p_{v_{z,j,2}}^{(0)} p_{c_{k,z,2}}^{(\ell)} p_{c_{l,z,2}}^{(\ell)}.
\end{aligned} \tag{5}$$

The other outgoing messages $\mathbf{V}_{z,j}^{(\ell)}$ and $\mathbf{V}_{z,k}^{(\ell)}$ would be computed similarly.

To simplify the computation and avoid numerical issues, in our implementation we compute (4) and (5) on the likelihood domain and log-likelihood domains, respectively.[5]

[2]For a situation that requires a high speed encoder design, rather than a low-power small-area design, the Gaussian elimination method that is suitable for a full-parallel design may be preferred.

[3]The degree of a node is the number of edges connected to this node in the Tanner graph.

[4]We assume without loss of generality that no weight is zero since that would eliminate an edge and reduce the degree of the check node.

[5]Details of probability domains are omitted. See, e.g., [33] for details.

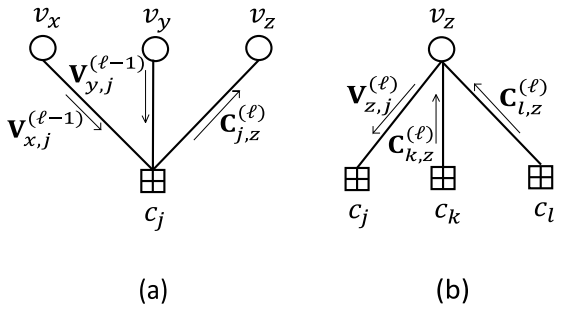


Fig. 7. Illustration of the message updates of SPA for ternary LDPC Tanner graphs: (a) computing check to variable messages and (b) computing variable to check messages.

4) Simulation Results: To compare the decoding performance of ternary LDPC codes, we design three different codes with rates $R = 1/2$, $1/3$, and $1/4$. The codes are all constructed for $M = 500$ ternary information symbols, corresponding to $m = 5$ and $s = 100$. The number of symbols was selected based on the sampling rate of 1kS/sec of the ECG sensor so that each packet could include a full QRS complex, which must be protected for accurate classification. Parameter k was selected as 5, 10 and 15, giving $K = 500, 1000$, and 1500, respectively. The circulant shift parameters $b_{j,i}$, are all randomly selected. In our design, to minimize complexity and latency, all γ_l in $\mathbf{H}_2$ are set to be 1 and only three $\delta_{j,i}$ are non-zero in each row block of $\mathbf{H}_1$ (group of s rows according to j). Moreover, to allow hardware re-use in the encoding computation we select the three non-zero $\delta_{j,i}$ values in a row group j to be $1, 2, 1$ in that order. Fixing the pattern for each j minimizes the hardware cost since it results in the same parity-check computation for each symbol, as described in Section IV, and is found to give an acceptable decoding performance, as shown below. We note that following this $\delta_{j,i}$ and γ_l structure results in an ensemble of ternary LDPC codes that all possess efficient hardware implementation.

The simulation results are shown in Fig. 8 for randomly drawn codes from the hardware-friendly ensemble. [6] When the code rate R is low, the decoder can correct more errors: for example, when the target SER is 10^{-3}, the rate $R = 1/4$ code has corresponding channel parameter $p \approx 0.21$ (21% of channel error), the rate $R = 1/3$ code has $p \approx 0.175$, and the rate $R = 1/2$ code has $p \approx 0.075$. In a practical design, there is trade-off between the error correction capability and the redundancy. Although in this paper we restrict our attention to randomly constructed codes to give an idea of the average performance, and select a particular $\delta_{j,i}$ and γ_l structure to minimize hardware cost, interesting future work involves optimization of the code structure to improve the performance and implementation complexity trade-off. Nevertheless, the purpose of the random code as presented is proof-of-concept that a ternary LDPC coding system can be designed and implemented in an efficient way for such a power-restricted device. Improved decoding performance can be expected for optimized parameters.

[6]To the best of our knowledge, there does not exist a ternary code that can meet the requirements of our system for comparison, so we select a code randomly from the ensemble to demonstrate average performance.

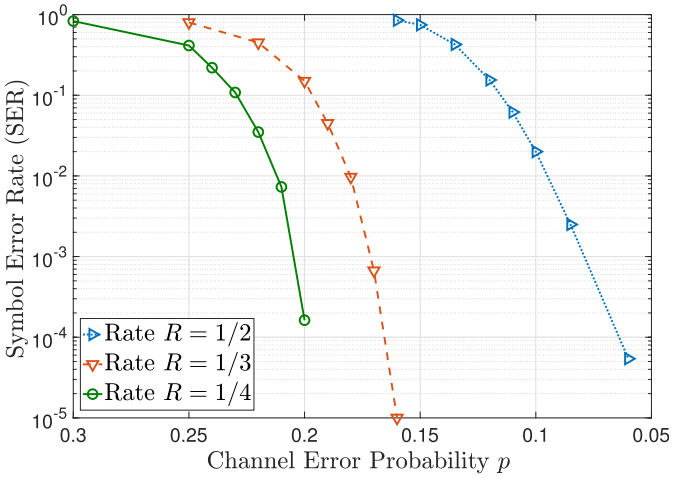


Fig. 8. Decoding performance of the constructed NB-LDPC codes versus decreasing channel error probability p.

IV. Hardware Design and Implementation

The primary considerations in the hardware implementation of the ternary LDPC encoder for wearable sensors are the power cost and hardware complexity. Since the target signal (ECG) is sampled at a relatively low frequency (1 kSample/second), the encoder can be designed with one computation circuit and serial generation of output parity symbols to save system power at the cost of a longer processing time, which is acceptable in our application. For this reason, our approach avoids using parallel encoding via a dense generator matrix $\mathbf{G}$ derived by Gaussian elimination, which requires a large number of operations and memory access. Instead, the encoder generates the parity symbols directly from the highly structured parity-check matrix of the ternary LDPC code in a serial fashion. The codeword is formed by concatenating the information symbols and the generated parity symbols. For example, with a 1/2 coding rate, the encoder generates 500 parity symbols from a group of 500 information symbols to form a 1000 symbol codeword. Such an encoder can be designed using a simple accumulator and a parity check module made of a look-up-table for operations on GF(3). This method saves the computing resource and the memory accessing cost of the hardware.

The block diagram of ternary LDPC encoder is shown in Fig. 9. The *Address Generator* #0 generates addresses to store the input serial ECG symbols $[u_0, u_1, \ldots, u_{M-1}]$ in the random access memory (*RAM*).[7] After the input data has been stored, the *Address Generator* #1 generates the appropriate addresses of the message symbols u_i involved in the current computation (using shift registers according to the parameters $b_{j,i}$ and s) and the data is read from RAM. This procedure

[7]Please note that Vivado does not provide verilog file of RAM. To perform FPGA synthesis, we programmed *RAM* with shift-registers. Firstly, all Sensor Inputs are stored in *RAM*. We then arrange the position of bits in shift-registers according to *Address Generator* #0.

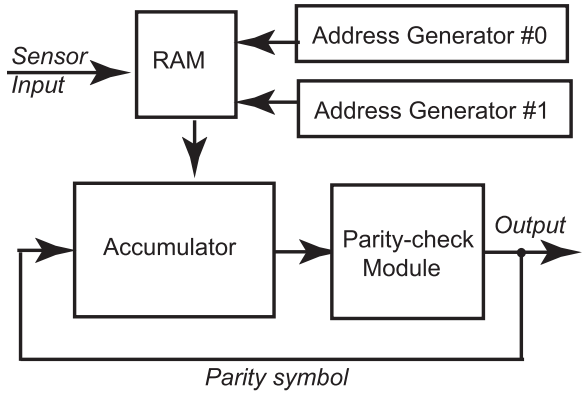


Fig. 9. Block diagram of the ternary LDPC encoder circuit.

will be repeated every time instant according to the current parity-check symbol computation. The data at a given time unit is sent to *Accumulator*, which performs the ternary logic operations (according to the $\delta_{j,i}$ and γ_i parameters in a row block (group of s rows) of $\mathbf{H}_{\mathrm{NB}}$, which will be explained below. The generated result that is required to determine the current parity-check symbol is sent to the *Parity-check Module* which outputs the parity-check symbol according to operation rules on GF(3). This parity-check symbol is one symbol of the *Output* and is also fed back to the *Accumulator* to generate the next parity-check symbol. The ternary LDPC encoder stops when all parity-check symbols are generated. This encoding algorithm is summarized in Algorithm 1.

For example, when computing the initial parity-check symbol p_0, the input to the accumulator is $[u_{s-b_{0,0}}, u_{2s-b_{0,1}}, \ldots, u_{ks-b_{0,m-1}}]$ and output is the sum of symbols in (1) excluding the p_0 term. When computing p_s using (2), input to the accumulator is $[u_{s-b_{1,0}}, u_{2s-b_{1,1}}, \ldots, u_{ks-b_{1,m-1}}]$, as well as p_0 from the *Parity-check Module*, and output is the sum of (2) excluding the p_s term, and so on. In our design, there are only three non-zero $\delta_{j,i}$ values in any equation, say $\delta_1^{(t)}, \delta_2^{(t)}$, and $\delta_3^{(t)}$, corresponding to three message symbols $u_1^{(t)}, u_2^{(t)}$, and $u_3^{(t)}$ at time t. Moreover, since the δ's were selected to be $1, 2, 1$, in that order, and each $\gamma^{(t)}$ was selected as 1, the accumulator need only output

$$\begin{aligned}\delta_1^{(t)} \times u_1^{(t)} + \delta_2^{(t)} \times u_2^{(t)} + \delta_3^{(t)} \times u_3^{(t)} + \gamma^{(t)} \times p^{(t-1)} \\ = 1 \times u_1^{(t)} + 2 \times u_2^{(t)} + 1 \times u_3^{(t)} + 1 \times p^{(t-1)} \quad (6)\end{aligned}$$

where $p^{(t-1)}$ denotes the parity-check symbol output at the last time instant (initialized as 0 for the first computation). Note that a general design would have to allow for up to m message symbols per unit time and different values for $\delta_{j,i}$ and γ_l with corresponding logic for the computations in each row block (value of j). Our design significantly reduces the implementation complexity and has good performance (as shown in Section III-B.4).

In order to implement the ternary LDPC encoder using binary hardware, the ternary information symbols and ternary parity check symbols are encoded into two-bit binary data strings. Specifically, binary strings "00", "01", and "10" represent ternary symbols "0", "1", and "2" in GF(3), respectively.

Algorithm 1 Encoding Process for Ternary ECG Data

Input: M ternary symbols $\mathbf{u}$ from ECG signal, $\mathbf{H}_{\mathrm{NB}}$
Output: Encoded codeword $\mathbf{v}$

1: Initialize *RAM* with $\mathbf{u}$
2: **for** parity-check block $j = 0, \ldots, s-1$ **do**
3: **for** $i = 0, \ldots, k-1$ **do**
4: Generate the locations of variable nodes (non-zero entries) in row $is + j + 1$ of $\mathbf{H}_{\mathrm{NB}}$
5: Read data from *RAM* according to the generated locations of message symbols
6: Determine symbols used for current parity-check computation (*Accumulator*)
7: Compute p_{is+j} (*Parity-check Module*) then output and feedback for next computation
8: **end for**
9: **end for**

The *Accumulator* adds the input binary strings (with no modulo arithmetic) to form a binary input for the *Parity-check Module*. For example, the sum of binary strings "10" and "01" results in binary string "11". Our design uses a shift-register to compute the multiplication between the coefficient $\delta_{j,i}$ and the information symbol before addition. In particular, in the case that the coefficient δ is 2, the binary information string is shifted cyclically left by one bit, while if the coefficient δ is 1, the information string keeps its original (binary) value. The multiplication results are extended to 4 bits to avoid overflow.[8]

The parity symbols can now be calculated. Specifically, the parity symbol $p^{(t)}$ at time t is determined by the three current information symbols, the previously computed parity symbol, and the appropriate coefficients δ and γ. The parity-check module computes the parity symbol using the 4-bit binary data output result of (6) from the *Accumulator*, which is the binary string according to the sum of the multiplication results between the coefficient δ and the information bits, plus the prior parity symbol. In our design, the coefficients δ are selected as [1, 2, 1] and the information bits are either "00", "01", or "10". Thus, the accumulation result of (6) has a maximum possible value of "1010" (decimal 10) when the three information bits and the previous parity symbol are all "10". Consequently, the range of the parity check module input (the result of (6)) is from "0000" to "1010". A look-up-table is then created to generate the corresponding ternary parity symbol based on the 4-bit binary input using Table II, which is computed based on Table I. Note that based on GF(3), a binary value higher than 3 is equivalent to its modulus of 3. For example, a 4-bit binary value "0101" means symbol "2" in GF(3) and its corresponding parity is determined as "1" since in GF(3) "2"+"1"="0" (Table I). Therefore, the output binary version is "01", which is equivalent to "1" in GF(3).

The operation of the encoder is presented as the timing diagram of Fig. 10. First, the write-enable signal (WEA) enables the writing process from the ternary Delta modulator to *RAM*.

[8]With different number of non-zero δ's involved in (6), different extension size should be considered.

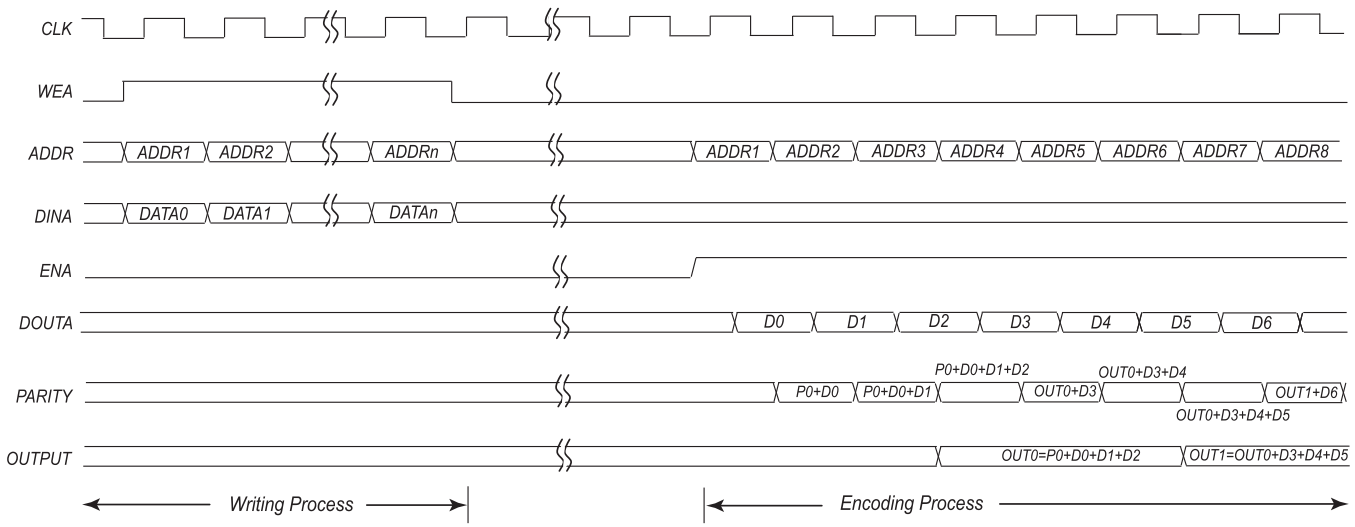


Fig. 10. Timing schedule of the ternary LDPC encoder.

TABLE II
PARITY-CHECK MODULE LOOK-UP TABLE

Binary input	GF(3) symbol	GF(3) parity	Binary output
0000	0	0	00
0001	1	2	10
0010	2	1	01
0011	0	0	00
0100	1	2	10
0101	2	1	01
0110	0	0	00
0111	1	2	10
1000	2	1	01
1001	0	0	00
1010	1	2	10

For each symbol group, a total number of 500 symbols from the data input pin DIN are written into the RAM according to the address $ADDR$ generated from *Address Generator* #0. Then in the encoding process, the encoder-enable signal ENA initiates the encoding process. The data is read from *RAM* through the data output pin $DOUTA$ according to the address $ADDR$ from *Address Generator* #1. The *Parity-check Module* is then calculating the parity symbols $PARITY$ based on the accumulator output and the parity symbols from the prior clock cycle. The generated parity symbol should be fed back to the accumulator to generate the next ternary parity symbol. The $OUTPUT$ signal is calculated using three prior $DOUTA$ bits and the one prior $PARITY$ bit, which updates every three clock cycles. Note that the parity-check sequence output is not in the numbering order. For example, the first calculated parity symbol is p_0, then the next calculated parity symbol is p_s instead of p_1, as described in Algorithm 1. The data could be reordered at the receiver, which typically has less power constraints.

The proposed encoder has been implemented on an FPGA and synthesised using a standard 32 nm CMOS technology. Since the Vivado tool does not provide a verilog file for the RAM, we used registers as memory blocks in order to evaluate the hardware cost for FPGA synthesis. To calculate

TABLE III
FPGA AND ASIC SYNTHESIS RESULTS OF THE ENCODER

		Encoder w/o SRAM	Encoder with SRAM
FPGA Synthesis	LUT	42	1074
	FF	30	2026
ASIC Synthesis	Cell	138	139
	Area	613 μm^2	9350μm^2
	Power @2kHz	0.47nW	0.71nW

the size of the memory blocks, we first consider that the codes are constructed for $M = 500$ ternary information symbols and each parity check is generated by every 3 symbols and each symbol has 2 bits. Therefore, we set a register size of 500/3*2 = 332 bit. A total 3 of such registers are used to represent $\delta_1^{(t)}$, $\delta_2^{(t)}$, and $\delta_3^{(t)}$. Each register has a buffer register to store the temporal data for reordering. Therefore, a total of six 332-bit registers are used as memory blocks. The FPGA and ASIC implementation results are presented in Table III. The power consumption is estimated by the synthesis tools based on the technology model parameters. The ASIC synthesis results include the core encoder circuits and the total encoder with a 2 Kb *RAM* module. The encoder is seen to occupy a small footprint and consumes little power compared to the sensor and the radio circuits, thanks to the proposed serial operation and the look-up table methods. The system parameters, such as the coefficients $\delta_{j,i}$ and γ_l, can be adjusted according to specific designs of the encoder. The overall system meets the design goals of power consumption and circuit area. The decoder circuit hardware is not a primary concern in the system since, unlike the encoder, the decoder in the receiver usually does not depend on a limited battery power supply.

V. SYSTEM PERFORMANCE ANALYSIS

The system performance is primarily evaluated by the classification accuracy under different noise levels. It is also

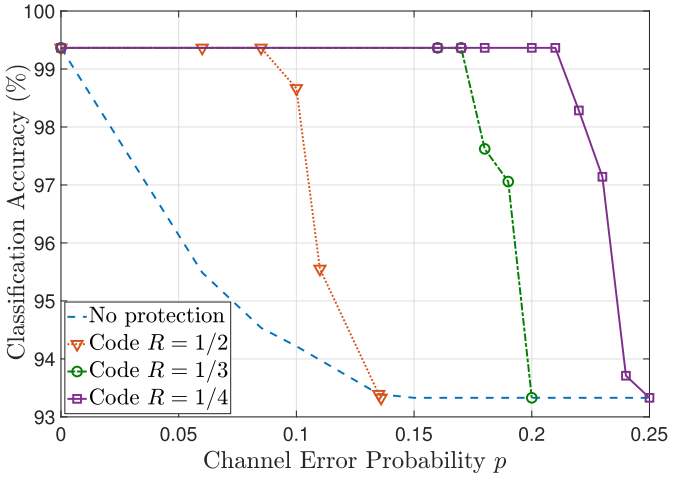


Fig. 11. Classification accuracy as a function of channel error probability p with LDPC coding at different coding rates.

evaluated by the extra power consumption for performing the error correction. The input signal is an ECG record from the MIT-BIH database, which is converted into ternary bitstreams by the Delta modulators. The bitstreams are encoded by the ternary generator matrix of the LDPC code as described in Section III-B to generate the transmitted symbols. Then noise is added to the transmitted symbols based on the ternary symmetric channel model. The LDPC decoder then estimates the transmitted symbols using the ternary SPA. Finally, the classification accuracy is compared between the data with and without error correction.

A. Classification Accuracy

As described in Section III-A, the arrhythmia classification accuracy is seen to decrease with increasing channel noise. The simulation result of the classification accuracy under different channel error probability p is presented in Fig. 11, where the dashed curve represents the transmitted ternary bitstreams converted from the ECG signal without error correction. The other curves represent the protected data by the ternary LDPC codes at different coding rates: from left to right $R = 1/2, 1/3,$ and $1/4$, respectively. As expected, we observe that the ternary LDPC codes with lower rates provide robust classification accuracy for a larger class of channels (increasing p). For example, with a code rate of $R = 1/4$, a classification accuracy can be maintained at approximately 99.4% for p less than or equal to approximately 0.21 (21% probability of channel error); whereas the $R = 1/3$ code can only maintain this accuracy for p up to approximately 0.17.

B. Power Analysis

The power cost of the error correction circuit in the ECG monitoring system has been studied. To transmit the bitstream from the ECG sensor, the total power cost of the system includes the sensing power, transmission power, and the power of the error correction circuitry. From our previous study [34], the LDPC encoder power is negligible compared to the transmission power. This is also valid in this work using 32 nm CMOS process since the estimated power cost of the ternary LDPC encoder is only 0.71 nW running with a 2kHz clock, which is fast enough to support the ternary sensor sampling at 1 kHz. Therefore, most of the extra power from the error correction system comes from the extra bits during communication. From [9], the power cost of the ternary Delta modulator is 720nW at a sampling rate of 1 kHz. Assuming the transmission power is 1nJ/bit [35] and the ternary bits are transmitted as two channels the total power without error correction is $1\text{nJ}\times 2\times 1\text{kHz}= 2\mu\text{W}$. With a code rate of $R = 1/4$, the total communication power becomes $2\mu\text{W}/(1/4) = 8\mu\text{W}$, and the additional power from the error correction are $8\mu\text{W}-2\mu\text{W}= 6\mu\text{W}$. Adding the sensing power of 720nW, the total power increased from 2.72μW to 8.72μW to boost the detection accuracy from 93.3% to above 99%. This power increase is acceptable for wearable ECG sensors. In the future, an unequal error protection (UEP) method can be applied to reduce the number of the extra bits for error correction to save the system power.

VI. Discussion

This paper provides several novel techniques and applications of wireless biosensors. First, the system and the target signals have special characteristics that need to be considered carefully in terms of error correction. To the best of our knowledge, this is the first paper that addresses error correction circuits for ternary Delta modulators for the application of ECG classification. Although the ternary Delta modulator was presented before, the error introduced during communication was not considered and may affect the decision-making process. As discussed above, few critical bits in error may lead to wrong treatment, which is not acceptable. We were unable to find any references in the literature that consider error correction with ternary data for wireless biosensors and therefore the error effects and mitigating error correction in communication for wearable biosensors with ternary sensors form a valuable contribution.

The second innovation is code design. We designed a novel irregular base parity-check matrix (i.e., selecting the γs and δs in the parity-check matrix) such that any code randomly selected from the resulting code ensemble permits efficient encoding. Otherwise, encoding an arbitrary ternary LDPC code would have unacceptably high complexity for such a device. In order to demonstrate an average performance, we select a random code from the ensemble and showed that the performance provides acceptable accuracy of classification over a wide variety of channel conditions. This delivers proof-of-concept that such a ternary LDPC coding system can be designed and implemented in an efficient way for such a power-restricted device.

The third contribution is the efficient encoder circuit design. Considering the application constraint of power and area in the wearable sensors, as well as the low frequency of the target signal, the implementation applies a serial generation of output parity symbols instead of conventional parallel encoding. The encoder generates the parity symbols directly from the highly structured parity-check matrix of the ternary LDPC code in a serial fashion. The codeword is formed by concatenating

the information symbols and the generated parity symbols. Compared to the regular parallel encoding scheme, this novel, power-efficient, serial encoding scheme is ideal for this application since speed is not the primary consideration.

To improve the system performance, several promising directions exist for future study. First, since a random code was selected in the current system to show proof-of-concept, we plan to perform code optimization while maintaining the novel efficient base structure. This is expected to increase the reliability/classification accuracy for a given code overhead or lower the required amount of overhead for a target reliability. Second, since the current channel model considers the worst-case scenario of uniformly likely errors, we plan to identify and generalize the ternary channel parameters using hardware devices to obtain a more realistic channel model. Third, we plan to apply an unequal error protection (UEP) method to reduce the number of extra bits for error correction to save the system power.

VII. Conclusion

A ternary LDPC error correction system is proposed to protect Delta modulated bitstreams in ECG sensors for arrhythmia classification. The channel is modeled as a symmetric ternary channel with an error probability p. A class of ternary LDPC codes are constructed by concatenating two-component sub-matrices with quasi-cyclic structure, which facilitates the hardware design. A hardware-efficient ternary LDPC encoder circuit is proposed using only an accumulator and a look-up table to generate the parity-check symbols serially. This is enabled by careful selection of coefficients in the code design to facilitate a low-power implementation of the encoder circuit. The hardware implementation and ASIC synthesis results show that the proposed encoder has a low power consumption and silicon footprint. The simulation results show that the arrhythmia classification accuracy can be maintained at approximately 99.4% for symbol error probability p up to 21% with the proposed ternary error correction system. The proposed design provides a promising solution for error correction in ternary wireless sensors.

References

[1] X. Tang, Q. Hu, and W. Tang, "Delta–Sigma encoder for low-power wireless bio-sensors using ultrawideband impulse radio," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 64, no. 7, pp. 747–751, Jul. 2017.

[2] X. Zhang, Z. Zhang, Y. Li, C. Liu, Y. X. Guo, and Y. Lian, "A 2.89-μW dry-electrode enabled clockless wireless ECG SoC for wearable applications," *IEEE J. Solid-State Circuits*, vol. 51, no. 10, pp. 2287–2298, Jul. 2016.

[3] M. Llamedo and J. P. Martinez, "Heartbeat classification using feature selection driven by database generalization criteria," *IEEE Trans. Biomed. Eng.*, vol. 58, no. 3, pp. 616–625, Mar. 2011.

[4] J. M. Bote, J. Recas, F. Rincón, D. Atienza, and R. Hermida, "A modular low-complexity ecg delineation algorithm for real-time embedded systems," *IEEE J. Biomed. Health Inform.*, vol. 22, no. 2, pp. 429–441, Mar. 2018.

[5] X. Liu *et al.*, "A 457 nW near-threshold cognitive multi-functional ECG processor for long-term cardiac monitoring," *IEEE J. Solid-State Circuits*, vol. 49, no. 11, pp. 2422–2434, Nov. 2014.

[6] D. Da He and C. G. Sodini, "A 58 nW ECG ASIC with motion-tolerant heartbeat timing extraction for wearable cardiovascular monitoring," *IEEE Trans. Biomed. Circuits Syst.*, vol. 9, no. 3, pp. 370–376, Sep. 2015.

[7] C. Ieong *et al.*, "A 0.83-μW QRS detection processor using quadratic spline wavelet transform for wireless ECG acquisition in 0.35-μm CMOS," *IEEE Trans. Biomed. Circuits Syst*, vol. 6, no. 6, pp. 586–595, Apr. 2012.

[8] Y. H. Awni *et al.*, "Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network," *Nature Med.*, vol. 25, no. 1, pp. 65–69, Jan. 2019.

[9] X. Tang, Q. Hu, and W. Tang, "A real-time QRS detection system with PR/RT interval and ST segment measurements for wearable ECG sensors using parallel delta modulators," *IEEE Trans. Biomed. Circuits Syst.*, vol. 12, no. 4, pp. 751–761, Aug. 2018.

[10] X. Tang and W. Tang, "A 151nW second-order ternary delta modulator for ECG slope variation measurement with baseline wandering resilience," in *Proc. IEEE Custom Integr. Circuits Conf. (CICC)*, Mar. 2020, pp. 1–4.

[11] X. Tang, Z. Ma, Q. Hu, and W. Tang, "A real-time arrhythmia heartbeats classification algorithm using parallel delta modulations and rotated linear-kernel support vector machines," *IEEE Trans. Biomed. Eng.*, vol. 67, no. 4, pp. 978–986, Apr. 2020.

[12] D. Zrilic, G. Petrovic, and W. Tang, "Novel solutions of a Delta–Sigma-based rectifying encoder," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 64, no. 10, pp. 1242–1246, Oct. 2017.

[13] R. G. Gallager, "Low-density parity-check codes," *IRE Trans. Inf. Theory*, vol. 8, no. 1, pp. 21–28, Jan. 1962.

[14] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inf. Theory*, vol. 45, no. 2, pp. 399–431, Mar. 1999.

[15] M. G. Luby *et al.*, "Improved low-density parity-check codes using irregular graphs," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 585–598, Feb. 2001.

[16] T. J. Richardson and R. L. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 599–618, Feb. 2001.

[17] C. Di, A. Montanari, and R. Urbanke, "Weight distributions of LDPC code ensembles: Combinatorics meets statistical physics," in *Proc. Int. Symp. Inf. Theory (ISIT)*, 2004, p. 102.

[18] D. Burshtein and G. Miller, "Efficient maximum-likelihood decoding of LDPC codes over the binary erasure channel," *IEEE Trans. Inf. Theory*, vol. 50, no. 11, pp. 2837–2844, Nov. 2004.

[19] A. Amraoui, R. Urbanke, and A. Montanari, "Finite-length scaling of irregular LDPC code ensembles," in *Proc. IEEE Inf. Theory Workshop*, Rotorua, New Zealand, Aug. 2005, p. 5.

[20] M. C. Davey and D. MacKay, "Low-density parity check codes over GF(q)," *IEEE Commun. Lett.*, vol. 2, no. 6, pp. 165–167, Jun. 1998.

[21] C. Poulliat, M. Fossorier, and D. Declercq, "Design of regular (2, d_c)-LDPC codes over GF(q) using their binary images," *IEEE Trans. Commun.*, vol. 56, no. 10, pp. 1626–1635, Oct. 2008.

[22] L. Zeng, L. Lan, Y. Y. Tai, B. Zhou, S. Lin, and K. A. S. Abdel-Ghaffar, "Construction of nonbinary cyclic, quasi-cyclic and regular LDPC codes: A finite geometry approach," *IEEE Trans. Commun.*, vol. 56, no. 3, pp. 378–387, Mar. 2008.

[23] J. Robinson, "An asymmetric error-correcting ternary code (Corresp.)," *IEEE Trans. Inf. Theory*, vol. IT-24, no. 2, pp. 258–261, Mar. 1978.

[24] N. Bitouzé, A. G. I. Amat, and E. Rosnes, "Error correcting coding for a nonsymmetric ternary channel," *IEEE Trans. Inf. Theory*, vol. 56, no. 11, pp. 5715–5729, Nov. 2010.

[25] D. V. Efanov, "Ternary parity codes: Features," in *Proc. IEEE East-West Design Test Symp. (EWDTS)*, Sep. 2019, pp. 1–5.

[26] S. Escalera, O. Pujol, and P. Radeva, "On the decoding process in ternary error-correcting output codes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 1, pp. 120–134, Jan. 2010.

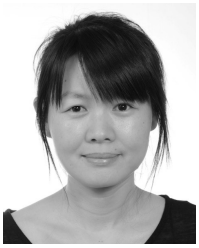
[27] F.-W. Fu, S. Ling, and C. Xing, "Constructions of nonbinary codes correcting T-symmetric errors and detecting all unidirectional errors: Magnitude error criterion," in *Coding, Cryptography Combinatorics*. Basel, Switzerland: Birkhäuser, 2004, pp. 139–152.

[28] C.-F. Lan, A. D. Liveris, K. Narayanan, Z. Xiong, and C. Georghiades, "Slepian-wolf coding of multiple M-ary sources using LDPC codes," in *Proc. Data Compress. Conf. (DCC)*, 2004, p. 549.

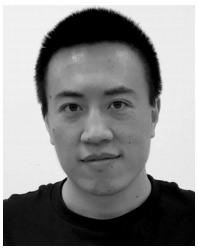
[29] Association for the Advancement of Medical Instrumentation and Others, *Testing and Reporting Performance Results of Cardiac Rhythm and ST Segment Measurement Algorithms*, Standard ANSI/AAMI EC38, 1998.

[30] S. Timakul and S. Choomchuay, "A construction of non binary LDPC codes by circular matrices," in *Proc. Adv. Mater. Res.*, vol. 909. Cham, Switzerland: Trans Tech Publications, 2014, pp. 338–341.

[31] Y. Sun, M. Karkooti, and J. Cavallaro, "High throughput, parallel, scalable LDPC encoder/decoder architecture for OFDM systems," in *Proc. IEEE Dallas/CAS Workshop Design, Appl., Integr. Softw.*, Oct. 2006, pp. 39–42.
[32] S. Lin and D. J. Costello, *Error Control Coding*, vol. 2, no. 4. New York, NY, USA: Pearson, 2001.
[33] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.
[34] Y. Liu, W. Tang, and D. G. M. Mitchell, "Efficient implementation of a threshold modified min-sum algorithm for LDPC decoders," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 67, no. 9, pp. 1599–1603, Sep. 2020.
[35] Q. Hu, X. Tang, and W. Tang, "Integrated asynchronous ultrawideband impulse radio with intrinsic clock and data recovery," *IEEE Microw. Wireless Compon. Lett.*, vol. 27, no. 4, pp. 416–418, Apr. 2017.
[36] Medtronic. *Seeq MCT Discontinued*. Accessed: Dec. 1, 2020. [Online]. Available: https://www.medtronic.com/us-en/healthcare-professionals/products/cardiac-rhythm/cardiac-monitors/mct.html
[37] N. Ravanshad, H. Rezaee-Dehsorkh, R. Lotfi, and Y. Lian, "A level-crossing based QRS-detection algorithm for wearable ECG sensors," *IEEE J. Biomed. Health Informat.*, vol. 18, no. 1, pp. 183–192, Jan. 2014.
[38] Z. Wang *et al.*, "A software-defined always-on system with 57–75-nW wake-up function using asynchronous clock-free pipelined event-driven architecture and time-shielding level-crossing ADC," *IEEE J. Solid-State Circuits*, early access, May 21, 2021, doi: 10.1109/JSSC.2021.3074636.
[39] X. Yang *et al.*, "An in-memory-computing charge-domain ternary CNN classifier," in *Proc. IEEE Custom Integr. Circuits Conf. (CICC)*, Apr. 2021, pp. 1–2.

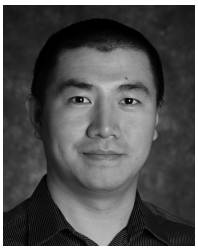


Yanfang Liu (Member, IEEE) received the B.S. degree in electronic information technology and the M.S. degree in detecting techniques and automation from the Beijing Information Technology Institute, Beijing, China, in 2004 and 2007, respectively, and the Ph.D. degree from the Departamento de Teoría de la Señal y Comunicaciones, Universidad Carlos III de Madrid, Leganés, Spain, and the Instituto de Investigación Sanitaria Gregorio Marañón (IiSGM), Madrid, Spain, in 2019. She was an IC Design Engineer with Beijing Telepath Technologies Company Ltd., Beijing, from 2008 to 2012, and the Institute of Electronics, Chinese Academic of Science, Beijing, from 2012 to 2014. She is currently with the Klipsch School of Electrical and Computer Engineering, New Mexico State University, Las Cruces, NM, USA. Her current research interests include low-density parity-check codes (LDPC), generalized LDPC codes, and hardware implementation.



Xiaochen Tang (Member, IEEE) received the B.E. degree in electronics science and technology and the M.E. degree in microelectronics and solid-state electronics from the Harbin Institute of Technology, Harbin, China, in 2010 and 2012, respectively, and the Ph.D. degree in electrical engineering from New Mexico State University, Las Cruces, NM, USA, in 2020. He is currently a Post-Doctoral Researcher with Texas A&M University, College Station, TX, USA. His research interests include low-power biomedical integrated circuits and wearable biomedical devices.

David G. M. Mitchell (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from The University of Edinburgh, Edinburgh, U.K., in 2009. From 2009 to 2015, he was a Post-Doctoral Research Associate and a Visiting Assistant Professor with the Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN, USA. Since 2015, he has been an Assistant Professor with the Klipsch School of Electrical and Computer Engineering, New Mexico State University, Las Cruces, NM, USA. His research interests include digital communications, with emphasis on error control coding and information theory. He is currently serving as an Associate Editor for the IEEE TRANSACTIONS ON INFORMATION THEORY.



Wei Tang (Member, IEEE) received the B.Sc. degree in microelectronics from Peking University, Beijing, China, in 2006, and the Ph.D. degree in electrical engineering from Yale University, New Haven, CT, USA, in 2012. In 2012, he joined the Klipsch School of Electrical and Computer Engineering, New Mexico State University, Las Cruces, NM, USA, as an Assistant Professor, where he was promoted to an Associate Professor in 2018. He is currently an Associate Professor with the Klipsch School of Electrical and Computer Engineering, New Mexico State University. His research interests include low-power analog/mixed-signal/digital/RF integrated circuits design and implementation, hardware-friendly digital signal processing and image processing algorithms, and biomedical sensors and wearable devices, circuits, and systems. He was a recipient of the National Science Foundation Faculty Early Career Award in 2017. He currently holds the Paul W. and Valerie Klipsch Distinguished Professorship at New Mexico State University.