lend themselves to computationally efficient structures that can be implemented in high-speed encoder hardware. However, high-speed encoder implementation can be expected to be a subject of future research.

This work was done by Dariush Divsalar, Christopher Jones, Samuel Dolinar, and Jeremy Thorpe of Caltech for NASA's Jet Propulsion Laboratory.

In accordance with Public Law 96-517, the contractor has elected to retain title to this invention. Inquiries concerning rights for its commercial use should be addressed to:

Innovative Technology Assets Management
JPL
Mail Stop 202-233
4800 Oak Grove Drive
Pasadena, CA 91109-8099
(818) 354-2240
E-mail: iaoffice@jpl.nasa.gov

Refer to NPO-42063, volume and number of this NASA Tech Briefs *issue, and the page number.*

Constructing LDPC Codes From Loop-Free Encoding Modules

High-speed iterative decoders can readily be implemented in hardware.

NASA's Jet Propulsion Laboratory, Pasadena, California

A method of constructing certain low-density parity-check (LDPC) codes by use of relatively simple loop-free coding modules has been developed. The subclasses of LDPC codes to which the method applies includes accumulate-repeat-accumulate (ARA) codes, accumulate-repeat-check-accumulate codes, and the codes described in "Accumulate-Repeat-Accumulate-Accumulate Codes" (NPO-41305), *NASA Tech Briefs*, Vol. 31, No. 9 (September 2007), page 90. All of the affected codes can be characterized as serial/parallel (hybrid) concatenations of such relatively simple modules as accumulators, repetition codes, differentiators, and punctured single-parity check codes. These are error-correcting codes suitable for use in a variety of wireless data-communication systems that include noisy channels. These codes can also be characterized as hybrid turbolike codes that have projected graph or protograph representations (for example see figure); these characteristics make it possible to design high-speed iterative decoders that utilize belief-propagation algorithms.

The present method comprises two related submethods for constructing LDPC codes from simple loop-free modules with circulant permutations. The first submethod is an iterative encoding method based on the erasure-decoding algorithm. The computations required by this method are well organized because they involve a parity-check matrix having a block-circulant structure.

The second submethod involves the use of block-circulant generator matrices. The encoders of this method are very similar to those of recursive convolutional codes. Some encoders according to this second submethod have been implemented in a small field-programmable gate array that operates at a speed of 100 megasymbols per second.

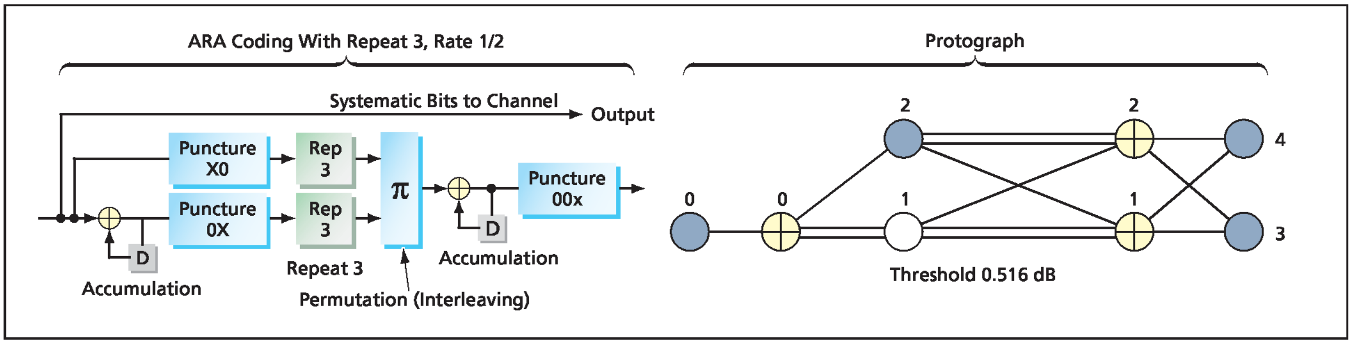
By use of density evolution (a computational-simulation technique for analyzing performances of LDPC codes), it has been shown through some examples that as the block size goes to infinity, low iterative decoding thresholds close to channel capacity limits can be achieved for the codes of the type in question having low maximum variable node degrees. The decoding thresholds in these examples are lower than those of the best-known unstructured irregular LDPC codes constrained to have the same maximum node degrees. Furthermore, the present method enables the construction of codes of any desired rate with thresholds that stay uniformly close to their respective channel capacity thresholds.

This work was done by Dariush Divsalar, Samuel Dolinar, Christopher Jones, Jeremy Thorpe, and Kenneth Andrews of Caltech for NASA's Jet Propulsion Laboratory.

In accordance with Public Law 96-517, the contractor has elected to retain title to this invention. Inquiries concerning rights for its commercial use should be addressed to:

Innovative Technology Assets Management
JPL
Mail Stop 202-233
4800 Oak Grove Drive
Pasadena, CA 91109-8099
(818) 354-2240
E-mail: iaoffice@jpl.nasa.gov

Refer to NPO-42042, volume and number of this NASA Tech Briefs *issue, and the page number.*



A Simple Rate-1/2 ARA Code is depicted here with its protograph representation as an example of codes to which the present method applies. An encoder for this code includes a precoder in the form of a punctured accumulator.