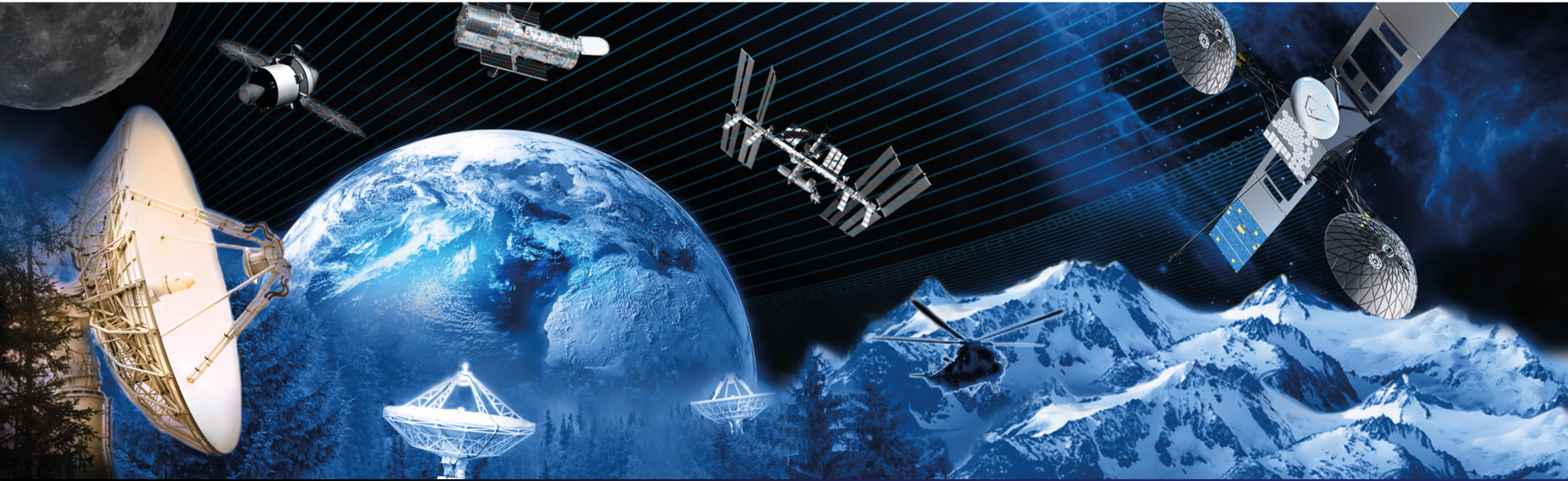
National Aeronautics and Space Administration

Exploration & SPACE Communications

More than you ever imagined...

Polar Coding for Forward Error Correction in Space Communications

By Naveed Naimipour, Haleh Safavi and Harry Shaw



Outline

1. Introduction and Background
2. Coding Schemes
   - Polar and Low-density parity-check (LDPC) Coding
3. Results
4. Discussion
5. Summary and Recommendations

Overview

Purpose: Exploring the applications of FEC codes in optical communications.

Mathematical verification and implementation of past, current, and future FEC methodologies for potential applications.

- Investigating the viability of polar codes for optical communications
- Analyzing the advantages and drawbacks of polar codes with respect to classical LDPC techniques
- Simulating various polar code techniques via MATLAB and C to test performance and accuracy

Why Forward Error Correction (FEC) Codes?

Rhee, I (2002). *US6421387B1*. North Carolina State University.

Background

Brief FEC Progression

1. Reed-Muller codes and convolutional codes
   - Common for the earlier days of space missions
2. Convolutional codes carried onto future missions and even included Viterbi decoders
   - Implemented on missions such as Voyager [1]
3. Concatenated convolutional and Reed-Soloman coding scheme
   - Ideal for deep-space missions requiring more powerful codes [2]
   - Variations of concatenated codes became common place for many missions
4. Turbo codes were seen as substantially more useful
   - Lower complexity and higher coding gains [3]
5. Low-density parity-check (LDPC) showed promise when implemented [3]
   - Flexibility in terms of degrees of freedom
   - Performance was even better after design inefficiencies were addressed
6. Polar codes have shown improved performance and immense promise
   - Lower complexity stemming from its block structure design and recursive nature

LDPC Coding Background

1. Introduced by Gallagher in the 1960s [4]
2. LDPC coding aims to implement a parity-check matrix
   - Main characteristics of the parity-check matrix
     - Sparse
     - Randomly generated
3. Not fully explored until the 1990s
   - Lower complexity
   - Ability to perform near the theoretically achievable coding gain
4. Developed to bypass the commonly used turbo codes for higher code rates.
   - Graphical representations included [5] [6]
5. Additional LDPC coding schemes (i.e. ones with Reed-Soloman outer coding) [7]
   - Improved performance
   - Better compatibility with lower code rates
6. We have evaluated promising techniques for LDPC codes in an optical environment
   - Tests for convolutional and no interleavers

Polar Coding Background

1. Described by Arikan in his 2009 work [8]
2. Have the ability to asymptotically achieve the Shannon capacity on many channels
3. Highly regular structure makes them particularly useful for real world applications
   - 3rd Generation Partnership Project's (3GPP) preliminary adoption of polar codes for Enhanced Mobile Broadband (eMBB) in 5G
4. Bypass the "error floors" that plague LDPC coding
   - Maintains low complexity
   - Maintains high performance
5. We have evaluated extremely promising techniques for polar codes in an optical environment
   - Cyclic redundancy checks (CRC)
   - Successive cancellation (SC)
   - Successive cancellation lists (SCL)

Coding Schemes

LDPC Coding

1. Implement a series of parity check equations based on binary parity check block codes
2. Symbols in the code satisfy *m* parity check equations with block codes that consist of binary vectors of fixed length *n*
3. Each codeword will contain:
    - *(n − m)* = *k* information digits
    - *m* check digits.
4. *m* x *n* sparse parity matrix, **H**, is created
5. A common way to implement a LDPC code is to invert **H** to obtain the generator matrix **G**.
    - Matrix multiplication can be utilized to encode.
    - Encoding complexity is quadratic in codeword length due to **G** typically being dense
6. Decoding typically done via quantized soft symbol detection at each node in the graph
    - log-likelihood ratios (LLR) for each variable node results in soft decisions

Polar Coding [8]

1. Binary-input discrete memoryless channel

$$W : \mathbb{X} \rightarrow \mathbb{Y} \quad (1)$$

2. Channel Transition Probabilities

$$W(y|x), x \in \mathbb{X} \text{ and } y \in \mathbb{Y} \quad (2)$$

3. Polarization effect of (3) is utilized to define the generator matrix (4)

$$\mathbf{G_1} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \quad (3) \qquad \mathbf{G_n} = \mathbf{G_1}^{\otimes n} \quad (4)$$

4. Remaining channels transmit “frozen bits”
5. Information bits carried on *K* most reliable polarized channels

$$W_N^{(j)}, j \in \mathbb{J} \quad (5)$$

6. For a binary source block, a code block can be mapped using *N-K* frozen bits using (6)

$$x_1^N = u_1^N \cdot \mathbf{G_n} \quad (6)$$

Coding Schemes

CRC

1. Typically add a fixed-length check value as their encoding scheme for messages
2. With *k* information bits and *m*-bit CRC sequence results in:
   - *K* = *k* + *m* bits for a *K* bit input block
3. Implementing the CRC in the source bits maintains the code rate as it is defined
4. Generator polynomials are typically needed
   - Act as the divisor in the long division operation
5. Parity bit is a basic example of CRC being used in error correction
   - Has a two term generator polynomial in *x + 1* and can be classified as a 1-bit CRC
6. Usually implemented with other FEC techniques, not solely by themselves

SC

1. Eliminates redundancies by cutting the polar codes into smaller pieces for processing
2. If the estimate of u_1^N be denoted as $\hat{u}_1^N$, then $\hat{u}_j^N$ can be found successively after y_1^N has been received using (7)

$$\hat{u}_j = \begin{cases} h_j(y_1^N, \hat{u}_1^{j-1}) & j \in \mathbb{J} \\ u_j & j \in \mathbb{F} \end{cases} \tag{7}$$

and the decision function is defined via the following:

$$h_j = \begin{cases} 0 & \text{if } \frac{W_N^{(j)}(y_1^N, \hat{u}_1^{j-1}|0)}{W_N^{(j)}(y_1^N, \hat{u}_1^{j-1}|1)} \geq 1 \\ 1 & \text{otherwise} \end{cases} \tag{8}$$

3. A decoder block error occurs if

$$\hat{u}_1^N \neq u_1^N \tag{9}$$

SCL

1. Improves the performance of SC decoding
   - SC struggles with small and medium length codes in polar coding
2. Keep a list of *L* survival code bits at each step
   - Done in place of a single survival path implemented by SC
3. If we let $\hat{u}_i$ be a random bit, the decoder makes *2L* candidates from the original *L*
   - Done by keeping both paths with $\hat{u}_i = 0$ and $\hat{u}_i = 1$
4. *L* can be used as a threshold to determine when it should discard the worst path
   - Improves performance of SC

Results

1. Classical LDPC codes have been implemented for many years
2. Previous work with focused on block interleavers,
3. Simulation Overview
    - Conventional DVB-S2 model with AWGN noise and LDPC coding per the new CCSDS protocols under consideration with convolutional interleavers [10]
    - Various interleaver sizes were tested as recommended in CCSDS for a standard message length size
    - Compared the performance under the best case scenario of no interleaver
4. Simulation Details
    - Convolutional interleaver size had a cap of 24, 000 binary digits for QPSK model
    - Anything larger would need advanced computing power
    - Fig. 1 shows the BER plots for two different interleaver sizes

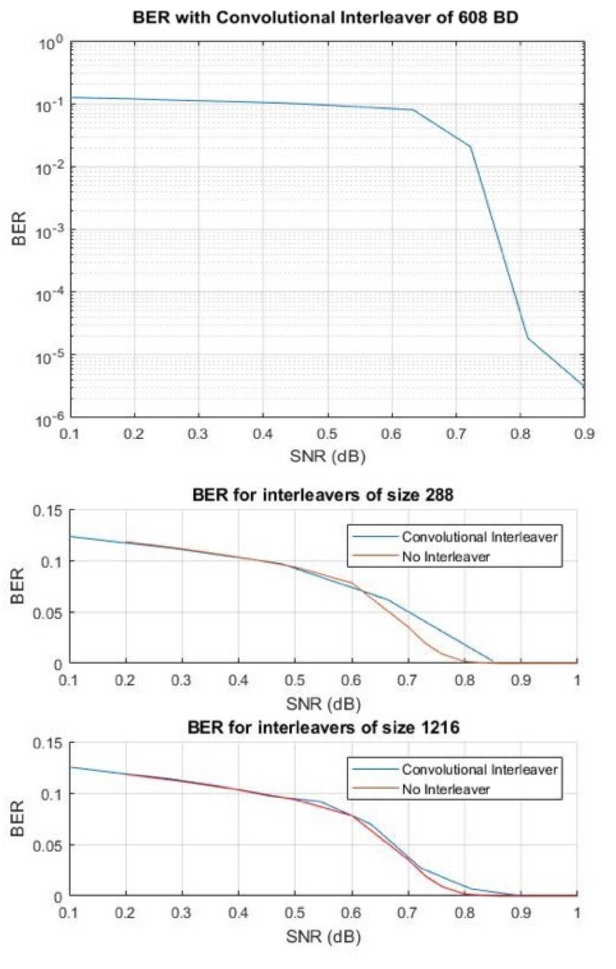


Fig. 1: BER Plots for LDPC Coding with Convolutional and No Interleaver

Results

1. SC and SCL Simulation Overview
   - Implemented a basic SC framework with different codeword lengths and observed the BER performance.
   - For that reason, we implemented a basic SCL framework using LLR to test its BER performance with different codeword lengths per CCSDS suggestions
2. SC and SCL Simulation Details
   - Simulations done on BPSK constellation
   - Tested on AWGN noise to signify the worst possible noisy scenario
   - Tested on burst noise to more accurately portray noise in space
   - BER results are shown in Fig. 2 for various message lengths of 256 bits and the two different types of noise.
3. CRC-Aided Polar codes were shown to beat turbo codes in terms of performance and decreased complexity [11]
4. CRC-Aided Polar Codes Simulation Details
   - Implemented a similar framework with CRC = 24 SCL decoding on a QPSK constellation
   - Implemented R=1/2 per CCSDS suggestions for all tests.
   - Primarily tested with AWGN noise to simulate the worst case scenario with relation to space applications
   - BER results are shown in Fig. 3 for various message lengths

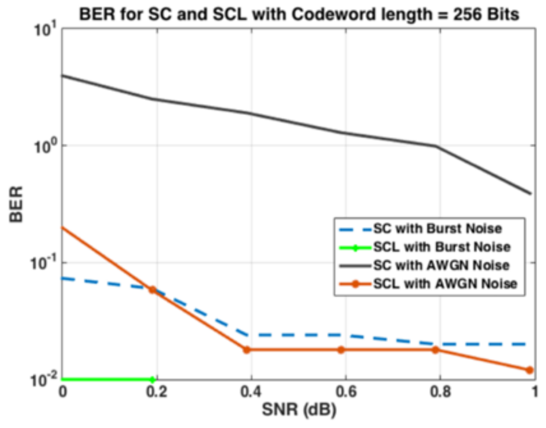


Fig. 2: BER plots for Polar coding with SC and SCL coding

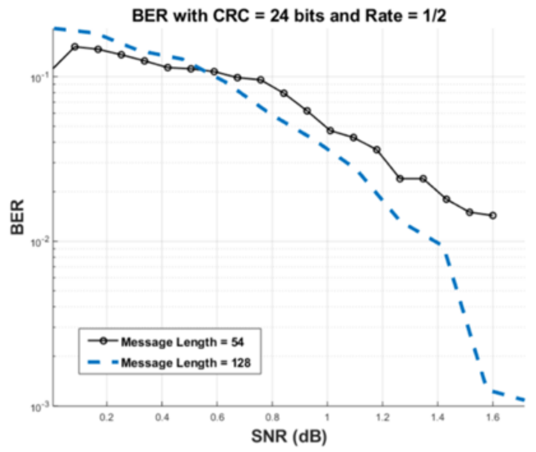


Fig. 3: BER plots for CRC-Aided Polar Coding

Discussion

1. Polar codes overcame the error floor that plague LDPC codes as expected, but polar coding typically did so at a higher SNR
2. We tested different sizes of data sets while maintaining code rate at CCSDS standards
   - Larger data sets implemented a sizable number of fixed length messages where the maximum message length was increased accordingly
     - Simulated data sets as large as MATLAB could handle (typically larger than 1024)
   - Smaller data sets involve a smaller number of fixed length messages with short messages
3. Polar coding processed data more efficiently, but had difficulty with accuracy when tested with larger data sets
   - Particularly true for shorter SC polar codes
4. SC polar codes underperformed with respect to CCSDS recommended LDPC codes with burst noise
5. SC nominally outperformed CCSDS recommended LDPC codes with AWGN noise.
6. SCL polar codes consistently outperformed SC polar codes
   - Showed promising performance when tested with burst noise regardless of code length
   - Still struggled to accurately process larger data volumes

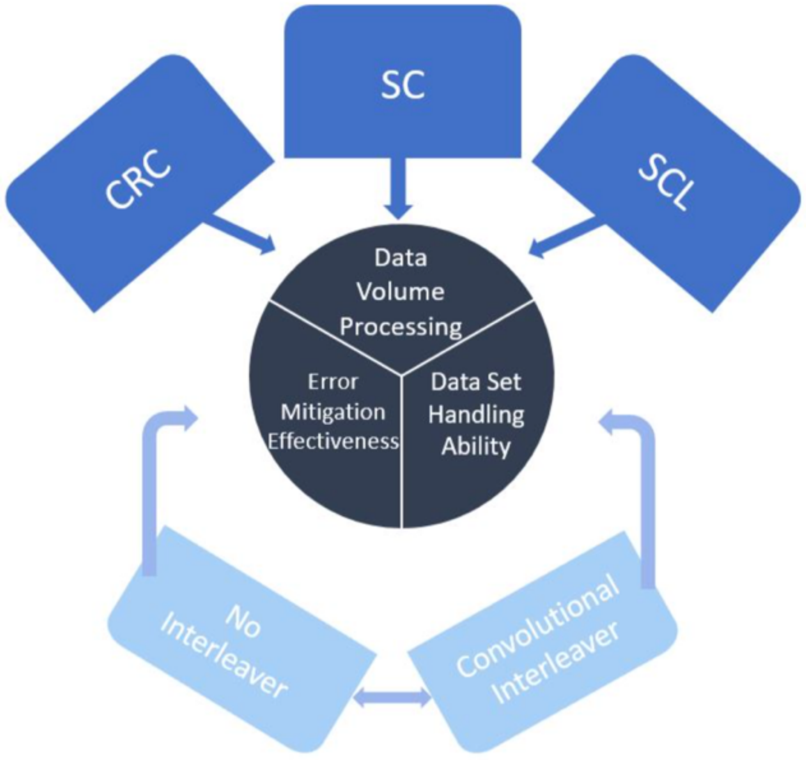


Fig. 4: Strengths of polar coding (top) and LDPC coding (bottom).

Discussion

1. CRC-Aided polar codes performed well when tested with all types of noise
   - However, larger power was necessary to obtain a meaningful BER when compared to LDPC codes
2. CRC-Aided codes were less accurate with larger data sets similar to SC and SCL codes
   - Fig. 2 shows its noticeable performance improvement with burst noise
     - Hence, promising for space applications
3. LDPC simulations exhibited that there was not a large BER improvement with a convolutional interleaver
   - Held true for binary digits ranging from 288 to 24,000
   - A lack of interleaver actually contributed to better BER performance for smaller binary digits such as 288
4. For codewords less than or equal to 1024 bits:
   - BER performance of the polar coding methodologies were comparable to LDPC coding
5. Polar coding methodologies were able to process data significantly faster than any type of LDPC coding
   - Most noticeable for any type of polar coding compared to LDPC codes with convolutional interleavers

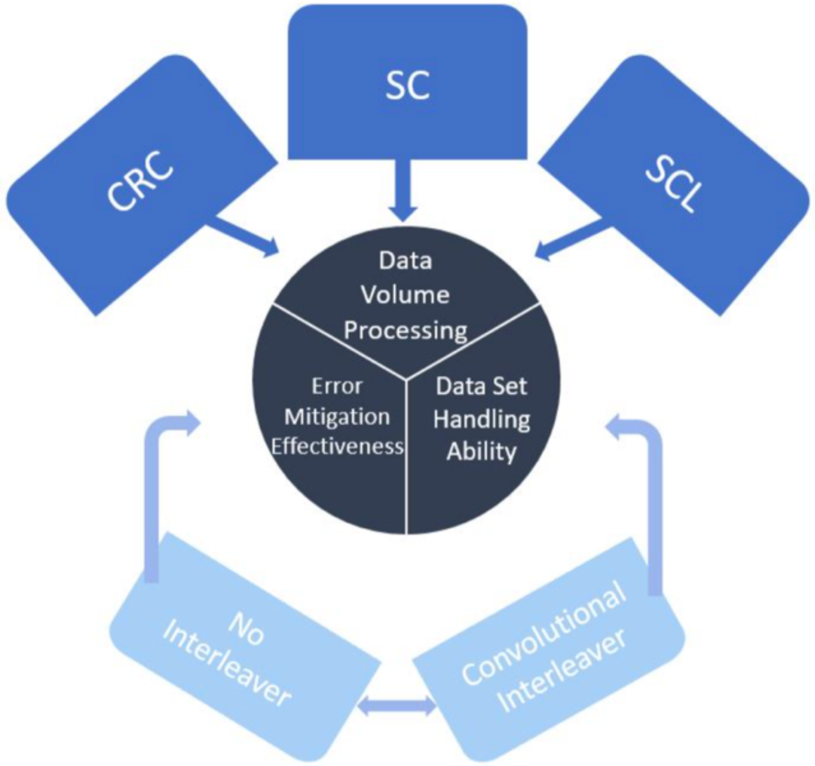
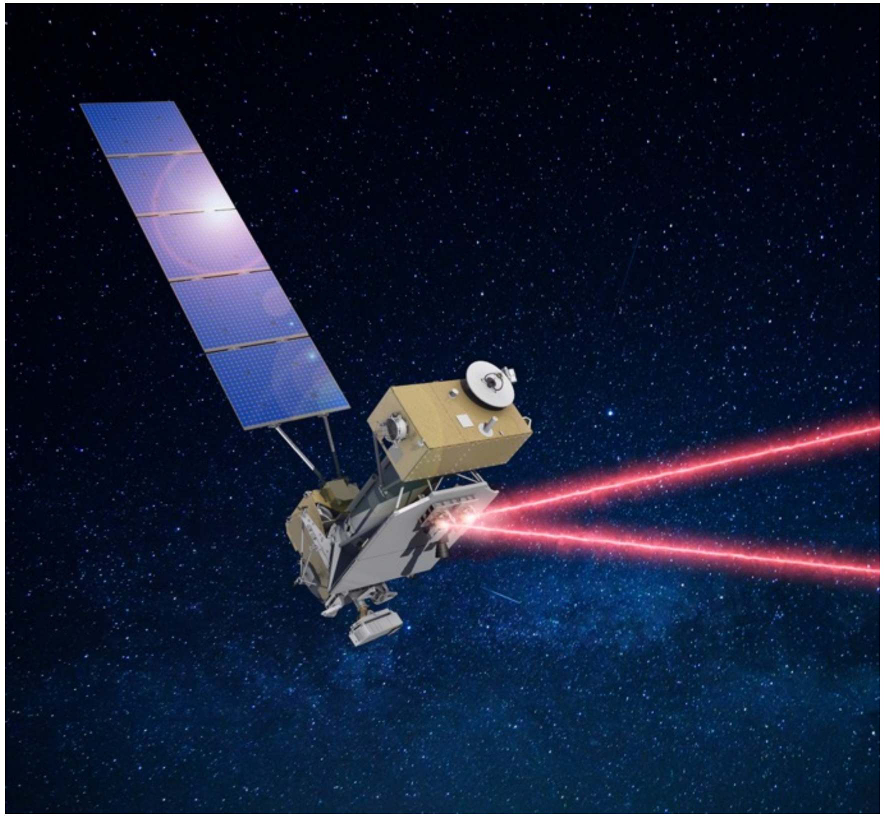


Fig. 4: Strengths of polar coding (top) and LDPC coding (bottom).

Recommendations and Summary

1. Our results can be directly applied to the current CCSDS standards being put in place
   - CCSDS protocols recommend codes of length *2048* and larger for optical communications, especially for GEO related missions
   - CCSDS protocols also suggest *7000* bytes and larger for the *1/2* rate, which is most commonly used in NASA missions
   - Based on these current frameworks, the polar coding methodologies mentioned in this paper would most likely not be a great fit for those space applications
2. Future work should involve the development of polar coding parameters to combat obstacles such as precision deficiencies
   - Current CCSDS standards do not take polar coding structures into account
   - This would greatly help in improving precision performance
3. Current missions that use RF communications and smaller slice lengths can benefit immensely from polar coding



THANK YOU!

References

[1] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Inf. Theory*, vol. IT-13, pp. 260-269, Apr. 1967.

[2] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *J. Soc. Ind. Math.*, vol. 8, pp. 260-269, Jun. 1960.

[3] K. S. Andrews, D. Divsalar, S. Dolinar, J. Hamkins, C. R. Jones and F. Pollara, "The Development of Turbo and LDPC Codes for Deep-Space Applications," in *Proceedings of the IEEE*, vol. 95, no. 11, pp. 2142-2156, Nov. 2007.

[4] R. Gallager, "Blow-density parity-check codes," Ph.D. dissertation, Massachusetts Inst. of Technol., Cambridge, Sep. 1960.

[5] D. Mackay, "Good error correcting codes based on very sparse matrices," *IEEE Trans. Inf. Theory*, vol. 45, pp. 399-431, Mar. 1999.

[6] K. Fagervik A. S. Larssen "Performance and complexity comparison of low density parity check codes and turbo codes" *Proc. Norwegian Signal Process. Symp*. pp. 2-4 Oct. 2003.

[7] Q. Diao, J. Li, S. Lin, and I. F. Blake, "New classes of partial geometries and their associated LDPC Codes," IEEE Trans. Inf. Theory, vol. 62, no.6, pp. 2947-2965, Jun., 2016.

[8] E. Arikan, "Channel Polarization: A Method for Constructing Capacity-Achieving Codes for Symmetric Binary-Input Memoryless Channels," in *IEEE Transactions on Information Theory*, vol. 55, no. 7, pp. 3051-3073, July 2009.

[9] I. Tal and A. Vardy, "List decoding of polar codes," I*EEE Transactions on Information Theory*, vol. 61, no. 5, pp. 2213-2226, May 2015.

[10] CCSDS 000.0-O-0, "OPTICAL HIGH DATA RATE (HDR) COMMUNICATION," Draft Orange Book. April 2017.