Construction of Low-Rate LDPC Codes from Rate-$1/2$ CCSDS Standard LDPC Codes

Semira Galijasevic
University of California
420 Westwood Plaza
Los Angeles, CA 90095
semiragali@g.ucla.edu

Linfang Wang
University of California
420 Westwood Plaza
Los Angeles, CA 90095
lfwang@g.ucla.edu

Jon Hamkins
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Dr.
Pasadena, CA 91109
jon.hamkins@jpl.nasa.gov

Richard Wesel
University of California
420 Westwood Plaza
Los Angeles, CA 90095
wesel@g.ucla.edu

Dariush Divsalar
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Dr.
Pasadena, CA 91109
dariush.divsalar@jpl.nasa.gov

Abstract— **The existing Consultative Committee for Space Data Systems (CCSDS) standard uses low-density parity-check (LDPC) codes for higher code rates including $1/2, 2/3$, and $4/5$, supporting message block lengths of $1024, 4096$, and 16384. For lower code rates, the CCSDS standard uses turbo codes, providing rates of $1/3, 1/4$ and $1/6$ and supporting message block lengths of $1784, 3568$, and 16384. However, the frame error rate (FER) performance of the turbo codes shows an error floor where the slope of frame error rate curve begins to flatten between FERs of 10^{-3} and 10^{-5}. The resulting FER performance is undesirable for certain space applications.**

This paper uses the Protograph-Based Raptor-Like (PBRL) approach to provide new lower LDPC code rates of $1/3, 1/4$ and $1/6$ for the existing LDPC message lengths of 1024, 4096, and 16384 that are rate-compatible with the existing CCSDS rate-$1/2$ LDPC codes and do not suffer from an error floor (at least above FER 10^{-7}).

TABLE OF CONTENTS

1. INTRODUCTION

History of LDPC Codes

Low-density parity-check (LDPC) codes are a class of linear block codes invented by Gallager in his 1963 doctoral dissertation [1]. Despite offering near-capacity performance across a wide range of data-transmission and data-storage channels, implementation seemed out of reach at the time of the invention. LDPC codes were largely overlooked until Tanner's work in 1981 that brought renewed attention to them. Tanner generalized LDPC codes and introduced a graphical representation of LDPC codes, now called a Tanner graph [2].

The introduction of turbo codes by Berrou et al. [3] in 1990s sparked further interest in LDPC codes, leading to significant contributions from numerous researchers including MacKay and Luby [4–7] who recognized the benefits of linear block codes with sparse (low-density) parity-check matrices.

A fundamental LDPC code structure that facilitates efficient encoding while achieving good frame error rate (FER) performance is the Repeat Accumulate (RA) code [8] introduced in 1998. The original RA codes were regular in the sense that all variable nodes have the same degree, and all check nodes have the same degree. In 2000 RA codes were extended to include Irregular RA (IRA) codes [9]. Some advantages offered by IRA codes include greater flexibility in selecting the repetition rate for each information bit which enables the design of high-rate codes. Additionally, their irregularity enables operation that approaches capacity more closely.

In 2001 Richardson, Urbanke, and Shokrollahi [10, 11] demonstrated that using density evolution to optimize the degree distribution of the variable and check nodes produces capacity approaching LDPC codes. In 2003, Thorpe [12] showed that LDPC codes can be constructed by copying and permuting a small base graph, called a protograph, with only few variable and check nodes. The parity-check matrix of the protograph is called protomatrix and the resulting code is called a Protograph LDPC code. In this paper, protograph and protomatrix are used interchangeably.

The introduction of Accumulate Repeat Accumulate (ARA) Codes in 2004 and later in 2007 [13] achieved a significant improvement in the performance of LDPC codes. A key contribution was the ability to reduce the iterative decoding threshold to obtain capacity approaching LDPC codes without increasing the degrees of certain variable nodes as was required in [10, 11]. This was accomplished by deleting (or ”puncturing”) selected bits from the LDPC transmitted codeword.

In 2005, Divsalar et al. [14] showed that the number of degree-2 nodes in LDPC code should be less than the number of check nodes attached to them in order to have an LDPC code with minimum distance growing linearly with block size. This discovery inspired the development of Accumulate Repeat Jagged Accumulate (ARJA) codes which were proposed to CCSDS in 2007 and became the international standard for space applications.

979-8-3503-5597-0/25/$31.00 ©2025 IEEE

A few years later, Protograph-based Raptor like (PBRL) LDPC codes were introduced [15, 16], enabling construction of low-rate LDPC codes with good performance. PBRL codes naturally provide a rate-compatible family of codes supporting a wide range of rates. The PBRL structure features a highest rate code (HRC) and an incremental redundancy code (IRC) that provides the addition symbols used to achieve the lower rates.

The current CCSDS LDPC codes are proposed for the Human Landing System on the Moon and the NASA Artemis program to explore Mars via the Moon. While the development of LDPC codes has been highly successful, offering performance and complexity benefits over turbo codes at high code rates, turbo codes remained the optimal solution for lower code rates until now and have been utilized in the CCSDS standard [17]. Replacing the low-rate turbo codes with better-performing PBRL LDPC codes would allow the CCSDS codes to be used for longer distances and would also support dynamic rate adaptation for fading channels.

Contribution

This paper presents a new class of low-rate PBRL LDPC Codes for lower rates $1/6, 1/4, 1/3$ for message lengths of $K = 1024, 4096$ and 16384 which avoid the error floor behavior seen in the current CCSDS turbo codes. Deep space communication to Mars and beyond requires transmitting data at low rates. The current CCSDS standard codes for rates less than $1/2$ are turbo codes, which are hindered by an error floor. Even increasing power cannot achieve error rates below this floor which is usually between 10^{-4} and 10^{-5} range of frame error rates.

As a remedy, this paper provides new PBRL LDPC codes with rates below $1/2$ by attaching a Low-Density Generator Matrix (LDGM) code to the existing rate-$1/2$ CCSDS LDPC code. In software simulations, the new low-rate LDPC/LDGM codes offer better performance than the current turbo codes. By overcoming the error floor, the LDPC codes provide the very low FERs necessary to support reliable transmission of compressed data. The lower-rate LDPC encoder will use the current CCSDS encoder and generate the additional parity check bits based on the output of current CCSDS code using Exclusive OR operation only. The overall decoder can be implemented as a single decoder that supports all code rates or the current LDPC decoder could be augmented with an LDGM decoder.

Fig.1 shows LDPC encoding and decoding for the new low code rate codes. In this paper, the selected HRC is exactly the existing CCSDS rate-$1/2$ code. The protograph for IRC parts for each message length is designed to achieve the best possible threshold for each successively lower rate according to Reciprocal Channel Approximation (RCA) [16]. The final rate-compatible LDPC codes are achieved by a two-step lifting procedure. First the protograph is lifted by four to create a protograph that has only zeros and ones and to ensure a girth of at least four. Using the concept of extrinsic message degree (EMD), the IRC protograph is lifted a second time by 128 or 512 or 2048 to optimize the approximate cycle EMD (ACE) [18] of the lowest rate code without changing the structure of HRC code, which is the rate-$1/2$ CCSDS LDPC code.

The addition of lower rates to the CCSDS standard would allow the codes to be used for longer distances and would also allow for dynamic rate adaptation to adapt to a fading channel. The current CCSDS LDPC codes are proposed to be used for Human Landing System on Moon and NASA Artemis program Moon/Mars. For future missions at much longer distances such as Mars, lower code rates with much better performance than turbo codes are desirable. Eventually the constructed low-rate LDPC codes will be proposed to CCSDS standard to be used for space applications. The lower rate LDPC encoder will use the current CCSDS encoder and generates the additional parity check bits from the output of the current CCSDS encoder using Exclusive-OR operations.

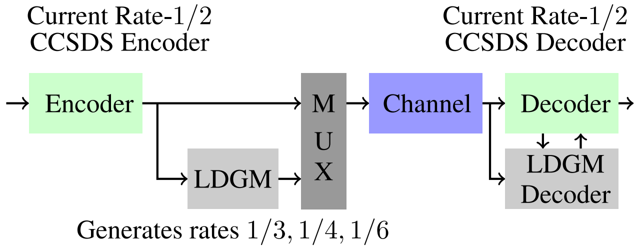

Figure 1. Encoder/Decoder for low-rate PBRL LDPC codes without changing the current CCSDS code.

In this paper we describe the construction of the new codes and show that their performance outperforms turbo code FERs by several orders of magnitude in the error floor region. The structure of 5G-NR LDPC codes proposed in 2017 have the PBRL code structure that was jointly proposed by UCLA and JPL/Caltech in IEEE Transaction on Communications 2015 [16] and before that in a 2011 conference paper [15].

We will gain improvements in the overall performance of the deep space coding system in terms of lower required signal-to-noise ratio, increased data return, and lower FER by at least by three orders of magnitude with respect to turbo codes. Low-rate codes enable more reliable communication for the NASA Artemis program for future Moon/Mars and Human Landing Systems and exploration beyond the Moon.

Organization

The rest of the paper proceeds as follows: Sec. 2 describes the PBRL approach to LDPC code construction and describes the newly designed LDPC codes. Sec. 3 compares the FER performance of the newly designed LDPC codes with the FER performance of CCSDS turbo codes having similar rates and block lengths. Sec. 4 concludes the paper.

2. PBRL LDPC Code Construction

Let $\boldsymbol{P}$ given in (1) be an $n_c \times n_v$ protomatrix of PBRL LDPC code with n_c check nodes and n_v variable nodes:

$$\boldsymbol{P} = \begin{bmatrix} \boldsymbol{P}_{\text{HRC}} & \boldsymbol{0} \\ \boldsymbol{P}_{\text{IRC}} & \boldsymbol{I} \end{bmatrix}. \tag{1}$$

In (1), the highest rate code of protograph is represented by $\boldsymbol{P}_{\text{HRC}}$ protomatrix. The $\boldsymbol{0}$ and $\boldsymbol{I}$ matrices are respectively the all-zeros and identity matrices of appropriate sizes. Together, the $\boldsymbol{P}_{\text{IRC}}$ and $\boldsymbol{I}$ matrices specify the additional rows of the parity check matrix that lower the rate by adding additional parity symbols to codewords of the highest rate code.

Let n_p denote the number of "punctured" variable nodes, i.e. protograph variable nodes intentionally excluded from transmission in the code structure. The rate of a protograph is

$R = \frac{n_v - n_c}{n_v - n_p}$. This is the lowest rate supported by the PBRL LDPC code.

Our goal is to design the protomatrix $\boldsymbol{P}$ and lift it to produce the needed low-rate LDPC codes.

The Highest Rate Code (HRC)

We use the protograph of the existing rate-$1/2$ CCSDS LDPC code to obtain $\boldsymbol{P}_{\mathrm{HRC}}$ for our new codes. The protograph of the existing rate$-1/2$ CCSDS LDPC code is described in (2). The last column of this protograph is designated for puncturing.

$$\boldsymbol{P_{CCSDS}} = \begin{bmatrix} 0 & 0 & 1 & 0 & 2 \\ 1 & 1 & 0 & 1 & 3 \\ 1 & 2 & 0 & 2 & 1 \end{bmatrix} \tag{2}$$

Since $\boldsymbol{P_{CCSDS}}$ does not have a favorable PBRL structure as in (1) to facilitate the design process, this paper obtains $\boldsymbol{P}_{\mathrm{HRC}}$ described in (3) by permuting rows and columns of the existing CCSDS rate-$1/2$ code given in (2) for all message lengths. Note that with the permutation the last column of CCSDS code in (2) becomes the first column of $\boldsymbol{P}_{\mathrm{HRC}}$ in (3) and thus the first column in (3) is designated to be punctured.

$$\boldsymbol{P}_{\mathrm{HRC}} = \begin{bmatrix} 3 & 1 & 1 & 1 & 0 \\ 1 & 1 & 2 & 2 & 0 \\ 2 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

We use (3) for the PBRL design process. However, after code optimization and lifting, the columns and rows of the HRC are inversely permuted to recover the rate-$1/2$ parity check matrix H as described in the CCSDS standard with exactly the circulant permutations of the CCSDS standard as the HRC in the final PBRL matrix.

Incremental Redundancy Code (IRC) Design

The identity matrix $\boldsymbol{I}$ in (1) generates degree-1 variable nodes, corresponding to the incremental redundancy symbols of $\boldsymbol{P}$. The check nodes that connect to the degree-1 variable nodes also have connections to the variable nodes in the HRC part. These connections are expressed by the sub-matrix $\boldsymbol{P}_{\mathrm{IRC}}$, which represents the incremental redundancy code. This section presents the design of P_{IRC} to support rates $1/3$, $1/4$ and $1/6$ for message lengths of $K = 1024, 4096, 16384$. The existing rates and code block lengths of CCSDS code along with new rates and code block lengths are presented in Table 1.

Table 1. Code Block Lengths for Supported CCSDS Code Rates and Newly Designed Code Rates

Message Length K	**Code block length:** $n - n_p$					
	CCSDS Rates			New Rates		
	4/5	2/3	1/2	1/3	1/4	1/6
1024	1280	1536	2048	3072	4096	6144
4096	5120	6144	8192	12288	16384	24576
16384	20480	24576	32768	49152	65536	98304

The protomatrix $\boldsymbol{P}_{\mathrm{IRC}}$ is designed row by row to achieve the best possible thresholds for each successively lower rate according to Reciprocal Channel Approximation (RCA) [16]. The RCA is a fast and accurate approximation to the density evolution algorithm with deviation in accuracy of less than 0.01 dB. The RCA for the binary-input (BI) AWGN channel (BPSK modulation with soft output demodulation) uses a single real-valued parameter s, the effective signal-to-noise ratio (SNR), and its reciprocal r to approximate the densities of the density evolution.

Algorithm 1 Greedy Design for Protomatrix $\boldsymbol{P}_{\mathrm{IRC}}$

```
Require: Initial $\boldsymbol{P_0} = \boldsymbol{P}_{\mathrm{HRC}}$, Row number $t = 8$
  $i \leftarrow 1$
  while $i \le t$ do
      $\boldsymbol{h}_i = \arg\min_{\boldsymbol{h}_i \in \boldsymbol{A}} \delta(\boldsymbol{P_i})$,
      where $\boldsymbol{P_i}$ is computed from $\boldsymbol{P_{i-1}}$ via (7)-(8).
      $i = i + 1$
  end while
  return $[\boldsymbol{h}_1^T \dots \boldsymbol{h}_t^T]^T$
```

The RCA algorithm essentially models all of the message densities as Gaussian distributions and passes a single parameter to describe each Gaussian density. In the context of the RCA algorithm, s_e is the message passed along an edge e from a variable node to a check node and r_e is the message passed along an edge e from a check node to a variable node. The message s_e may be interpreted the SNR for the Gaussian message density corresponding to that edge on that iteration. The message r_e is the *reciprocal* SNR for the Gaussian message density corresponding to that edge on that iteration. The reciprocal SNR r_e is defined so that $C(s_e) + C(r_e) = 1$, where $C(x)$ is the capacity of the BI-AWGN channel with SNR x.

Let s_{chl} be the channel SNR, where

$$s_{chl} = 2\frac{E_c}{N_0} = 2R\frac{E_b}{N_0} \tag{4}$$

where $2E_c/N_0$ is the code-symbol signal-to-noise ratio, E_b is energy per information bit, and R is the code rate. If the edge e is connected to a punctured variable node the message s_e is initialized to 0. Otherwise s_e is initialized to s_{chl}. RCA computes a sequence of messages $({s_e}^{(n)}, {r_e}^{(n)}), n = 0, \dots, N$, where N represents the maximum number of iterations. The approximated iterative decoding threshold s_{th} is determined as the minimum s_{chl} such that ${s_e}^{(N)} > T$ for all edges e in the graph, where T is a stopping threshold. The full RCA algorithm can be found in [16].

Alg. 1 describes IRC design process. At every iteration i, a new row $\boldsymbol{h}_i$ of $\boldsymbol{P}_{\mathrm{IRC}}$ is designed to minimize the threshold $\delta(\boldsymbol{P}_i)$of $\boldsymbol{P}_i$, where

$$\boldsymbol{P_i} = \begin{bmatrix} \boldsymbol{P}_{\mathrm{HRC}} & \boldsymbol{0} \\ \boldsymbol{P}_{\mathrm{IRC,i}} & \boldsymbol{I} \end{bmatrix}, \tag{5}$$

and

$$\boldsymbol{P}_{\mathrm{IRC,i}} = [\boldsymbol{h}_1^T \dots \boldsymbol{h}_i^T]^T. \tag{6}$$

At iteration i, $\boldsymbol{P_{i-1}}$ has been completely specified as

$$\boldsymbol{P_{i-1}} = \begin{bmatrix} \boldsymbol{P}_{\mathrm{HRC}} & \boldsymbol{0}_{3\times i} \\ \boldsymbol{P}_{\mathrm{IRC,i-1}} & \boldsymbol{I}_{i-1\times i-1} \end{bmatrix}, \tag{7}$$

so that only the row vector $\boldsymbol{h}_i$ is allowed to vary to improve the RCA threshold. Thus, $\boldsymbol{h}_i$ is selected to minimize the threshold $\delta(\boldsymbol{P_i})$ (calculated using RCA [16]) of

$$\boldsymbol{P_i} = \begin{bmatrix} \boldsymbol{P}_{\text{HRC}} & \mathbf{0}_{3\times i} & \mathbf{0}_{3\times 1} \\ \boldsymbol{P}_{\text{IRC,i-1}} & \boldsymbol{I}_{i-1\times i-1} & 0 \\ \boldsymbol{h}_i & \mathbf{0}_{1\times i-1} & 1 \end{bmatrix}. \qquad (8)$$

After t iterations, the final IRC protomatrix is given by:

$$\boldsymbol{P}_{\text{IRC}} = [\boldsymbol{h}_1^T \dots \boldsymbol{h}_t^T]^T. \qquad (9)$$

To obtain the lowest code rate of $1/6$, we set $t = 8$ and consider the following $\boldsymbol{h}_i$'s:

$$\boldsymbol{h}_i \in \boldsymbol{A} = \{[1\ b_1\ b_2\ b_3] \mid b_j \in \{0,1\}\}. \qquad (10)$$

For all message lengths considered in this paper a single protomatrix $\boldsymbol{P}_{\text{IRC}}$ is designed consisting of 8 rows to support the lowest code rate of $1/6$. The final designed protomatrix $\boldsymbol{P}$ in (1) is shown below, with dimensions 11×13.

$$\boldsymbol{P} = \begin{bmatrix} 3 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 2 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Protographs for rates $1/4$ and $1/3$ are obtained by puncturing the degree-one variable nodes corresponding to the last four or six columns of $\boldsymbol{P}$, respectively.

Quasi-Cyclic LDPC Code and Lifting

CCSDS LDPC codes are members of a class of codes called Quasi-Cyclic (QC) codes. QC LDPC codes are characterized by a parity-check matrix that consists of square sub-matrices that are either the zero matrix or a cyclic permutation σ^i of the identity matrix, also called a circulant where σ is the right circular shift of the identity matrix of the same size. Such parity check matrices are constructed from a protomatrix through a two-step process known as lifting. We construct the QC-LDPC parity check matrix $\boldsymbol{H}$ from protomatrix $\boldsymbol{P}$ in (1) as follows.

The first step of lifting constructs the 44×52 matrix $\boldsymbol{H}_1$ in Fig. 2 by replacing each element in $\boldsymbol{P}$ with square sub-matrices of size $Z_1 = 4$. If the element in $\boldsymbol{P}$ is 0, then the sub-matrix is a zero matrix of size $Z_1 = 4$. If the element in $\boldsymbol{P}$ is $k > 0$, then the sub-matrix equals to $\sigma^{s_1} + \ldots + \sigma^{s_k}$, where $s_1 \ldots s_k \in \{0 \ldots Z_1 - 1\}$ and $s_i \neq s_j$ for $i \neq j$.

Note that the lifting by 4 also preserves original rate-$1/2$ CCSDS matrix with circulants, i.e. the first 12×20 part of $\boldsymbol{H}_1$ matrix after inverse permutation of columns is exactly the rate-$1/2$ CCSDS code lifted by 4 and provided in the CCSDS standard [19].

The second lifting step constructs the QC LDPC parity-check matrix $\boldsymbol{H}$ by replacing the elements in $\boldsymbol{H}_1$ with zero matrices or circulant matrices of size $Z_2 \times Z_2$ to optimize approximate-cycle extrinsic-message-degree (ACE) [18]. The value of Z_2 is 128, 512, or 2048, corresponding to message lengths $K = 1024, 4096, 16384$, respectively. The ACE algorithm with parameters of d_{ACE} and η ensures that the lifted parity check matrix has the property that all the cycles whose length is $2d_{\text{ACE}}$ or less in the bipartite graph associated with the parity check matrix have ACE values of at least η. Table 2 provides the values of d_{ACE} and η that were enforced for the three message lengths of interest.

3. Comparison of LDPC Codes to turbo Codes

The current CCSDS standard codes with rates less than $1/2$ are turbo codes hindered by an error floor—we cannot decode data frames with arbitrarily low error rates as we increase our transmit power. In the error floor region, the slope of the turbo code FER curve becomes significantly flatter [17]. The newly constructed LDPC codes are designed to overcome the error floor drawbacks of the current CCSDS turbo codes.

Since CCSDS turbo codes do not support message lengths of 1024 and 4096 we compare the performance of LDPC codes to the turbo codes by simulation and by using normal approximation by Polyanskiy [20]. The normal approximation refines the classical Shannon capacity result by providing tight approximations of the achievable coding rate for finite block lengths n as short as 100 bits. The maximum coding rate $R^*(n,\epsilon)$ with a final block length n and error probability ϵ is approximated by:

$$R^*(n,\epsilon) \approx C - \sqrt{\frac{V}{n}}Q^{-1}(\epsilon) + \frac{\log_2 n}{2n} \qquad (11)$$

where C is channel capacity, V is channel dispersion and $Q^{-1}(\epsilon)$ is inverse Gaussian function. Comparison of the newly constructed LDPC codes to the turbo codes is obtained by comparing frame error rates (FERs) of both codes to normal approximation in addition to simulations. The closer the frame error rate of the code is to the rate predicted by the normal approximation, the better the code performs.

Turbo Codes Simulation Setup

As described in CCSDS Blue Book [21] and presented in Fig. 3, the CCSDS turbo encoder consists of two encoders that generate parity symbols for two recursive convolutional

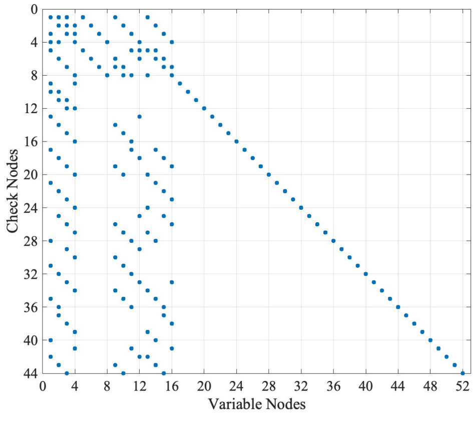

Figure 2. The 44 × 52 matrix $\boldsymbol{H}_1$ after lifting by 4 of 11 × 13 LDPC protograph. Dots indicate ones.

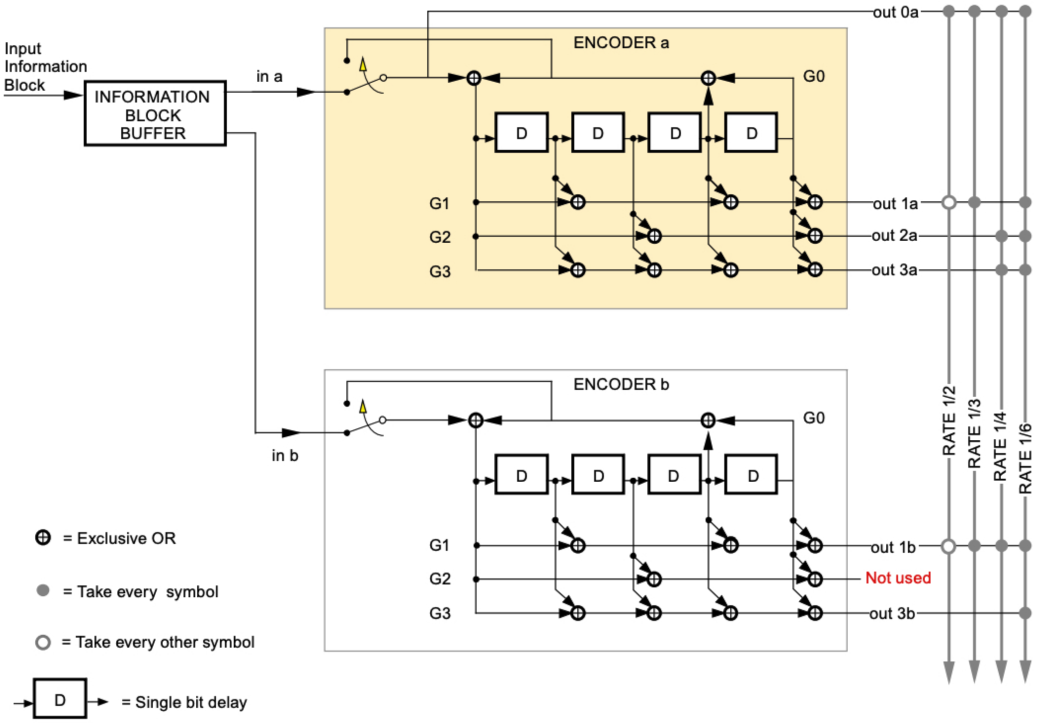

Figure 3. Turbo code encoder block diagram from CCSDS Blue Book.

Table 2. **d_{ACE} and η for different message lengths**

Message Length K	1024	4096	16384
d_{ACE}	5	5	5
η	21	26	33
Z_2	128	512	2048

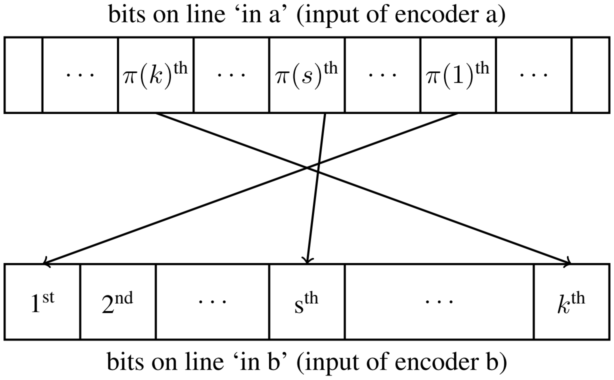

Figure 4. **Interpretation of Turbo Code Permutation.**

codes, each with a small number of states. A main characteristic of turbo codes is the use of an interleaver, which permutes bit-wise the original K information bits before input to the second encoder. The interleaver for turbo codes is a fixed bit-by-bit permutation of the entire block of data. Since the turbo code encoder is systematic, the information bits are sent uncoded in output 0a of Fig. 3. The turbo code permutation for any specified message length K must follow a specific reordering of the integers $1, 2, ..., K$ determined by the following algorithm which is in the CCSDS Blue Book.

Table 3. **Parameters k_1 and k_2 for Specified Information Block Lengths from CCSDS Blue Book.**

Information block length	k_1	k_2
1016	8	127
1024	8	128
4072	8	509
4096	8	128×4
16384	8	128×16

1. K shall be expressed as $K = k_1 k_2$, where k_1 and k_2 for the specified block sizes are given in Table 3.
2. The following operations shall be performed for $s = 1$ to $s = k$ to obtain permutation numbers $\pi(s)$:

$$m = (s-1) \mod 2$$

$$i = \lfloor \frac{s-1}{2k_2} \rfloor$$

$$j = \lfloor \frac{s-1}{2} \rfloor - ik_2$$

$$t = (19i + 1) \mod \frac{k_1}{2}$$

$$q = t \mod 8 + 1$$

$$c = (p_q j + 21m) \mod k_2$$

$$\pi(s) = 2\left(t + c\frac{k_1}{2} + 1\right) - m$$

Where $\lfloor x \rfloor$ denotes the largest integer less than or equal to x,

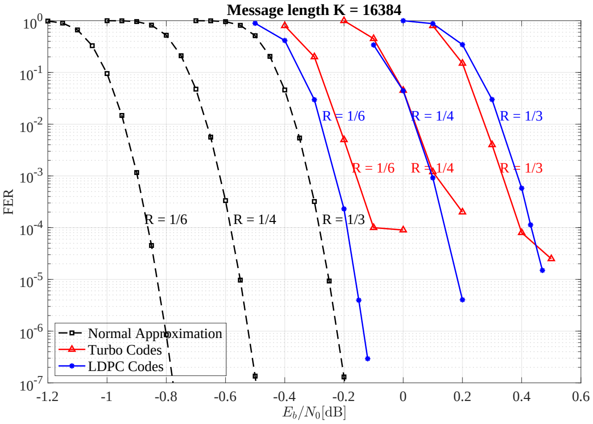

Figure 5. FER as a function of E_b/N_0 of LPDC Code compared to normal approximation and turbo codes for rates $R = 1/6, 1/4, 1/3$ and message length $K = 16384$.

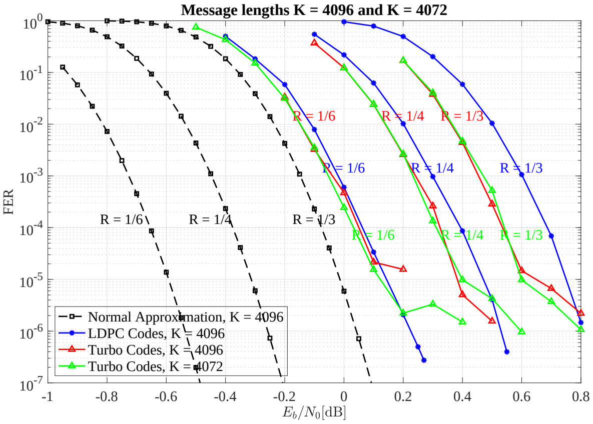

Figure 6. FER as a function of E_b/N_0 of LPDC Code with message length $K = 4096$ compared to normal approximation with $K = 4096$ and turbo codes with message lengths $K = 4096$ and $K = 4072$ for rates $R = 1/6, 1/4$ and $1/3$.

and p_q denotes one of the following eight prime integers

$$p_1 = 31; p_2 = 37; p_3 = 43; p_4 = 47;$$
$$p_5 = 53; p_6 = 59; p_7 = 61; p_8 = 67$$

The permutation numbers shall be interpreted such that the sth bit read out on line ’in b’ in Fig. 3 is the $\pi(s)$th bit of the input information block, as shown in Fig. 4.

For input block size of $K = 16384$ and rates R of $1/6,\ 1/4$ and $1/3$ FER performance of the turbo codes is provided in CCSDS [17]. To compare FER performance of the designed low-rate LDPC codes with CCSDS turbo codes for input block sizes $K = 1024$ and $K = 4096$ FER simulations for the turbo codes are performed. We also performed simulations for input block sizes of $K = 1016$ and $K = 4072$ to explore how input length can affect interleaver performance and error floor. For $K = 1016$ and $K = 4072$ we selected k_2 as the nearest prime number to $k_2 = K/8$ to see if a prime k_2 would improve the error floor performance. The resulting message block sizes K are 1016 and 4072, which are very close to the LDPC code input block sizes and sometimes provide better interleavers with lower error floors but now with a message length that is not a power of 2. In CCSDS turbo codes there are $4-$bit tails to terminate the trellis, so the actual code rates for turbo codes are $RK/(K+4)$ which are slightly lower than the ideal Rs of $1/6,\ 1/4,\ 1/3$ and $1/2$ achieved by the LDPC code.

Message Length $K = 16384$

Fig. 5 shows FER of the designed LDPC code compared to FER obtained with normal approximation and FER of the CCSDS turbo code, plotted as a function of E_b/N_0 for a code rates of $R = 1/6,\ 1/4,\ 1/3$ and a message length of $K = 16384$. In the waterfall region down to FER 10^{-3} or 10^{-4}, the rate-$1/6$ LDPC code outperforms the rate-$1/6$ turbo code by less than 0.1 dB. The rate-$1/4$ LDPC and turbo codes have almost identical waterfall performance, and at rate-$1/3$ the turbo code outperforms the LDPC code in the waterfall region by less than 0.1 dB

After the waterfall region, the LDPC code has much better performance for all three rates. Below 10^{-4} FER, the rate-$1/6$ LDPC code outperforms the rate-$1/6$ turbo code by a few orders of magnitude since turbo code has an error floor at 10^{-4}. The rate-$1/3$ and rate-$1/6$ turbo codes have error floors at FER of 10^{-4} and rate-$1/4$ at FER of 10^{-3}. The LDPC codes did not show any error floor at the simulated SNRs. The data for turbo codes FER curves in Fig. 5 is obtained from the CCSDS standard [17].

Message Lengths $K = 4096$, $K = 4072$

The CCSDS standard provides turbo code for message length $K = 3568$. In order to compare FER performance of the designed LDPC code rates to FER performance of the turbo code, we simulated turbo codes for message lengths $K = 4072$ and $K = 4096$ for rates $1/6,\ 1/4$ and $1/3$. Fig. 6 illustrates FER of the designed LDPC code for message length $K = 4096$ compared to normal approximation and FER of the turbo code for message lengths of $K = 4096$ and $K = 4072$, plotted as a function of E_b/N_0 for code rates $R = 1/6,\ 1/4,\ 1/3$.

For rates $1/4$ and $1/3$, the designed LDPC codes show worse waterfall-region FER performance compared to the turbo codes. For rate $1/3$, the turbo code outperforms the LDPC codes by almost $0.2\ dB$ in the waterfall region. However, turbo code is hindered with an error floor at FER of about 10^{-5} and therefore the LDPC codes perform significantly better for lower FERs. When using input block size of $K = 4072$, where the interleaver benefits from a prime k_2, the turbo code shows slightly lower error floor for rates 1/6 and 1/3.

Message Length $K = 1024$, $K = 1016$

The CCSDS standard provides turbo codes for message length $K = 1784$, which is the closest message length to $K = 1024$ for which we designed LDPC code. For the purposes of performance comparison, we simulated turbo code FER performance for message lengths $K = 1024$ and $K = 1016$, where the interleaver benefits from a prime k_2. Fig. 7 shows frame error rate of the designed LDPC codes, CCSDS turbo codes, and normal approximation as a function of E_b/N_0 for rates $R = 1/6,\ 1/4$, and $1/3$ and for message length $K = 1024$. For message length $K = 1024$ the CCSDS turbo code suffers from a high error floor at FER of 10^{-3}.

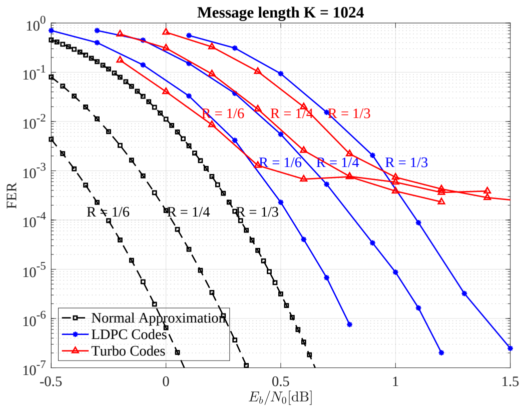

Figure 7. FER as a function of E_b/N_0 of LPDC Code compared to turbo codes and normal approximation for rates $R = 1/6, 1/4, 1/3$ and message length $K = 1024$.

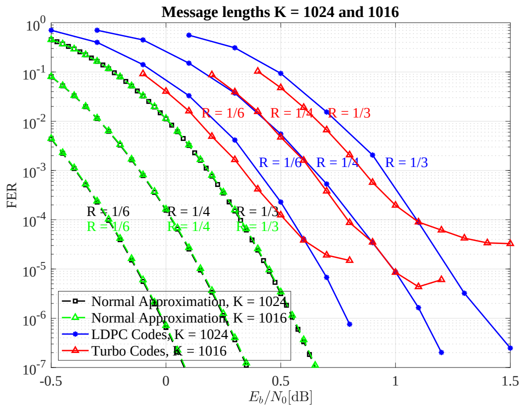

Figure 8. FER as a function of E_b/N_0 of LPDC Code with message length $K = 1024$ compared to turbo codes with message length $K = 1016$ for rates $R = 1/6, 1/4, 1/3$.

Fig. 8 shows FER as a function of E_b/N_0 for the designed LDPC codes and normal approximation with message length $K = 1024$ and CCSDS turbo codes with message length $K = 1016$, where the interleaver benefits from a prime k_2, for rates $R = 1/6$, $1/4$, and $1/3$. For $K = 1016$ turbo code shows a lower error floor at FER of 10^{-4} because of a prime k_2. Figs. 7 and 8 emphasize the benefit of the LDPC approach at low FERs. The LDPC code outperforms the turbo code FER by orders of magnitude. Note that LDPC code has a steep FER curve and no error floor for simulated data points down to FER of 10^{-7}.

Fig. 9 compares the FER vs. E_b/N_0 performance of the rate-$1/2$ CCSDS LDPC code with input block size of $K = 1024$ to the rate-$1/2$ CCSDS turbo codes with input block sizes of $K = 1024$ and $K = 1016$. The CCSDS turbo codes have error floors at FERs of 10^{-3} and 10^{-4} respectively for $K = 1024$ and $K = 1016$. The CCSDS rate-$1/2$ LDPC code, simulated down to FER of 10^{-7}, shows steep curve without error floor implying orders of magnitude superior performance when compared to the rate-$1/2$ CCSDS turbo codes.

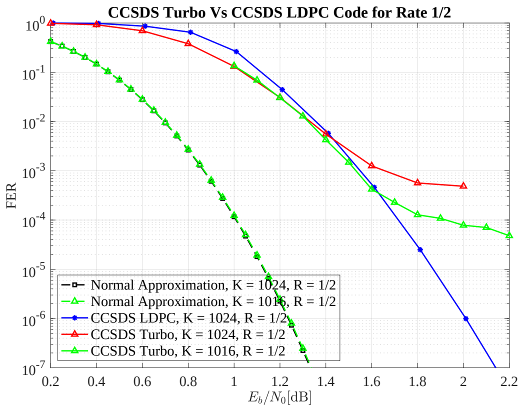

Figure 9. FER as a function of E_b/N_0 of the CCSDS LPDC Code with message length $K = 1024$ compared to the CCSDS turbo code and normal approximation with message lengths $K = 1016$ and $K = 1024$ for rate $R = 1/2$.

4. Conclusion

Deep space communication to Mars and other distant missions requires data transmission at low rates. For rates below $1/2$, the current CCSDS standard provides turbo codes which suffer from an error floor phenomenon where the FER does not improve significantly despite increasing the transmission power beyond a certain threshold. To overcome this error floor limitation, we developed new LDPC codes with rates $R = 1/3$, $1/4$, and $1/6$ while maintaining the original rate-$1/2$ CCSDS LDPC code [19] as a part of the new code.

Simulations show that the FER of LDPC codes outperforms current CCSDS turbo codes in the error floor region of the turbo codes by orders of magnitude. The turbo codes have error floors between 10^{-3} and 10^{-5} while the newly designed LDPC codes had no error floor visible in our simulations. The newly designed LDPC codes performed similarly to the turbo codes in the waterfall region, generally within 0.1 dB better or worse that the turbo codes. The most waterfall performance loss was with the $K = 4096$ LDPC code, which was almost 0.2 dB worse that the turbo code in the waterfall region.

In summary, the turbo codes provided in the CCSDS standard have an error floor between FERs of 10^{-3} and 10^{-5}. Since the published CCSDS performance results do not provide enough data to observe the error floor for all the rates/block lengths, we simulated CCSDS turbo codes for rates $1/2$, $1/3$, $1/4$ and $1/6$ and message lengths $K = 1016, 1024, 4072$ and $K = 4096$ to provide more robust comparison with our LDPC codes. The simulated results confirm the existence of error floor of turbo codes at about FER between 10^{-3} or 10^{-5} and show that the newly designed LDPC codes do not display this error floor behavior.

Acknowledgments

This work was carried out in part at the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration. The authors would like to thank the JPL Center of Innovation Fund (CIF) for its support. UCLA also would like to acknowledge National Science Foundation (NSF) grant CCF-1911166.

REFERENCES

[1] R. G. Gallager, "Low-Density Parity-Check Codes," *Cambridge, MA: MIT Press, 1963*, 1963.

[2] R. Tanner, "A recursive approach to low complexity codes," *IEEE Transactions on Information Theory*, vol. 27, no. 5, pp. 533–547, 1981.

[3] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: turbo-codes," *IEEE Transactions on Communications*, vol. 44, no. 10, pp. 1261–1271, 1996.

[4] D. MacKay and R. Neal, "Good codes based on very sparse matrices," in *Cryptography and Coding, 5th IMA Conference*, C. Boyd, Ed. Berlin: Springer-Verlag, October 1995.

[5] D. MacKay, "Good error correcting codes based on very sparse matrices," *IEEE Transactions on Information Theory*, vol. 45, no. 3, pp. 399–431, March 1999.

[6] N. Alon and M. Luby, "A linear time erasure-resilient code with nearly optimal recovery," *IEEE Transactions on Information Theory*, vol. 42, no. 11, pp. 1732–1736, November 1996.

[7] J. Byers, M. Luby, M. Mitzenmacher, and A. Rege, "A digital fountain approach to reliable distribution of bulk data," *ACM SIGCOMM Computer Communication Review*, vol. 28, no. 4, October 1998.

[8] D. Divsalar, H. Jin, and R. McEliece, "Coding theorems for 'turbo-like' codes," in *Proceedings of the 1998 Allerton Conference*, 1998.

[9] H. Jin, A. Khandekar, and R. McEliece, "Irregular Repeat-Accumulate Codes," in *Proceedings of the 2nd International Symposium on Turbo Codes*, 2000.

[10] T. Richardson, M. Shokrollahi, and R. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 619–637, 2001.

[11] T. Richardson and R. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 599–618, 2001.

[12] J. Thorpe, "Low-Density Parity-Check (LDPC) Codes Constructed from Protographs,," *JPL IPN Progress Report 42-154*, August 15, 2003.

[13] A. Abbasfar, D. Divsalar, and K. Yao, "Accumulate-Repeat-Accumulate Codes," *IEEE Transactions on Communications*, vol. 55, no. 4, pp. 692–702, 2007.

[14] D. Divsalar, C. Jones, S. Dolinar, and J. Thorpe, "Protograph based ldpc codes with minimum distance linearly growing with block size," in *GLOBECOM '05. IEEE Global Telecommunications Conference, 2005.*, vol. 3, 2005, pp. 5 pp.–.

[15] T.-Y. Chen, D. Divsalar, J. Wang, and R. D. Wesel, "Protograph-Based Raptor-Like LDPC Codes for Rate Compatibility with Short Blocklengths," in *2011 IEEE Global Telecommunications Conference - GLOBECOM 2011*, 2011, pp. 1–6.

[16] T.-Y. Chen, K. Vakilinia, D. Divsalar, and R. D. Wesel, "Protograph-Based Raptor-Like LDPC Codes," *IEEE Transactions on Communications*, vol. 63, no. 5, pp. 1522–1532, 2015.

[17] "TM synchronization and Channel coding—summary of concept and rationale," *Report Concerning Space Data System Standards (Green Book). Informational Report CCSDS, CCSDS 131.1-G-3*, 2020.

[18] T. Tian, C. Jones, J. Villasenor, and R. Wesel, "Selective avoidance of cycles in irregular LDPC code construction," *IEEE Transactions on Communications*, vol. 52, no. 8, pp. 1242–1247, 2004.

[19] "Low density parity check codes for use in near-earth and deep space applications," *Experimental Specification, CCSDS 131.1-O-1. Orange Book, Issue 1.*, August 2006.

[20] Y. Polyanskiy, H. V. Poor, and S. Verdu, "Channel Coding Rate in the Finite Blocklength Regime," *IEEE Transactions on Information Theory*, vol. 56, no. 5, pp. 2307–2359, 2010.

[21] "TM Synchronization and Channel coding—summary of concept and rationale," *Blue Book*, 2023.

BIOGRAPHY

***Semira Galijasevic** (Student Member, IEEE) received her BS degree (Summa Cum Laude) and MS degree in Electrical Engineering from the University of California at Los Angeles (UCLA) in 2021 and 2023, respectively. She is currently pursuing a Ph.D. degree in Electrical Engineering at UCLA. Her research interests include information theory, channel coding theory, LDPC codes, adaptive coding in fading optical channels, coding for storage, and optimal modulation design.*

***Linfang Wang** received the M.Sc. degree in information and communication engineering from the Harbin Institute of Technology, Harbin, China, in 2018. This work was performed when he was pursuing the Ph.D. degree with the University of California at Los Angeles (UCLA). He received his Ph.D. degree in 2023. His current research interest lies in coding theory, especially for LDPC codes, trellis codes, and the application of deep learning/machine learning on channel coding.*

***Jon Hamkins** received the B.S. degree in electrical engineering from the California Institute of Technology (Caltech), Pasadena, in 1990 and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Illinois at Urbana- Champaign in 1993 and 1996, respectively. Since then, he has been with the Jet Propulsion Laboratory, Caltech, where he is now the Chief Technologist of the Communications, Tracking, and Radar Division. His research interests include channel coding theory, information theory, signal processing, autonomous receivers, ranging, and optical communications.*

Dariush Divsalar *received the Ph.D. degree in electrical engineering from UCLA, in 1978. Since then, he has been with the Jet Propulsion Laboratory (JPL), California Institute of Technology (Caltech), Pasadena, where he is a Fellow. At JPL, he has been involved with developing state-of-the-art technology for advanced deep-space communications systems and future NASA space exploration. Since 1986, he has taught graduate courses in communications and coding at UCLA and Caltech. He has published more than 290 papers, coauthored a book entitled* An Introduction to Trellis Coded Modulation with Applications*, contributed to three other books, and holds 30 U.S. patents. Dr. Divsalar was a co-recipient of the 1986 paper award of the IEEE Transactions on Vehicular Technology. He was also a co-recipient of the joint paper award of the IEEE Information Theory and IEEE Communication Theory societies in 2008. The IEEE Communication Society has selected one of his papers for inclusion in a book entitled The Best of the Best: Fifty Years of Communications and Networking Research. He served as an Editor for the IEEE Transactions on Communications from 1989 to 1996. A Fellow of IEEE since 1997. He has received over 50 NASA Tech Brief awards, a NASA Exceptional Engineering Achievement Medal in 1996, IEEE Alexander Graham Bell Medal in 2014, an Ellis Island Medal of Honor in 2023, and a NASA Distinguished Public Service Medal in 2023. He was elected to the National Academy of Engineering in 2024.*

Richard Wesel *received the B.S. and M.S. degrees in electrical engineering from the Massachusetts Institute of Technology in 1989 and the Ph.D. degree in electrical engineering from Stanford University in 1996. He is currently a Professor with the Electrical and Computer Engineering Department, Henry Samueli School of Engineering and Applied Science, UCLA, where he is also the Associate Dean of Academic and Student Affairs. His research interests include communication theory with a particular interests include short-blocklength communication with and without feedback, list decoding, low-density parity-check codes, and optimal modulation design. Dr. Wesel is a Fellow of the IEEE and has received the National Science Foundation CAREER Award, the Okawa Foundation Award for Research in Information Theory and Telecommunications, and the Excellence in Teaching Award from the Samueli School of Engineering. He has served as an Associate Editor for Coding and Coded Modulation and IEEE Transactions on Communications and as an Associate Editor for Coding and Decoding and IEEE Transactions on Information Theory.*