

FILED

January 22, 2024

CLERK, U.S. DISTRICT COURT
WESTERN DISTRICT OF TEXAS

**UNITED STATES DISTRICT COURT
WESTERN DISTRICT OF TEXAS
WACO DIVISION**

BY: CAV
DEPUTY

The CALIFORNIA INSTITUTE OF
TECHNOLOGY,

Plaintiff,

v.

DELL TECHNOLOGIES INC. and DELL
INC.,

Defendants.

§
§
§
§
§
§
§
§
§
§

Civil Action No.: 6:20-cv-1042

JURY TRIAL DEMANDED

SECOND AMENDED COMPLAINT FOR PATENT INFRINGEMENT

Plaintiff the California Institute of Technology (“Caltech” or “Plaintiff”), by and through its undersigned counsel, complains and alleges against Dell Technologies Inc. and Dell Inc. (collectively “Dell” or “Defendants”) as follows:

NATURE OF THE ACTION

1. This is a civil action for infringement of U.S. Patent No. 7,116,710 (the “’710 patent”), U.S. Patent No. 7,421,032 (the “’032 patent”), U.S. Patent No. 7,916,781 (the “’781 patent”), and U.S. Patent No. 8,284,833 (the “’833 patent”) (collectively, “the Asserted Patents”) arising under the patent laws of the United States, 35 U.S.C. §§ 1 et seq.

2. In January of 2020, a jury found that Apple Inc.’s (“Apple’s”) and Broadcom Limited’s (“Broadcom’s”) Wi-Fi products infringed the ’710, ’032, and ’781 patents and awarded Caltech over \$1.1 billion in damages. *Caltech v. Broadcom Limited, et al.*, No. 16-cv-3714-GW, Dkt. No. 2114 (C.D. Cal. Jan. 29, 2020). The Court of Appeals for the Federal Circuit affirmed the findings of the lower court that the Asserted Patents are valid, and that Apple and Broadcom infringed those patents. *Caltech v. Broadcom Limited, et al.*, 25 F.4th 976 (Fed. Cir. 2022). It also remanded the case for a further jury trial to determine damages arising from this infringement. *Id.* As in the case against Apple and Broadcom, Caltech seeks a reasonable royalty from Dell as

compensation for its infringement of the '710, '032, and '781 patents. Caltech also seeks a reasonable royalty from Dell as compensation for its infringement of the '833 patent.

THE PARTIES

3. Caltech is a non-profit private university organized under the laws of the State of California, with its principal place of business at 1200 East California Boulevard, Pasadena, California 91125.

4. Caltech is a world-renowned science and engineering research and education institution, where extraordinary faculty and students seek answers to complex questions, discover new knowledge, lead innovation, and transform our future. To date, 40 Caltech alumni and faculty have won a total of 41 Nobel Prizes. The mission of Caltech is to expand human knowledge and benefit society through research integrated with education. Caltech investigates the most challenging, fundamental problems in science and technology in a singularly collegial, interdisciplinary atmosphere, while educating outstanding students to become creative members of society. Caltech's investment in research has led Caltech to have more inventions disclosed and patents granted per faculty member than any other university in the nation, and to be consistently ranked as having one of the top university patent portfolios in strength and number of patents issued.

5. On information and belief, Dell Technologies Inc. is a Delaware corporation with its principal place of business at One Dell Way, Round Rock, Texas 78682.

6. On information and belief, Dell Inc. is a Delaware corporation with its principal place of business at One Dell Way, Round Rock, Texas 78682. Dell, Inc. has additional offices at 1404 Park Center Dr., Austin, Texas, 701 E. Parmer Lane, Bldg. PS2, Austin, Texas, 12500 Tech Ridge Road, Austin, Texas, 9715 Burnet Road, Austin, Texas, and 4309 Emma Browning Avenue, Austin, Texas.

JURISDICTION AND VENUE

7. This Court has jurisdiction over the subject matter of this action under 28 U.S.C. §§ 1331 and 1338(a).

8. This Court has personal jurisdiction over Dell pursuant to due process and/or the Texas Long Arm Statute because Dell has committed and continues to commit acts of patent

infringement, including acts giving rise to this action, within the State of Texas and this District, and because Dell recruits Texas residents, directly or through an intermediary located in this state, for employment inside or outside this state. The Court's exercise of jurisdiction over Dell would not offend traditional notions of fair play and substantial justice because Dell has established minimum contacts with the forum.

9. Venue is proper in this judicial district pursuant to 28 U.S.C. §§ 1391 and 1400 because a substantial part of the events or omissions giving rise to the claims occurred in this District, and Dell has committed acts of infringement and has a regular and established place of business in this District.

10. Dell has committed acts of infringement in this District, directly and/or through intermediaries, by, among other things, making, using, offering to sell, selling, and/or importing products and/or services that infringe the Asserted Patents, as alleged herein.

11. Dell has a regular and established places of business in this District including a shared corporate office at One Dell Way, Round Rock, Texas 78682. Dell is also registered to do business in Texas.

CALTECH'S ASSERTED PATENTS

12. On October 3, 2006, the United States Patent Office issued U.S. Patent No. 7,116,710, titled "Serial Concatenation of Interleaved Convolutional Codes Forming Turbo-Like Codes." A true and correct copy of the '710 patent is attached hereto as **Exhibit A**.

13. On September 2, 2008, the United States Patent Office issued U.S. Patent No. 7,421,032, titled "Serial Concatenation of Interleaved Convolutional Codes Forming Turbo-Like Codes." A true and correct copy of the '032 patent is attached hereto as **Exhibit B**. The '032 patent is a continuation of the application that led to the '710 patent.

14. On March 29, 2011, the United States Patent Office issued U.S. Patent No. 7,916,781, titled "Serial Concatenation of Interleaved Convolutional Codes Forming Turbo-Like Codes." A true and correct copy of the '781 patent is attached hereto as **Exhibit C**. The '781 patent is a continuation of the application that led to the '032 patent, which is a continuation of the application that led to the '710 patent.

15. On October 9, 2012, the United States Patent Office issued U.S. Patent No. 8,284,833, titled “Serial Concatenation of Interleaved Convolutional Codes Forming Turbo-Like Codes.” A true and correct copy of the ’833 patent is attached hereto as **Exhibit D**. The ’833 patent is a continuation of the application that led to the ’781 patent, which is a continuation of the application that led to the ’032 patent, which is a continuation of the application that led to the ’710 patent.

16. The ’710, ’032, ’781, and ’833 patents identify Hui Jin, Aamod Khandekar, and Robert J. McEliece as the inventors.

17. Caltech is the owner of all right, title, and interest in and to each of the Asserted Patents with full and exclusive right to bring suit to enforce the Asserted Patents, including the right to recover for past damages and/or royalties prior to the expiration of the ’710, ’032, ’781, and ’833 patents.

18. The Asserted Patents are valid and enforceable.

BACKGROUND

Caltech’s IRA Codes Patents

19. The ’710, ’032, ’781, and ’833 patents (“IRA Patents”) disclose seminal improvements to coding systems and methods. The IRA Patents introduce a new class of error correction codes called “irregular repeat and accumulate codes” (or “IRA codes”). The claimed methods and apparatuses in the IRA Patents are directed to encoders and decoders. For example, the claimed encoders in the IRA Patents generate an IRA “codeword” from message or information bits by reordering irregularly repeated instances of those bits in a randomized but known way and performing other logical operations such as summing and accumulating bits. The claimed decoders in the IRA Patents facilitate recovery of the message or information bits from the codewords even when the codewords have been corrupted by noise such as the noise that is experienced when transmitting a codeword over a wireless communications channel. These IRA codes are at least as effective at correcting errors in transmissions as prior coding techniques such as turbo codes, but use simpler encoding and decoding circuitry and provide other technical and practical advantages,

allowing for improved transmission rates and performance. Indeed, the IRA codes disclosed in the IRA Patents enable a transmission rate close to the theoretical limit.

20. The IRA Patents implement these novel IRA codes using novel encoders and decoders. The claims in the IRA Patents enable a person of ordinary skill in the art to implement IRA codes using simple circuitry, providing improved performance over prior art encoders and decoders.

21. In September 2000, the inventors of the IRA Patents published a paper regarding their invention, titled “Irregular Repeat-Accumulate Codes” for the Second International Conference on Turbo Codes (attached hereto as **Exhibit E**). This paper has been widely cited by experts in the field.

22. The IRA Patents and publications describing IRA codes have been widely recognized and cited by academics and experts in the field of digital communications for their improvements over prior art error correction codes. For example, a paper by Aline Roumy, Souad Guemghar, Giuseppe Caire, and Sergio Verdú praising these IRA codes was published in August 2004 in the IEEE Transactions on Information Theory. This paper, titled “Design Methods for Irregular Repeat-Accumulate Codes,” and attached hereto as **Exhibit F**, states:

IRA codes are, in fact, special subclasses of both irregular LDPCs and irregular turbo codes. . . . IRA codes are an appealing choice because the encoder is extremely simple, their performance is quite competitive with that of turbo codes and LDPCs, and they can be decoded with a very-low-complexity iterative decoding scheme.

This paper also notes that, four years after publication of the September 2000 paper, the inventors of the IRA Patents were the only ones to propose a method to design IRA codes.

IEEE 802.11 Wi-Fi Standard

23. The Institute of Electrical and Electronics Engineers (“IEEE”) has developed standards for wireless communications over local area networks (also referred to as “Wi-Fi”). Wi-Fi usage is widespread in modern electronic products, including smartphones, laptops, routers, televisions, cameras, cars, and other devices that have wireless connections.

24. The IEEE standard upon which Wi-Fi is based is IEEE 802.11. The 802.11 standardization process began in the 1990s and the first version of 802.11 was referred to as IEEE 802.11-1997. In the following years, subsequent versions of the 802.11 standard were adopted.

25. One of the key improvements to the 802.11n version of the standard involved a “High Throughput (HT)” mode that is implemented using specific LDPC (Low-Density Parity Check) error correction codes. The same LDPC error correction codes introduced in the 802.11n version of the standard are also implemented in the subsequent 802.11ac version (finalized by IEEE in 2013 and providing the basis for Wi-Fi 5) and 802.11ax version (nearing finalization and providing the basis for Wi-Fi 6) of the standard. The LDPC codes specified by the 802.11n, 802.11ac, and 802.11ax standards may be implemented using Caltech’s patented IRA/LDPC encoder and decoder technology.

Caltech’s Case Against Apple and Broadcom

26. In May 2016, Caltech filed a patent infringement action against Apple and Broadcom in the Central District of California involving the ’710, ’032, ’781, and ’833 patents. On January 29, 2020, a jury rendered a verdict finding that Apple’s and Broadcom’s Wi-Fi products infringed the ’710, ’032, and ’781 Patents and awarded Caltech over \$1.1 billion in damages. *Caltech v. Broadcom et al.*, No. 16-cv-3714-GW, Dkt. No. 2114 (C.D. Cal. Jan. 29, 2020).

27. The trial followed over three years of litigation during which the court dismissed the vast majority of Apple’s and Broadcom’s defenses and counter-claims. For example, the court denied Apple’s and Broadcom’s motion for summary judgment seeking to invalidate Caltech’s ’781 Patent under 35 U.S.C. § 101, and granted Caltech’s motion for summary judgment of validity of Caltech’s ’710 and ’032 Patents under 35 U.S.C. § 101. The court also denied Apple and Broadcom’s motions for summary judgment of non-infringement.

28. In addition, Apple filed ten *inter partes* review (“IPRs”) petitions with the United States Patent and Trademark Office’s Patent Trial and Appeal Board (“PTAB”) seeking to invalidate the ’710, ’032, ’781, and ’833 patents, and the PTAB either denied institution or upheld the patentability of claims in all ten petitions.

Dell

29. Dell manufactures, uses, imports, offers for sale, and/or sells Wi-Fi products that incorporate encoders and/or decoders claimed in the Asserted Patents (“Accused Products”). The Accused Products include, but are not limited to, laptops (e.g., Latitude, Vostro, Inspiron, XPS, G-Series, Rugged, Chromebook Enterprise, Education, and Alienware), desktops and all-in-ones (e.g., OptiPlex, Precision, Vostro, Inspiron, and XPS), tablets and 2-in-1s (e.g., XPS, Latitude, Inspiron, Rugged, Chromebook Enterprise, and Education), workstations (e.g., Precision), and thin clients. Upon information and belief, the Accused Products are compliant with the 802.11n, 802.11ac, and/or 802.11ax standards and the LDPC codes defined in those standards.

COUNT I

Infringement of the ’710 Patent

30. Caltech re-alleges and incorporates by reference the allegations of the preceding paragraphs of this Complaint as if fully set forth herein.

31. In violation of 35 U.S.C. § 271(a), Dell has infringed the ’710 patent by making, using, selling, offering for sale, and/or importing into the United States, without authority, the Accused Products which practice each and every limitation of at least claim 20 of the ’710 patent. Dell has infringed literally and/or under the doctrine of equivalents.

32. Upon information and belief, the Accused Products comply with the 802.11n, 802.11ac, and/or 802.11ax standards and the 12 LDPC error correction codes defined in those standards. In addition, upon information and belief, the Accused Products are implemented in a manner that not only complies with the 802.11n, 802.11ac, and/or 802.11ax standards, but also infringes the ’710 patent. This is because implementations of the 802.11n, 802.11ac, and/or 802.11ax standards that infringe the ’710 patent perform substantially fewer computations, have substantially more efficient circuitry, use less memory, consume less semiconductor die area, consume less power, and are otherwise more efficient and cost effective than implementations that do not infringe the ’710 patent.

33. The 12 LDPC codes were originally defined in the 802.11n version of the standard and include three 1/2 rate, three 2/3 rate, three 3/4 rate, and three 5/6 rate LDPC codes as shown in Table 20-14 of the standard below.¹

Table 20-14—LDPC parameters

Coding rate (R)	LDPC information block length (bits)	LDPC codeword block length (bits)
1/2	972	1944
1/2	648	1296
1/2	324	648
2/3	1296	1944
2/3	864	1296
2/3	432	648
3/4	1458	1944
3/4	972	1296
3/4	486	648
5/6	1620	1944
5/6	1080	1296
5/6	540	648

34. On information and belief, the Accused Products encode information or message bits using an LDPC encoder that supports the 12 LDPC codes defined in the standards. The LDPC encoder encodes the information or message bits to generate a codeword as described in Section 20.3.11.6.3 of the 802.11n standard shown below:²

¹ See IEEE 802.11n-2009 at § 20.3.11.6.2 (emphasis added); see also 802.11-2012 at § 20.3.11.7.2.

² See IEEE 802.11n-2009 at § 20.3.11.6.3(emphasis added); see also IEEE 802.11-2012 at § 20.3.11.7.3.

20.3.11.6.3 LDPC encoder

For each of the three available codeword block lengths, the LDPC encoder supports rate 1/2, rate 2/3, rate 3/4, and rate 5/6 encoding. The LDPC encoder is systematic, i.e., it encodes an information block, $\mathbf{c}=(i_0, i_1, \dots, i_{(k-1)})$, of size k , into a codeword, \mathbf{c} , of size n , $\mathbf{c}=(i_0, i_1, \dots, i_{(k-1)}, p_0, p_1, \dots, p_{(n-k-1)})$, by adding $n-k$ parity bits obtained so that $\mathbf{H} \times \mathbf{c}^T = \mathbf{0}$, where \mathbf{H} is an $(n-k) \times n$ parity-check matrix. The selection of the codeword block length (n) is achieved via the LDPC PDU encoding process described in 20.3.11.6.5.

35. On information and belief, the LDPC encoders in the Accused Products encode information or message bits in accordance with the 12 parity-check matrices defined in the 802.11n standard. A parity-check matrix \mathbf{H} for each of the 12 block sizes and code rates is defined in Tables R.1 to R.3 of the 802.11n. The parity-check matrix for one of the 12 LDPC codes is shown below.³

Table R.1 defines the matrix prototypes of the parity-check matrices for a codeword block length $n=648$ bits, with a subblock size $Z=27$ bits.

Table R.1—Matrix prototypes for codeword block length $n=648$ bits, subblock size is $Z = 27$ bits

* * *

(c) Coding rate $R = 3/4$.																							
16	17	22	24	9	3	14	-	4	2	7	-	26	-	2	-	21	-	1	0	-	-	-	-
25	12	12	3	3	26	6	21	-	15	22	-	15	-	4	-	-	16	-	0	0	-	-	-
25	18	26	16	22	23	9	-	0	-	4	-	4	-	8	23	11	-	-	-	0	0	-	-
9	7	0	1	17	-	-	7	3	-	3	23	-	16	-	-	21	-	0	-	-	0	0	-
24	5	26	7	1	-	-	15	24	15	-	8	-	13	-	13	-	11	-	-	-	-	0	0
2	2	19	14	24	1	15	19	-	21	-	2	-	24	-	3	-	2	1	-	-	-	-	0

36. Each parity-check matrix includes a left-hand side that corresponds to information or message bits, and a right-hand side that corresponds to parity bits. In the parity-check matrix shown above, the left-hand side that corresponds to information or message bits includes columns 1-18, and the right-hand side that corresponds to the parity bits includes columns 19-24. The left-hand side is structured in a way that corresponds to the use of irregular repetition, scrambling and summing in the encoding process, while the right-hand side is structured in a way that corresponds to using accumulation in the encoding process. Further, the left-hand side is structured in a way

³ See IEEE 802.11n-2009 at Annex R, Table R.1; see also IEEE 802.11-2012 at Annex F, Table F-1.

that corresponds to the use of a low-density generator matrix for performing operations of irregular repetition, scrambling and summing.

37. On information and belief, the LDPC encoders in the Accused Products are implemented in a manner that meets each and every limitation of claim 20 of the '710 patent. This is because implementations of the 802.11n, 802.11ac, and/or 802.11ax standards that infringe claim 20 of the '710 patent perform substantially fewer computations, have substantially more efficient circuitry, use less memory, consume less semiconductor die area, consume less power and are otherwise more efficient and cost effective than implementations that do not infringe this claim. The LDPC encoders in the Accused Products are coders. The LDPC encoders in the Accused Products include first coders which are low-density generator matrix coders and correspond to the left-hand sides of the parity-check matrices. The first coders have an input configured to receive a stream of bits (e.g., information or message bits). The first coders repeat the stream of bits irregularly and scramble the repeated bits. The irregular repetition and scrambling that occurs in the LDPC encoders in the Accused Products corresponds to the irregular repetition and scrambling depicted in the left-hand sides of the parity-check matrices.

38. On information and belief, the LDPC encoders in the Accused Products include second coders which correspond to the right-hand sides of the parity-check matrices. The second coders encode bits output from the first coder at a rate within 10% of one. The encoding of output bits at a rate within 10% of one that occurs in the LDPC encoders in the Accused Products corresponds to the accumulation depicted in the right-hand sides of the parity-check matrices.

39. Dell is not licensed or otherwise authorized to practice the claims of the '710 patent.

40. By reason of Dell's infringement, Caltech has suffered substantial damages.

41. Caltech is entitled to recover the damages sustained as a result of Dell's wrongful acts in an amount subject to proof at trial.

42. Caltech has complied with the requirements of 35 U.S.C. § 287(a) at least because neither Caltech nor any party that has held a license to the '710 patent have made, offered for sale, or sold any products in the United States subject to the marking requirements of 35 U.S.C. § 287(a).

43. Dell's infringement of the '710 patent is exceptional and entitles Caltech to attorneys' fees and costs incurred in prosecuting this action under 35 U.S.C. § 285.

COUNT II

Infringement of the '032 Patent

44. Caltech re-alleges and incorporates by reference the allegations of the preceding paragraphs of this Complaint as if fully set forth herein.

45. In violation of 35 U.S.C. § 271(a), Dell has infringed the '032 patent by making, using, selling, offering for sale, and/or importing into the United States, without authority, the Accused Products which practice each and every limitation of at least claim 11 of the '032 patent. Dell has infringed literally and/or under the doctrine of equivalents.

46. Upon information and belief, the Accused Products comply with the 802.11n, 802.11ac, and/or 802.11ax standards and the 12 LDPC error correction codes defined in those standards. In addition, upon information and belief, the Accused Products are implemented in a manner that not only complies with the 802.11n, 802.11ac, and/or 802.11ax standards, but also infringes the '032 Patent. This is because implementations of the 802.11n, 802.11ac, and/or 802.11ax standards that infringe the '032 patent perform substantially fewer computations, have substantially more efficient circuitry, use less memory, consume less semiconductor die area, consume less power, and are otherwise more efficient and cost effective than implementations that do not infringe the '032 patent.

47. The 12 LDPC codes were originally defined in the 802.11n version of the standard and include three 1/2 rate, three 2/3 rate, three 3/4 rate, and three 5/6 rate LDPC codes as shown in Table 20-14 of the standard below.⁴

⁴ See IEEE 802.11n-2009 at § 20.3.11.6.2 (emphasis added); see also 802.11-2012 at § 20.3.11.7.2.

Table 20-14—LDPC parameters

Coding rate (R)	LDPC information block length (bits)	LDPC codeword block length (bits)
1/2	972	1944
1/2	648	1296
1/2	324	648
2/3	1296	1944
2/3	864	1296
2/3	432	648
3/4	1458	1944
3/4	972	1296
3/4	486	648
5/6	1620	1944
5/6	1080	1296
5/6	540	648

48. On information and belief, the Accused Products encode information or message bits using an LDPC encoder that supports the 12 LDPC codes defined in the standards. The LDPC encoder encodes the information or message bits to generate a codeword as described in Section 20.3.11.6.3 of the 802.11n standard shown below:⁵

20.3.11.6.3 LDPC encoder

For each of the three available codeword block lengths, the LDPC encoder supports rate 1/2, rate 2/3, rate 3/4, and rate 5/6 encoding. The LDPC encoder is systematic, i.e., it encodes an information block, $\mathbf{c}=(i_0, i_1, \dots, i_{(k-1)})$, of size k , into a codeword, \mathbf{c} , of size n , $\mathbf{c}=(i_0, i_1, \dots, i_{(k-1)}, p_0, p_1, \dots, p_{(n-k-1)})$, by adding $n-k$ parity bits obtained so that $\mathbf{H} \times \mathbf{c}^T = \mathbf{0}$, where \mathbf{H} is an $(n-k) \times n$ parity-check matrix. The selection of the codeword block length (n) is achieved via the LDPC PPDU encoding process described in 20.3.11.6.5.

⁵ See IEEE 802.11n-2009 at § 20.3.11.6.3(emphasis added); see also IEEE 802.11-2012 at § 20.3.11.7.3.

49. On information and belief, the LDPC encoders in the Accused Products encode information or message bits in accordance with the 12 parity-check matrices defined in the 802.11n standard. A parity-check matrix H for each of the 12 block sizes and code rates is defined in Tables R.1 to R.3 of the 802.11n. The parity-check matrix for one of the 12 LDPC codes is shown below.⁶

Table R.1 defines the matrix prototypes of the parity-check matrices for a codeword block length $n=648$ bits, with a subblock size $Z=27$ bits.

Table R.1—Matrix prototypes for codeword block length $n=648$ bits, subblock size is $Z = 27$ bits

* * *

(c) Coding rate $R = 3/4$.																							
16	17	22	24	9	3	14	-	4	2	7	-	26	-	2	-	21	-	1	0	-	-	-	-
25	12	12	3	3	26	6	21	-	15	22	-	15	-	4	-	-	16	-	0	0	-	-	-
25	18	26	16	22	23	9	-	0	-	4	-	4	-	8	23	11	-	-	-	0	0	-	-
9	7	0	1	17	-	-	7	3	-	3	23	-	16	-	-	21	-	0	-	-	0	0	-
24	5	26	7	1	-	-	15	24	15	-	8	-	13	-	13	-	11	-	-	-	-	0	0
2	2	19	14	24	1	15	19	-	21	-	2	-	24	-	3	-	2	1	-	-	-	-	0

50. Each parity-check matrix includes a left-hand side that corresponds to information or message bits, and a right-hand side that corresponds to parity bits. In the parity-check matrix shown above, the left-hand side that corresponds to information or message bits includes columns 1-18, and the right-hand side that corresponds to the parity bits includes columns 19-24. The left-hand side is structured in a way that corresponds to the use of irregular repetition, scrambling and summing in the encoding process, while the right-hand side is structured in a way that corresponds to using accumulation in the encoding process. Further, the left-hand side is structured in a way that corresponds to the use of a low-density generator matrix for performing operations of irregular repetition, scrambling, and summing.

51. A Tanner graph can be constructed from any parity-check matrix. A unique and valuable characteristic of IRA codes is apparent in the Tanner graphs for IRA codes. For example, when constructing a Tanner graph from the 12 LDPC parity-check matrices in the 802.11 standard, message bits are repeated, different subsets of the information bits are repeated different numbers

⁶ See IEEE 802.11n-2009 at Annex R, Table R.1; see also IEEE 802.11-2012 at Annex F, Table F-1.

of times, check nodes are connected to information bits in a random but known pattern, and parity bits are connected to check nodes which enforce a constraint that facilitates the determination of parity bits. While this is not true for a generic LDPC code, it is true for the 12 LDPC codes in the 802.11 standard.

52. On information and belief, the LDPC encoders in the Accused Products are implemented in a manner that meets each and every limitation of claim 11 of the '032 patent. This is because implementations of the 802.11n, 802.11ac, and/or 802.11ax standards that infringe claim 11 of the '032 patent perform substantially fewer computations, have substantially more efficient circuitry, use less memory, consume less semiconductor die area, consume less power, and are otherwise more efficient and cost effective than implementations that do not infringe this claim. The Accused Products are devices that include LDPC encoders. The LDPC encoders receive a collection of message bits and encode the message bits to generate a collection of parity bits. The LDPC encoders in the Accused Products encode the collection of message bits in accordance with the Tanner graph depicted in claim 11. The Tanner graph depicted in claim 11 is a graph representing an IRA code as a set of parity-checks where every message bit is repeated, at least two different subsets of message bits are repeated a different number of times, and check nodes, randomly connected to the repeated message bits, enforce constraints that determine the parity bits.

53. Dell is not licensed or otherwise authorized to practice the claims of the '032 patent.

54. By reason of Dell's infringement, Caltech has suffered substantial damages.

55. Caltech is entitled to recover the damages sustained as a result of Dell's wrongful acts in an amount subject to proof at trial.

56. Caltech has complied with the requirements of 35 U.S.C. § 287(a) at least because neither Caltech nor any party that has held a license to the '032 patent have made, offered for sale, or sold any products in the United States subject to the marking requirements of 35 U.S.C. § 287(a).

57. Dell's infringement of the '032 patent is exceptional and entitles Caltech to attorneys' fees and costs incurred in prosecuting this action under 35 U.S.C. § 285.

COUNT III

Infringement of the '781 Patent

58. Caltech re-alleges and incorporates by reference the allegations of the preceding paragraphs of this Complaint as if fully set forth herein.

59. In violation of 35 U.S.C. § 271(a), Dell has infringed the '781 patent through its use and testing of the Dell Accused Products. Through its use and testing of the Dell Accused Products, Dell performs each and every limitation of at least claim 13 of the '781 patent. Dell has infringed literally and/or under the doctrine of equivalents.

60. Upon information and belief, the Accused Products comply with the 802.11n, 802.11ac, and/or 802.11ax standards and the 12 LDPC error correction codes defined in those standards. In addition, upon information and belief, the Accused Products are implemented in a manner that not only complies with the 802.11n, 802.11ac, and/or 802.11ax standards, but also infringes the '781 Patent. This is because implementations of the 802.11n, 802.11ac, and/or 802.11ax standards that infringe the '781 patent perform substantially fewer computations, have substantially more efficient circuitry, use less memory, consume less semiconductor die area, consume less power, and are otherwise more efficient and cost effective than implementations that do not infringe the '781 patent.

61. The 12 LDPC codes were originally defined in the 802.11n version of the standard and include three 1/2 rate, three 2/3 rate, three 3/4 rate, and three 5/6 rate LDPC codes as shown in Table 20-14 of the standard below.⁷

⁷ See IEEE 802.11n-2009 at § 20.3.11.6.2 (emphasis added); see also 802.11-2012 at § 20.3.11.7.2.

Table 20-14—LDPC parameters

Coding rate (R)	LDPC information block length (bits)	LDPC codeword block length (bits)
1/2	972	1944
1/2	648	1296
1/2	324	648
2/3	1296	1944
2/3	864	1296
2/3	432	648
3/4	1458	1944
3/4	972	1296
3/4	486	648
5/6	1620	1944
5/6	1080	1296
5/6	540	648

62. On information and belief, the Accused Products encode information or message bits using an LDPC encoder that supports the 12 LDPC codes defined in the standards. The LDPC encoder encodes the information or message bits to generate a codeword as described in Section 20.3.11.6.3 of the 802.11n standard shown below:⁸

20.3.11.6.3 LDPC encoder

For each of the three available codeword block lengths, the LDPC encoder supports rate 1/2, rate 2/3, rate 3/4, and rate 5/6 encoding. The LDPC encoder is systematic, i.e., it encodes an information block, $\mathbf{c}=(i_0, i_1, \dots, i_{(k-1)})$, of size k , into a codeword, \mathbf{c} , of size n , $\mathbf{c}=(i_0, i_1, \dots, i_{(k-1)}, p_0, p_1, \dots, p_{(n-k-1)})$, by adding $n-k$ parity bits obtained so that $\mathbf{H} \times \mathbf{c}^T = \mathbf{0}$, where \mathbf{H} is an $(n-k) \times n$ parity-check matrix. The selection of the codeword block length (n) is achieved via the LDPC PPDU encoding process described in 20.3.11.6.5.

63. On information and belief, the LDPC encoders in the Accused Products encode information or message bits in accordance with the 12 parity-check matrices defined in the 802.11n

⁸ See IEEE 802.11n-2009 at § 20.3.11.6.3(emphasis added); see also IEEE 802.11-2012 at § 20.3.11.7.3.

standard. A parity-check matrix H for each of the 12 block sizes and code rates is defined in Tables R.1 to R.3 of the 802.11n. The parity-check matrix for one of the 12 LDPC codes is shown below.⁹

Table R.1 defines the matrix prototypes of the parity-check matrices for a codeword block length $n=648$ bits, with a subblock size $Z=27$ bits.

Table R.1—Matrix prototypes for codeword block length $n=648$ bits, subblock size is $Z = 27$ bits

* * *

(c) Coding rate $R = 3/4$.																							
16	17	22	24	9	3	14	-	4	2	7	-	26	-	2	-	21	-	1	0	-	-	-	-
25	12	12	3	3	26	6	21	-	15	22	-	15	-	4	-	-	16	-	0	0	-	-	-
25	18	26	16	22	23	9	-	0	-	4	-	4	-	8	23	11	-	-	-	0	0	-	-
9	7	0	1	17	-	-	7	3	-	3	23	-	16	-	-	21	-	0	-	-	0	0	-
24	5	26	7	1	-	-	15	24	15	-	8	-	13	-	13	-	11	-	-	-	-	0	0
2	2	19	14	24	1	15	19	-	21	-	2	-	24	-	3	-	2	1	-	-	-	-	0

64. Each parity-check matrix includes a left-hand side that corresponds to information or message bits, and a right-hand side that corresponds to parity bits. In the parity-check matrix shown above, the left-hand side that corresponds to information or message bits includes columns 1-18, and the right-hand side that corresponds to the parity bits includes columns 19-24. The left-hand side is structured in a way that corresponds to the use of irregular repetition, scrambling and summing in the encoding process, while the right-hand side is structured in a way that corresponds to using accumulation in the encoding process. Further, the left-hand side is structured in a way that corresponds to the use of a low-density generator matrix for performing operations of irregular repetition, scrambling and summing.

65. On information and belief, the LDPC encoders in the Accused Products are implemented in a manner that meets each and every limitation of claim 13 of the '781 patent. This is because implementations of the 802.11n, 802.11ac, and/or 802.11ax standards that infringe claim 13 of the '781 patent perform substantially fewer computations, have substantially more efficient circuitry, use less memory, consume less semiconductor die area, consume less power, and are otherwise more efficient and cost effective than implementations that do not infringe this claim.

⁹ See IEEE 802.11n-2009 at Annex R, Table R.1; see also IEEE 802.11-2012 at Annex F, Table F-1.

The LDPC encoders perform a method of encoding a signal. The LDPC encoders receive a block of data in the signal to be encoded. The block of data includes information bits. The LDPC encoders perform an encoding operation using the information bits as an input. The encoding operation includes an accumulation of mod-2 or exclusive-OR sums of bits in subsets of the information bits. The non-null values in each row in the left-hand side of the parity-check matrices correspond to the subsets of information bits that are summed.¹⁰ The accumulation of the sums of bits in subsets of the information bits corresponds to the accumulation operations depicted in the left-hand side of the parity-check matrices.

66. Dell is not licensed or otherwise authorized to practice the claims of the '781 patent.

67. By reason of Dell's infringement, Caltech has suffered substantial damages.

68. Caltech is entitled to recover the damages sustained as a result of Dell's wrongful acts in an amount subject to proof at trial.

69. Caltech has complied with the requirements of 35 U.S.C. § 287(a) at least because neither Caltech nor any party that has held a license to the '781 patent have made, offered for sale, or sold any products in the United States subject to the marking requirements of 35 U.S.C. § 287(a).

70. Dell's infringement of the '781 patent is exceptional and entitles Caltech to attorneys' fees and costs incurred in prosecuting this action under 35 U.S.C. § 285.

COUNT IV

Infringement of the '833 Patent

71. Caltech re-alleges and incorporates by reference the allegations of the preceding paragraphs of this Complaint as if fully set forth herein.

72. In violation of 35 U.S.C. § 271(a), Dell has infringed the '833 patent by making, using, selling, offering for sale, and/or importing into the United States, without authority, the Accused Products which practice each and every limitation of at least claim 1 of the '833 patent. Dell has infringed literally and/or under the doctrine of equivalents.

¹⁰ The null values are represented by “-” in the parity-check matrices. The non-null values are represented by numbers.

73. Upon information and belief, the Accused Products comply with the 802.11n, 802.11ac, and/or 802.11ax standards and the 12 LDPC error correction codes defined in those standards. In addition, upon information and belief, the Accused Products are implemented in a manner that not only complies with the 802.11n, 802.11ac, and/or 802.11ax standards, but also infringes the '833 patent. This is because implementations of the 802.11n, 802.11ac, and/or 802.11ax standards that infringe the '833 patent perform substantially fewer computations, have substantially more efficient circuitry, use less memory, consume less semiconductor die area, consume less power, and are otherwise more efficient and cost effective than implementations that do not infringe the '833 patent.

74. The 12 LDPC codes were originally defined in the 802.11n version of the standard and include three 1/2 rate, three 2/3 rate, three 3/4 rate, and three 5/6 rate LDPC codes as shown in Table 20-14 of the standard below.¹¹

Table 20-14—LDPC parameters

Coding rate (R)	LDPC information block length (bits)	LDPC codeword block length (bits)
1/2	972	1944
1/2	648	1296
1/2	324	648
2/3	1296	1944
2/3	864	1296
2/3	432	648
3/4	1458	1944
3/4	972	1296
3/4	486	648
5/6	1620	1944
5/6	1080	1296
5/6	540	648

¹¹ See IEEE 802.11n-2009 at § 20.3.11.6.2 (emphasis added); see also 802.11-2012 at § 20.3.11.7.2.

75. On information and belief, the Accused Products encode information or message bits using an LDPC encoder that supports the 12 LDPC codes defined in the standards. The LDPC encoder encodes the information or message bits to generate a codeword as described in Section 20.3.11.6.3 of the 802.11n standard shown below:¹²

20.3.11.6.3 LDPC encoder

For each of the three available codeword block lengths, the LDPC encoder supports rate 1/2, rate 2/3, rate 3/4, and rate 5/6 encoding. The LDPC encoder is systematic, i.e., it encodes an information block, $\mathbf{c}=(i_0, i_1, \dots, i_{(k-1)})$, of size k , into a codeword, \mathbf{c} , of size n , $\mathbf{c}=(i_0, i_1, \dots, i_{(k-1)}, p_0, p_1, \dots, p_{(n-k-1)})$, by adding $n-k$ parity bits obtained so that $\mathbf{H} \times \mathbf{c}^T = \mathbf{0}$, where \mathbf{H} is an $(n-k) \times n$ parity-check matrix. The selection of the codeword block length (n) is achieved via the LDPC PPDU encoding process described in 20.3.11.6.5.

76. On information and belief, the LDPC encoders in the Accused Products encode information or message bits in accordance with the 12 parity-check matrices defined in the 802.11n standard. A parity-check matrix \mathbf{H} for each of the 12 block sizes and code rates is defined in Tables R.1 to R.3 of the 802.11n. The parity-check matrix for one of the 12 LDPC codes is shown below.¹³

Table R.1 defines the matrix prototypes of the parity-check matrices for a codeword block length $n=648$ bits, with a subblock size $Z=27$ bits.

Table R.1—Matrix prototypes for codeword block length $n=648$ bits, subblock size is $Z = 27$ bits

* * *

(c) Coding rate $R = 3/4$.																							
16	17	22	24	9	3	14	-	4	2	7	-	26	-	2	-	21	-	1	0	-	-	-	-
25	12	12	3	3	26	6	21	-	15	22	-	15	-	4	-	-	16	-	0	0	-	-	-
25	18	26	16	22	23	9	-	0	-	4	-	4	-	8	23	11	-	-	-	0	0	-	-
9	7	0	1	17	-	-	7	3	-	3	23	-	16	-	-	21	-	0	-	-	0	0	-
24	5	26	7	1	-	-	15	24	15	-	8	-	13	-	13	-	11	-	-	-	-	0	0
2	2	19	14	24	1	15	19	-	21	-	2	-	24	-	3	-	2	1	-	-	-	-	0

77. Each parity-check matrix includes a left-hand side that corresponds to information or message bits, and a right-hand side that corresponds to parity bits. In the parity-check matrix shown above, the left-hand side that corresponds to information or message bits includes columns

¹² See IEEE 802.11n-2009 at § 20.3.11.6.3(emphasis added); see also IEEE 802.11-2012 at § 20.3.11.7.3.

¹³ See IEEE 802.11n-2009 at Annex R, Table R.1; see also IEEE 802.11-2012 at Annex F, Table F-1.

1-18, and the right-hand side that corresponds to the parity bits includes columns 19-24. The left-hand side is structured in a way that corresponds to the use of irregular repetition, scrambling and summing in the encoding process, while the right-hand side is structured in a way that corresponds to using accumulation in the encoding process. Further, the left-hand side is structured in a way that corresponds to the use of a low-density generator matrix for performing operations of irregular repetition, scrambling and summing.

78. On information and belief, the LDPC encoders in the Accused Products are implemented in a manner that meets each and every limitation of claim 1 of the '833 patent. This is because implementations of the 802.11n, 802.11ac, and/or 802.11ax standards that infringe claim 1 of the '833 patent perform substantially fewer computations, have substantially more efficient circuitry, use less memory, consume less semiconductor die area, consume less power, and are otherwise more efficient and cost effective than implementations that do not infringe this claim. The LDPC encoders in the Accused Products are an apparatus for performing encoding operations. The LDPC encoders in the Accused Products include a first a first set of memory locations to store information bits where two or more memory locations of the first set of memory locations are read by the permutation module different times from one another. The LDPC encoders in the Accused Products also include a second set of memory locations to store parity bits. The LDPC encoders in the Accused Products further include a permutation module to read a bit from the first set of memory locations and combine the read bit to a bit in the second set of memory locations based on a corresponding index of the first set of memory locations and a corresponding index of the second set of memory locations. The LDPC encoders in the Accused Products include an accumulator to perform accumulation operations on the bits stored in the second set of memory locations.

79. Dell is not licensed or otherwise authorized to practice the claims of the '833 patent.

80. By reason of Dell's infringement, Caltech has suffered substantial damages.

81. Caltech is entitled to recover the damages sustained as a result of Dell's wrongful acts in an amount subject to proof at trial.

82. Caltech has complied with the requirements of 35 U.S.C. § 287(a) at least because neither Caltech nor any party that has held a license to the '833 patent have made, offered for sale, or sold any products in the United States subject to the marking requirements of 35 U.S.C. § 287(a).

83. Dell's infringement of the '833 patent is exceptional and entitles Caltech to attorneys' fees and costs incurred in prosecuting this action under 35 U.S.C. § 285.

DEMAND FOR JURY TRIAL

Pursuant to Rule 38 of the Federal Rules of Civil Procedure, Plaintiff hereby demands a trial by jury as to all issues so triable.

PRAYER FOR RELIEF

WHEREFORE, Plaintiff respectfully prays for the following relief:

- (a) A judgment that Defendants have infringed each and every one of the Asserted Patents;
- (b) Damages adequate to compensate Caltech for Defendants' infringement of the Asserted Patents pursuant to 35 U.S.C. § 284;
- (c) Pre-judgment interest;
- (d) Post-judgment interest;
- (e) A declaration that this action is exceptional pursuant to 35 U.S.C. § 285, and an award to Caltech of its attorneys' fees, costs, and expenses incurred in connection with this action; and
- (f) Such other relief as the Court deems just and equitable.

DATED: January 19, 2024

Respectfully submitted,

By /s/ J. Mark Mann

J. Mark Mann
MANN TINDEL THOMPSON
300 West Main Street
Henderson, Texas 75652
Telephone: (903) 657-8540
Facsimile: (903) 657-6003
Mark@themannfirm.com

James R. Asperger
QUINN EMANUEL URQUHART &
SULLIVAN, LLP
865 S. Figueroa Street, 10th Floor
Los Angeles, CA 90017-2543
Telephone: (213) 443-3000
Facsimile: (213) 443 3100
jimasperger@quinnemanuel.com

Kevin Johnson
Todd Briggs
QUINN EMANUEL URQUHART &
SULLIVAN, LLP
555 Twin Dolphin Drive, 5th Floor
Redwood Shores, California 94065-2139
Telephone: (650) 801 5000
Facsimile: (650) 801 5100
kevinjohnson@quinnemanuel.com
toddbriggs@quinnemanuel.com

Brian Biddinger
QUINN EMANUEL URQUHART &
SULLIVAN, LLP
51 Madison Avenue, 22nd Floor
New York, New York 10010-1601
Telephone: (212) 849 7000
Facsimile: (212) 849 7100
brianbiddinger@quinnemanuel.com

*Attorneys for Plaintiff California Institute of
Technology*

EXHIBIT A



US007116710B1

(12) **United States Patent**
Jin et al.

(10) **Patent No.:** **US 7,116,710 B1**
 (45) **Date of Patent:** **Oct. 3, 2006**

(54) **SERIAL CONCATENATION OF INTERLEAVED CONVOLUTIONAL CODES FORMING TURBO-LIKE CODES**

(75) Inventors: **Hui Jin**, Glen Gardner, NJ (US);
Aamod Khandekar, Pasadena, CA (US); **Robert J. McEliece**, Pasadena, CA (US)

(73) Assignee: **California Institute of Technology**, Pasadena, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 735 days.

(21) Appl. No.: **09/861,102**

(22) Filed: **May 18, 2001**

Related U.S. Application Data

(60) Provisional application No. 60/205,095, filed on May 18, 2000.

(51) **Int. Cl.**
H04B 1/66 (2006.01)

(52) **U.S. Cl.** **375/240; 375/262; 375/265; 375/341; 341/51; 341/102; 714/752**

(58) **Field of Classification Search** **375/259, 375/262, 265, 285, 296, 341, 346, 348; 714/746, 714/752, 755, 756, 786, 792, 794, 795, 796; 341/51, 52, 56, 102, 103**

See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,392,299 A 2/1995 Rhines et al.
 5,751,739 A * 5/1998 Seshadri et al. 714/746

5,881,093 A 3/1999 Wang et al.
 6,014,411 A * 1/2000 Wang 375/259
 6,023,783 A 2/2000 Divsalar et al.
 6,031,874 A 2/2000 Chennakeshu et al.
 6,032,284 A 2/2000 Bliss
 6,044,116 A 3/2000 Wang
 6,396,423 B1 * 5/2002 Laumen et al. 341/95
 6,437,714 B1 * 8/2002 Kim et al. 341/81
 2001/0025358 A1 9/2001 Eidson et al.

OTHER PUBLICATIONS

Wiberg et al., "Codes and Iterative Decoding on General Graphs", 1995 Intl. Symposium on Information Theory, Sep. 1995, p. 506.*
 Appendix A.1 "Structure of Parity Check Matrices of Standardized LDPC Codes," Digital Video Broadcasting (DVB) User guidelines for the second generation system for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications (DVB-S2) ETSI TR 102 376 V1.1.1. (2005-02) Technical Report. pp. 64.
 Benedetto et al., "Bandwidth efficient parallel concatenated coding schemes," Electronics Letters 31(24):2067-2069 (Nov. 23, 1995).
 Benedetto et al., "Soft-output decoding algorithms in iterative decoding of turbo codes," The Telecommunications and Data Acquisition (TDA) Progress Report 42-124 for NASA and California Institute of Technology Jet Propulsion Laboratory, Joseph H. Yuen, Ed., pp. 63-87 (Feb. 15, 1996).

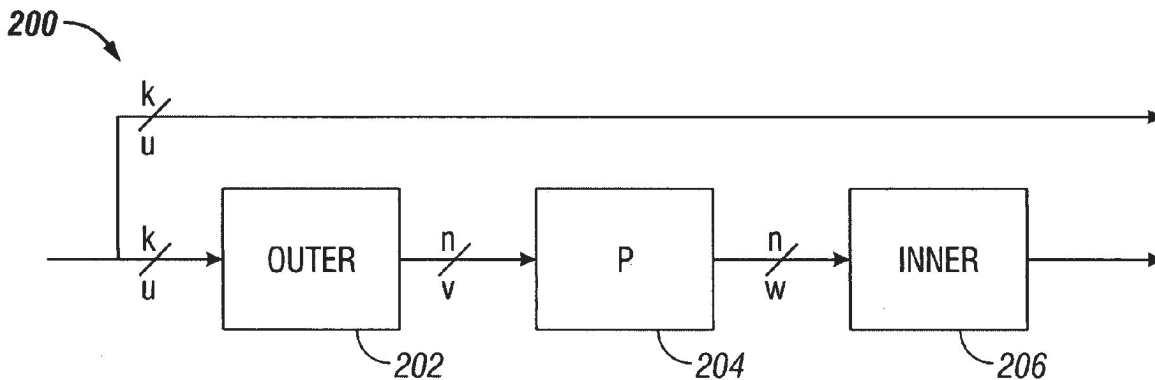
(Continued)

Primary Examiner—Dac V. Ha
 (74) *Attorney, Agent, or Firm*—Fish & Richardson P.C.

(57) **ABSTRACT**

A serial concatenated coder includes an outer coder and an inner coder. The outer coder irregularly repeats bits in a data block according to a degree profile and scrambles the repeated bits. The scrambled and repeated bits are input to an inner coder, which has a rate substantially close to one.

33 Claims, 5 Drawing Sheets



US 7,116,710 B1

Page 2

OTHER PUBLICATIONS

- Benedetto et al., "Serial Concatenation of Interleaved Codes: Performance Analysis, Design, and Iterative Decoding," The Telecommunications and Data Acquisition (TDA) Progress Report 42-126 for NASA and California Institute of Technology Jet Propulsion Laboratory, Joseph H. Yuen, Ed., pp. 1-26 (Aug. 15, 1996).
- Benedetto et al., "A Soft-Input Soft-Output Maximum A Posteriori (MAP) Module to Decode Parallel and Serial Concatenated Codes," The Telecommunications and Data Acquisition (TDA) Progress Report 42-127 for NASA and California Institute of Technology Jet Propulsion Laboratory, Joseph H. Yuen, Ed., pp. 1-20 (Nov. 15, 1996).
- Benedetto et al., "Parallel Concatenated Trellis Coded Modulation," ICC '96, IEEE, pp. 974-978, (Jun. 1996).
- Benedetto, S. et al., "A Soft-Input Soft-Output APP Module for Iterative Decoding of Concatenated Codes," IEEE Communications Letters 1(1):22-24 (Jan. 1997).
- Benedetto et al., "Serial Concatenation of interleaved codes: performance analysis, design, and iterative decoding," Proceedings from the IEEE 1997 International Symposium on Information Theory (ISIT), Ulm, Germany, p. 106, Jun. 29-Jul. 4, 1997.
- Benedetto et al., "Serial Concatenated Trellis Coded Modulation with Iterative Decoding," Proceedings from IEEE 1997 International Symposium on Information Theory (ISIT), Ulm, Germany, p. 8, Jun. 29-Jul. 4, 1997.
- Benedetto et al., "Design of Serially Concatenated Interleaved Codes," ICC 97, Montreal, Canada, pp. 710-714, (Jun. 1997).
- Berrou et al., "Near Shannon Limit Error-Correcting Coding and Decoding: Turbo Codes," ICC pp. 1064-1070 (1993).
- Digital Video Broadcasting (DVB) User guidelines for the second generation system for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications (DVB-S2) ETSI TR 102 376 V1.1.1. (Feb. 2005) Technical Report, pp. 1-104 (Feb. 15, 2005).
- Divsalar et al., "Coding Theorems for 'Turbo-Like' Codes," Proceedings of the 36th Annual Allerton Conference on Communication, Control, and Computing, Sep. 23-25 1998, Allerton House, Monticello, Illinois, pp. 201-210 (1998).
- Divsalar, D. et al., "Multiple Turbo Codes for Deep-Space Communications," The Telecommunications and Data Acquisition (TDA) Progress Report 42-121 for NASA and California Institute of Technology Jet Propulsion Laboratory, Joseph H. Yuen, Ed., pp. 60-77 (May 15, 1995).
- Divsalar, D. et al., "On the Design of Turbo Codes," The Telecommunications and Data Acquisition (TDA) Progress Report 42-123 for NASA and California Institute of Technology Jet Propulsion Laboratory, Joseph H. Yuen, Ed., pp. 99-131 (Nov. 15, 1995).
- Divsalar, D. et al., "Low-rate turbo codes for Deep Space Communications," Proceedings from the 1995 IEEE International Symposium on Information Theory, Sep. 17-22, 1995, Whistler, British Columbia, Canada, p. 35.
- Divsalar, D. et al., "Turbo Codes for PCS Applications," ICC 95, IEEE, Seattle, WA, pp. 54-59 (Jun. 1995).
- Divsalar, D. et al., "Multiple Turbo Codes," MILCOM 95, San Diego, CA pp. 279-285 (Nov. 5-6, 1995).
- Divsalar et al., "Effective free distance of turbo codes," Electronics Letters 32(5): 445-446 (Feb. 29, 1996).
- Divsalar, D. et al., "Hybrid concatenated codes and Iterative Decoding," Proceedings from the IEEE 1997 International Symposium on Information Theory (ISIT), Ulm, Germany, p. 10 (Jun. 29-Jul. 4, 1997).
- Divsalar, D. et al., "Serial Turbo Trellis Coded Modulation with Rate-1 Inner Code," Proceedings from the IEEE 2000 International Symposium on Information Theory (ISIT), Italy, pp. 1-14 (Jun. 2000).
- Jin et al., "Irregular Repeat - Accumulate Codes," 2nd International Symposium on Turbo Codes & Related Topics, Sep. 4-7, 2000, Brest, France, 25 slides, (presented on Sep. 4, 2000).
- Jin et al., "Irregular Repeat-Accumulate Codes," 2nd International Symposium on Turbo Codes & Related Topics, Sep. 4-7, 2000, Brest, France, pp. 1-8 (2000).
- Richardson, et al., "Design of capacity approaching irregular low density parity check codes," IEEE Trans, Inform. Theory 47: 619-637 (Feb. 2001).
- Richardson, T. and R. Urbanke, "Efficient encoding of low-density parity check codes," IEEE Trans. Inform. Theory 47: 638-656 (Feb. 2001).

* cited by examiner

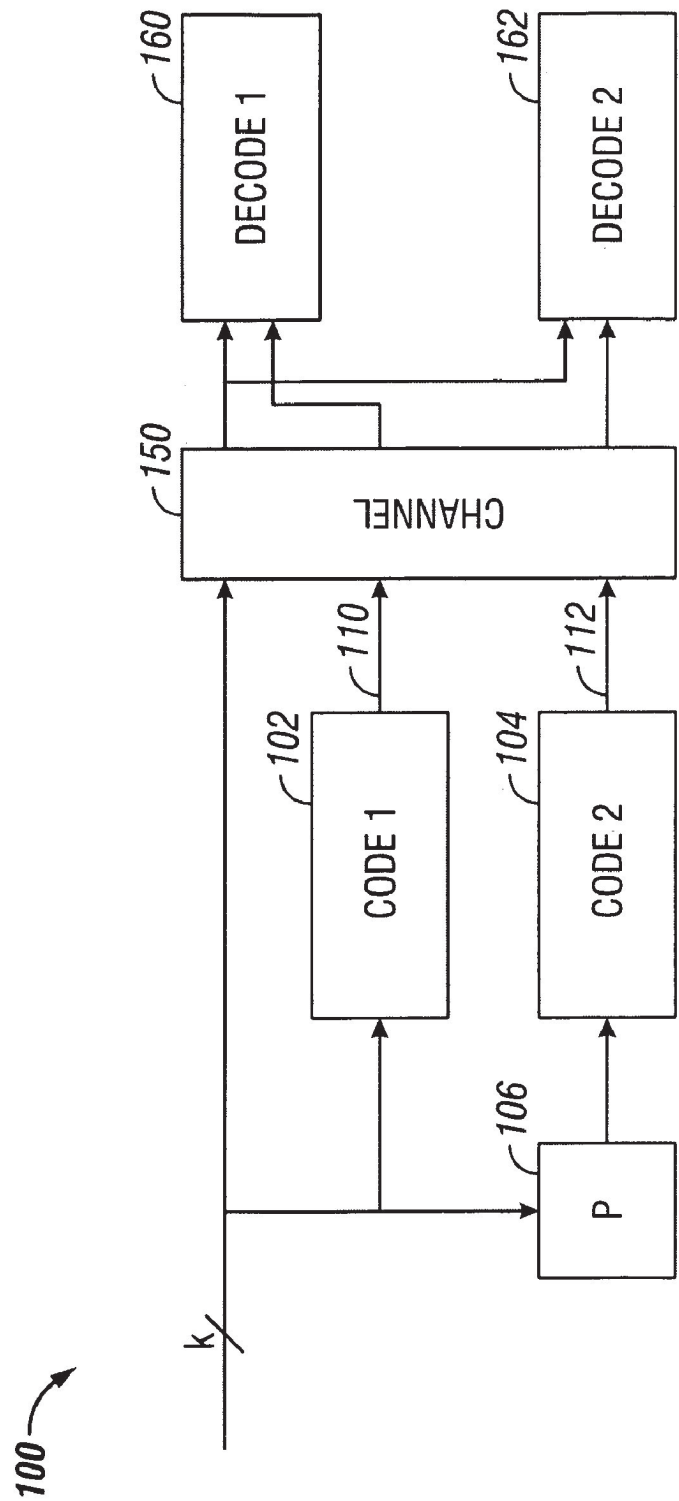


FIG. 1
(Prior Art)

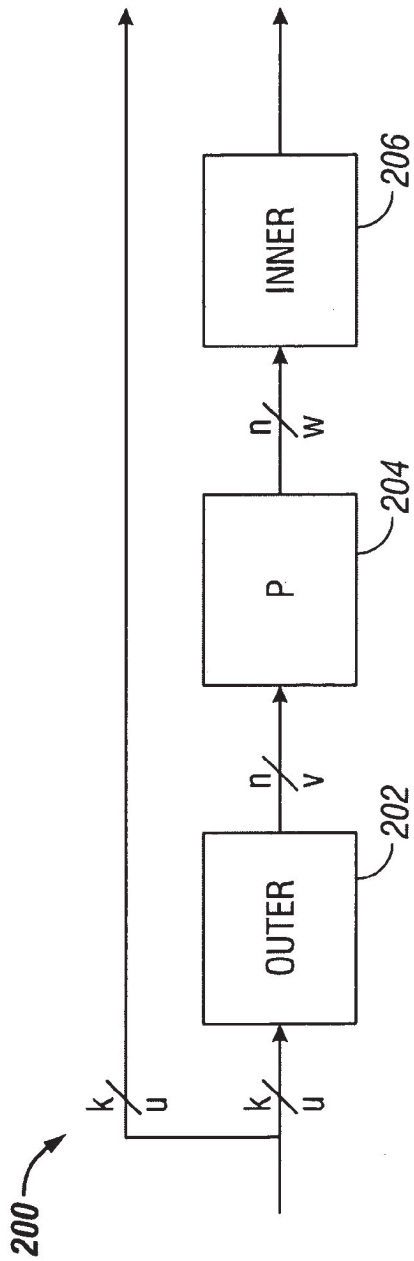


FIG. 2

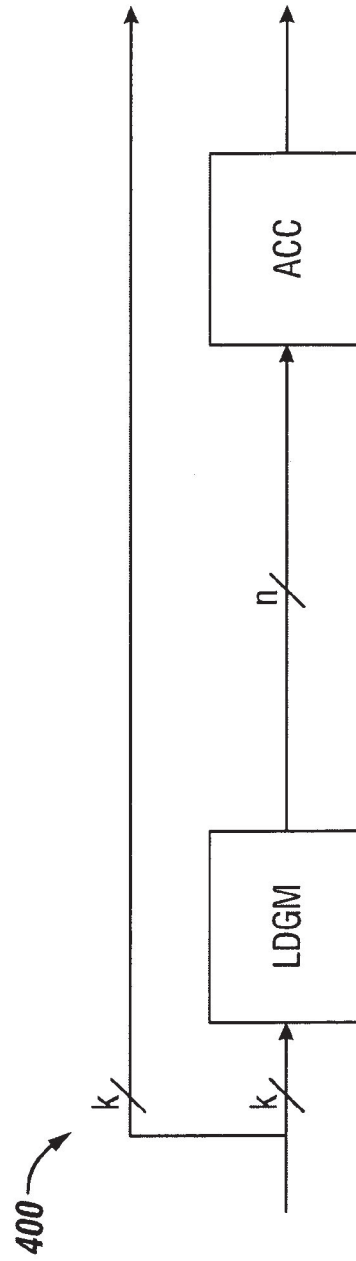


FIG. 4

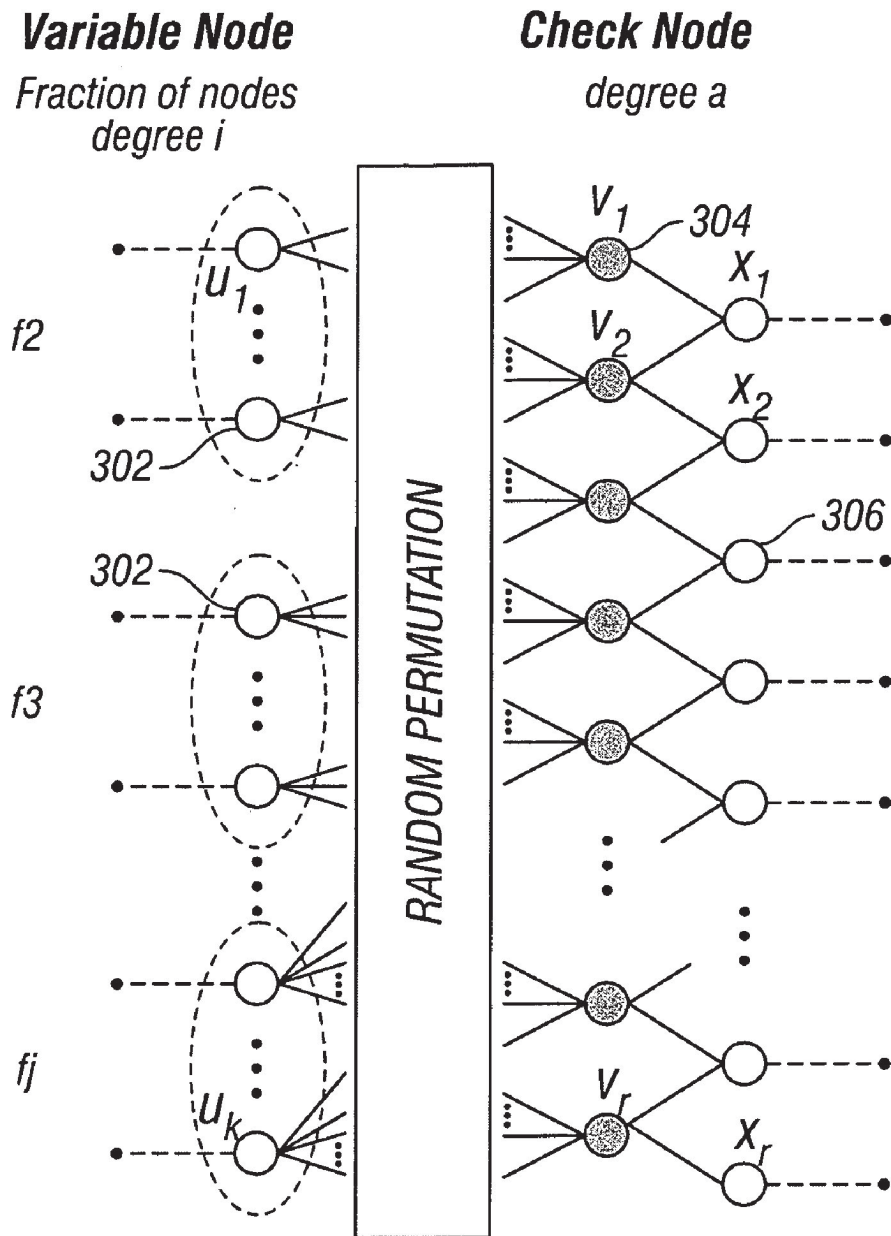


FIG. 3

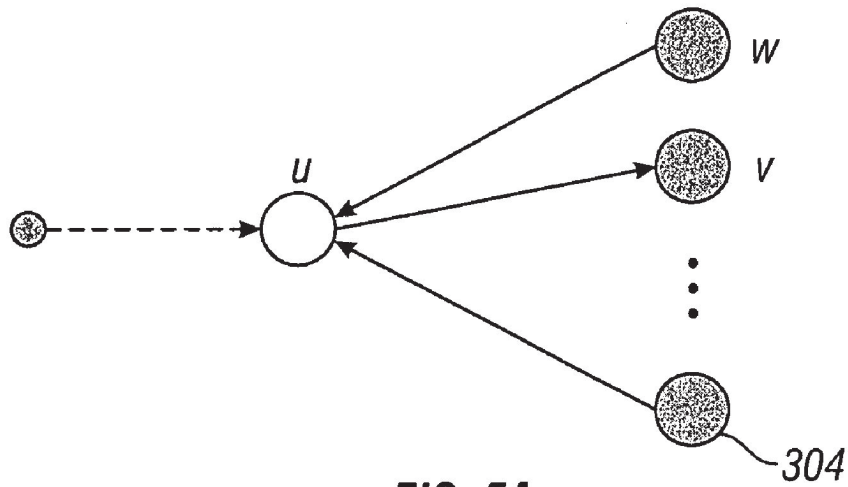


FIG. 5A

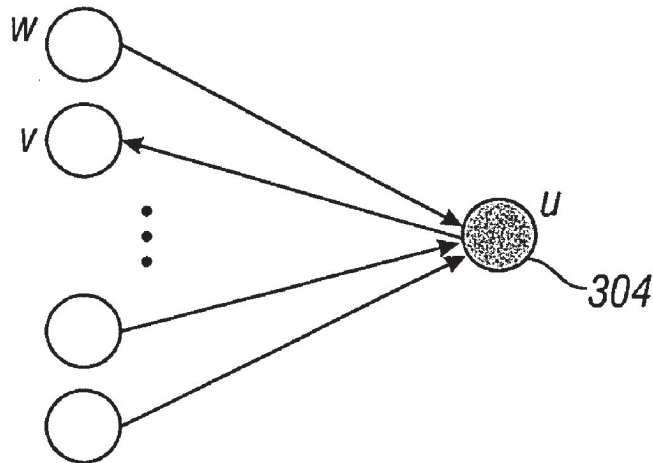


FIG. 5B

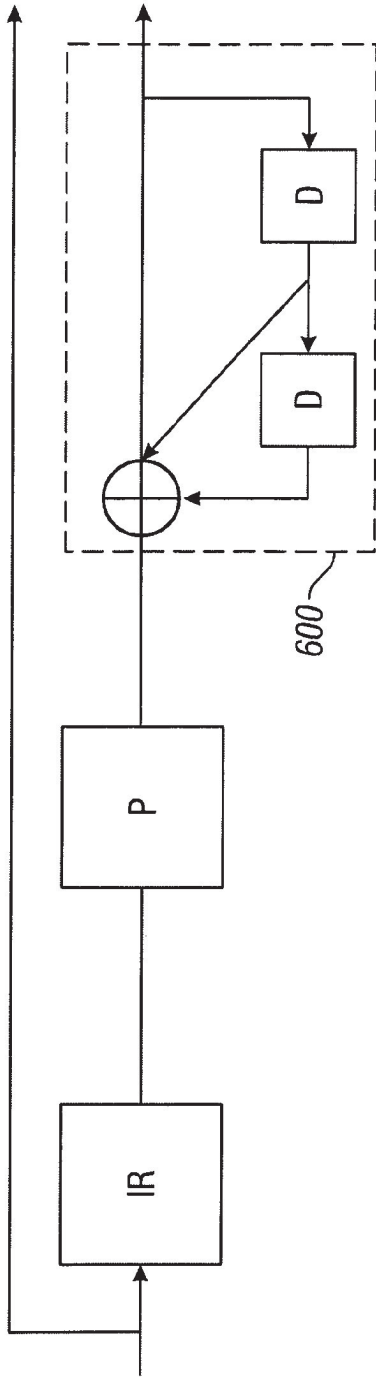


FIG. 6

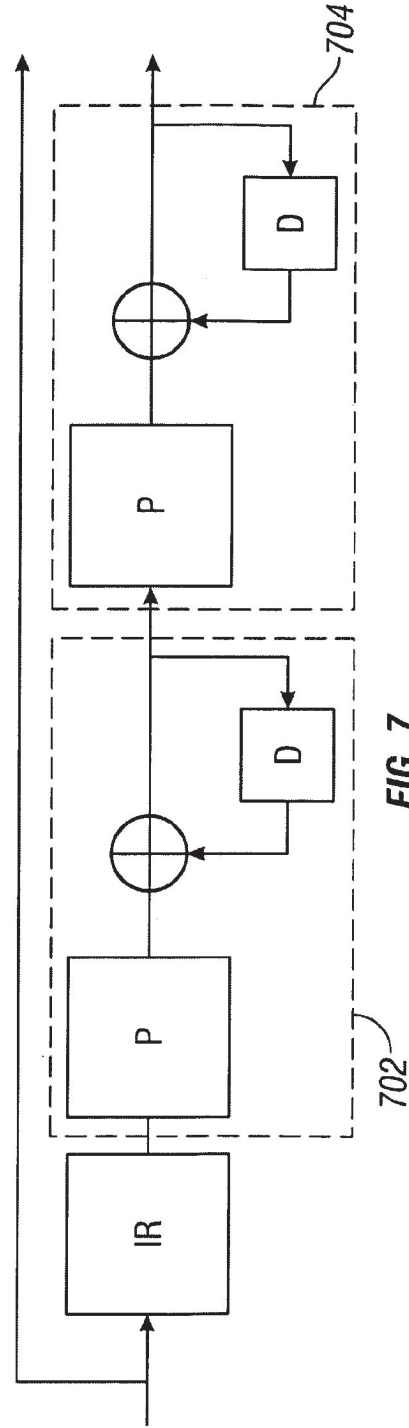


FIG. 7

US 7,116,710 B1

1

SERIAL CONCATENATION OF INTERLEAVED CONVOLUTIONAL CODES FORMING TURBO-LIKE CODES

CROSS-REFERENCE TO RELATED APPLICATIONS

This application claims priority to U.S. Provisional Application Ser. No. 60/205,095, filed on May 18, 2000, and to U.S. application Ser. No. 09/922,852, filed on Aug. 18, 2000 and entitled Interleaved Serial Concatenation Forming Turbo-Like Codes.

GOVERNMENT LICENSE RIGHTS

The U.S. Government has a paid-up license in this invention and the right in limited circumstances to require the patent owner to license others on reasonable terms as provided for by the terms of Grant No. CCR-9804793 awarded by the National Science Foundation.

BACKGROUND

Properties of a channel affect the amount of data that can be handled by the channel. The so-called "Shannon limit" defines the theoretical limit of the amount of data that a channel can carry.

Different techniques have been used to increase the data rate that can be handled by a channel. "Near Shannon Limit Error-Correcting Coding and Decoding: Turbo Codes," by Berrou et al. ICC, pp 1064-1070, (1993), described a new "turbo code" technique that has revolutionized the field of error correcting codes. Turbo codes have sufficient randomness to allow reliable communication over the channel at a high data rate near capacity. However, they still retain sufficient structure to allow practical encoding and decoding algorithms. Still, the technique for encoding and decoding turbo codes can be relatively complex.

A standard turbo coder **100** is shown in FIG. 1. A block of k information bits is input directly to a first coder **102**. A k bit interleaver **106** also receives the k bits and interleaves them prior to applying them to a second coder **104**. The second coder produces an output that has more bits than its input, that is, it is a coder with rate that is less than 1. The coders **102**, **104** are typically recursive convolutional coders.

Three different items are sent over the channel **150**: the original k bits, first encoded bits **110**, and second encoded bits **112**. At the decoding end, two decoders are used: a first constituent decoder **160** and a second constituent decoder **162**. Each receives both the original k bits, and one of the encoded portions **110**, **112**. Each decoder sends likelihood estimates of the decoded bits to the other decoders. The estimates are used to decode the uncoded information bits as corrupted by the noisy channel.

SUMMARY

A coding system according to an embodiment is configured to receive a portion of a signal to be encoded, for example, a data block including a fixed number of bits. The coding system includes an outer coder, which repeats and scrambles bits in the data block. The data block is apportioned into two or more sub-blocks, and bits in different sub-blocks are repeated a different number of times according to a selected degree profile. The outer coder may include a repeater with a variable rate and an interleaver. Alternatively, the outer coder may be a low-density generator matrix (LDGM) coder.

2

The repeated and scrambled bits are input to an inner coder that has a rate substantially close to one. The inner coder may include one or more accumulators that perform recursive modulo two addition operations on the input bit stream.

The encoded data output from the inner coder may be transmitted on a channel and decoded in linear time at a destination using iterative decoding techniques. The decoding techniques may be based on a Tanner graph representation of the code.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a schematic diagram of a prior "turbo code" system.

FIG. 2 is a schematic diagram of a coder according to an embodiment.

FIG. 3 is a Tanner graph for an irregular repeat and accumulate (IRA) coder.

FIG. 4 is a schematic diagram of an IRA coder according to an embodiment.

FIG. 5A illustrates a message from a variable node to a check node on the Tanner graph of FIG. 3.

FIG. 5B illustrates a message from a check node to a variable node on the Tanner graph of FIG. 3.

FIG. 6 is a schematic diagram of a coder according to an alternate embodiment.

FIG. 7 is a schematic diagram of a coder according to another alternate embodiment.

DETAILED DESCRIPTION

FIG. 2 illustrates a coder **200** according to an embodiment. The coder **200** may include an outer coder **202**, an interleaver **204**, and inner coder **206**. The coder may be used to format blocks of data for transmission, introducing redundancy into the stream of data to protect the data from loss due to transmission errors. The encoded data may then be decoded at a destination in linear time at rates that may approach the channel capacity.

The outer coder **202** receives the uncoded data. The data may be partitioned into blocks of fixed size, say k bits. The outer coder may be an (n,k) binary linear block coder, where $n > k$. The coder accepts as input a block u of k data bits and produces an output block v of n data bits. The mathematical relationship between u and v is $v = T_0 u$, where T_0 is an $n \times k$ matrix, and the rate of the coder is k/n .

The rate of the coder may be irregular, that is, the value of T_0 is not constant, and may differ for sub-blocks of bits in the data block. In an embodiment, the outer coder **202** is a repeater that repeats the k bits in a block a number of times q to produce a block with n bits, where $n = qk$. Since the repeater has an irregular output, different bits in the block may be repeated a different number of times. For example, a fraction of the bits in the block may be repeated two times, a fraction of bits may be repeated three times, and the remainder of bits may be repeated four times. These fractions define a degree sequence, or degree profile, of the code.

The inner coder **206** may be a linear rate-1 coder, which means that the n -bit output block x can be written as $x = T_1 w$, where T_1 is a nonsingular $n \times n$ matrix. The inner coder **210** can have a rate that is close to 1, e.g., within 50%, more preferably 10% and perhaps even more preferably within 1% of 1.

In an embodiment, the inner coder **206** is an accumulator, which produces outputs that are the modulo two (mod-2) partial sums of its inputs. The accumulator may be a

US 7,116,710 B1

3

truncated rate-1 recursive convolutional coder with the transfer function $1/(1+D)$. Such an accumulator may be considered a block coder whose input block $[x_1, \dots, x_n]$ and output block $[y_1, \dots, y_n]$ are related by the formula

$$y_1 = x_1$$

$$y_2 = x_1 \oplus x_2$$

$$y_3 = x_1 \oplus x_2 \oplus x_3$$

$$y_n = x_1 \oplus x_2 \oplus x_3 \oplus \dots \oplus x_n$$

where “ \oplus ” denotes mod-2, or exclusive-OR (XOR), addition. An advantage of this system is that only mod-2 addition is necessary for the accumulator. The accumulator may be embodied using only XOR gates, which may simplify the design.

The bits output from the outer coder **202** are scrambled before they are input to the inner coder **206**. This scrambling may be performed by the interleaver **204**, which performs a pseudo-random permutation of an input block v , yielding an output block w having the same length as v .

The serial concatenation of the interleaved irregular repeat code and the accumulate code produces an irregular repeat and accumulate (IRA) code. An IRA code is a linear code, and as such, may be represented as a set of parity checks. The set of parity checks may be represented in a bipartite graph, called the Tanner graph, of the code. FIG. 3 shows a Tanner graph **300** of an IRA code with parameters $(f_1, \dots, f_r; a)$, where $f_i \geq 0$, $\sum_i f_i = 1$ and “ a ” is a positive integer. The Tanner graph includes two kinds of nodes: variable nodes (open circles) and check nodes (filled circles). There are k variable nodes **302** on the left, called information nodes. There are r variable nodes **306** on the right, called parity nodes. There are $r = (k \sum_i f_i) / a$ check nodes **304** connected between the information nodes and the parity nodes. Each information node **302** is connected to a number of check nodes **304**. The fraction of information nodes connected to exactly i check nodes is f_i . For example, in the Tanner graph **300**, each of the f_2 information nodes are connected to two check nodes, corresponding to a repeat of $q=2$, and each of the f_3 information nodes are connected to three check nodes, corresponding to $q=3$.

Each check node **304** is connected to exactly “ a ” information nodes **302**. In FIG. 3, $a=3$. These connections can be made in many ways, as indicated by the arbitrary permutation of the ra edges joining information nodes **302** and check nodes **304** in permutation block **310**. These connections correspond to the scrambling performed by the interleaver **204**.

In an alternate embodiment, the outer coder **202** may be a low-density generator matrix (LDGM) coder that performs an irregular repeat of the k bits in the block, as shown in FIG. 4. As the name implies, an LDGM code has a sparse (low-density) generator matrix. The IRA code produced by the coder **400** is a serial concatenation of the LDGM code and the accumulator code. The interleaver **204** in FIG. 2 may be excluded due to the randomness already present in the structure of the LDGM code.

If the permutation performed in permutation block **310** is fixed, the Tanner graph represents a binary linear block code with k information bits (u_1, \dots, u_k) and r parity bits (x_1, \dots, x_r) , as follows. Each of the information bits is associated with one of the information nodes **302**, and each of the parity bits is associated with one of the parity nodes **306**. The value of a parity bit is determined uniquely by the condition that the mod-2 sum of the values of the variable nodes connected

4

to each of the check nodes **304** is zero. To see this, set $x_0=0$. Then if the values of the bits on the ra edges coming out of the permutation box are (v_1, \dots, v_{ra}) , then we have the recursive formula

$$x_j = x_{j-1} + \sum_{i=1}^{\lambda} v_{(j-1)\lambda+i}$$

for $j=1, 2, \dots, r$. This is in effect the encoding algorithm.

Two types of IRA codes are represented in FIG. 3, a nonsystematic version and a systematic version. The nonsystematic version is an (r,k) code, in which the codeword corresponding to the information bits (u_1, \dots, u_k) is (x_1, \dots, x_r) . The systematic version is a $(k+r, k)$ code, in which the codeword is $(u_1, \dots, u_k; x_1, \dots, x_r)$.

The rate of the nonsystematic code is

$$R_{n\text{sys}} = \frac{a}{\sum_i f_i}$$

The rate of the systematic code is

$$R_{\text{sys}} = \frac{a}{a + \sum_i f_i}$$

For example, regular repeat and accumulate (RA) codes can be considered nonsystematic IRA codes with $a=1$ and exactly one f_i equal to 1, say $f_q=1$, and the rest zero, in which case $R_{n\text{sys}}$ simplifies to $R=1/q$.

The IRA code may be represented using an alternate notation. Let λ_i be the fraction of edges between the information nodes **302** and the check nodes **304** that are adjacent to an information node of degree i , and let ρ_i be the fraction of such edges that are adjacent to a check node of degree $i+2$ (i.e., one that is adjacent to i information nodes). These edge fractions may be used to represent the IRA code rather than the corresponding node fractions. Define $\lambda(x) = \sum_i \lambda_i x^{i-1}$ and $\rho(x) = \sum_i \rho_i x^{i-1}$ to be the generating functions of these sequences. The pair (λ, ρ) is called a degree distribution. For $L(x) = \sum_i f_i x_i$,

$$f_i = \frac{\lambda_i / i}{\sum_j \lambda_j / j}$$

$$L(x) = \int_0^x \lambda(t) dt / \int_0^1 \lambda(t) dt$$

The rate of the systematic IRA code given by the degree distribution is given by

$$\text{Rate} = \left(1 + \frac{\sum_j \rho_j / j}{\sum_j \lambda_j / j} \right)^{-1}$$

“Belief propagation” on the Tanner Graph realization may be used to decode IRA codes. Roughly speaking, the belief

US 7,116,710 B1

5

propagation decoding technique allows the messages passed on an edge to represent posterior densities on the bit associated with the variable node. A probability density on a bit is a pair of non-negative real numbers $p(0)$, $p(1)$ satisfying $p(0)+p(1)=1$, where $p(0)$ denotes the probability of the bit being 0, $p(1)$ the probability of it being 1. Such a pair can be represented by its log likelihood ratio, $m=\log(p(0)/p(1))$. The outgoing message from a variable node u to a check node v represents information about u , and a message from a check node u to a variable node v represents information about u , as shown in FIGS. 5A and 5B, respectively.

The outgoing message from a node u to a node v depends on the incoming messages from all neighbors w of u except v . If u is a variable message node, this outgoing message is

$$m(u \rightarrow v) = \sum_{w \neq v} m(w \rightarrow u) + m_0(u)$$

where $m_0(u)$ is the log-likelihood message associated with u . If u is a check node, the corresponding formula is

$$\tanh \frac{m(u \rightarrow v)}{2} = \prod_{w \neq v} \tanh \frac{m(w \rightarrow u)}{2}$$

Before decoding, the messages $m(w \rightarrow u)$ and $m(u \rightarrow v)$ are initialized to be zero, and $m_0(u)$ is initialized to be the log-likelihood ratio based on the channel received information. If the channel is memoryless, i.e., each channel output only relies on its input, and y is the output of the channel code bit u , then $m_0(i)=\log(p(u=0|y)/p(u=1|y))$. After this initialization, the decoding process may run in a fully parallel and local manner. In each iteration, every variable/check node receives messages from its neighbors, and sends back updated messages. Decoding is terminated after a fixed number of iterations or detecting that all the constraints are satisfied. Upon termination, the decoder outputs a decoded sequence based on the messages $m(u)=\sum w_m(w \rightarrow u)$.

Thus, on various channels, iterative decoding only differs in the initial messages $m_0(u)$. For example, consider three memoryless channel models: a binary erasure channel (BEC); a binary symmetric channel (BSC); and an additive white Gaussian noise (AGWN) channel.

In the BEC, there are two inputs and three outputs. When 0 is transmitted, the receiver can receive either 0 or an erasure E. An erasure E output means that the receiver does not know how to demodulate the output. Similarly, when 1 is transmitted, the receiver can receive either 1 or E. Thus, for the BEC, $y \in \{0, E, 1\}$, and

$$m_0(u) = \begin{cases} +\infty & \text{if } y = 0 \\ 0 & \text{if } y = E \\ -\infty & \text{if } y = 1 \end{cases}$$

In the BSC, there are two possible inputs (0,1) and two possible outputs (0, 1). The BSC is characterized by a set of

6

conditional probabilities relating all possible outputs to possible inputs. Thus, for the BSC $y \in \{0, 1\}$,

$$m_0(u) = \begin{cases} \log \frac{1-p}{p} & \text{if } y = 0 \\ -\log \frac{1-p}{p} & \text{if } y = 1 \end{cases}$$

and

In the AWGN, the discrete-time input symbols X take their values in a finite alphabet while channel output symbols Y can take any values along the real line. There is assumed to be no distortion or other effects other than the addition of white Gaussian noise. In an AWGN with a Binary Phase Shift Keying (BPSK) signaling which maps 0 to the symbol with amplitude $\sqrt{E_s}$ and 1 to the symbol with amplitude $-\sqrt{E_s}$, output $y \in \mathbb{R}$, then

$$m_0(u) = 4y\sqrt{E_s}N_0$$

where $N_0/2$ is the noise power spectral density.

The selection of a degree profile for use in a particular transmission channel is a design parameter, which may be affected by various attributes of the channel. The criteria for selecting a particular degree profile may include, for example, the type of channel and the data rate on the channel. For example, Table 1 shows degree profiles that have been found to produce good results for an AWGN channel model.

TABLE 1

a	2	3	4
λ_2	0.139025	0.078194	0.054485
λ_3	0.2221555	0.128085	0.104315
λ_5		0.160813	
λ_6	0.638820	0.036178	0.126755
λ_{10}			0.229816
λ_{11}			0.016484
λ_{12}		0.108828	
λ_{13}		0.487902	
λ_{14}			
λ_{16}			
λ_{27}			0.450302
λ_{28}			0.017842
Rate	0.333364	0.333223	0.333218
σ_{GA}	1.1840	1.2415	1.2615
σ^*	1.1981	1.2607	1.2780
(Eb/N0) * (dB)	0.190	-0.250	-0.371
S.L. (dB)	-0.4953	-0.4958	-0.4958

Table 1 shows degree profiles yielding codes of rate approximately $1/3$ for the AWGN channel and with $a=2, 3, 4$. For each sequence, the Gaussian approximation noise threshold, the actual sum-product decoding threshold and the corresponding energy per bit (E_b)-noise power (N_0) ratio in dB are given. Also listed is the Shannon limit (S.L.).

As the parameter “a” is increased, the performance improves. For example, for $a=4$, the best code found has an iterative decoding threshold of $E_b/N_0=-0.371$ dB, which is only 0.12 dB above the Shannon limit.

The accumulator component of the coder may be replaced by a “double accumulator” 600 as shown in FIG. 6. The double accumulator can be viewed as a truncated rate 1 convolutional coder with transfer function $1/(1+D+D^2)$.

Alternatively, a pair of accumulators may be added, as shown in FIG. 7. There are three component codes: the “outer” code 700, the “middle” code 702, and the “inner”

US 7,116,710 B1

7

code **704**. The outer code is an irregular repetition code, and the middle and inner codes are both accumulators.

IRA codes may be implemented in a variety of channels, including memoryless channels, such as the BEC, BSC, and AWGN, as well as channels having non-binary input, non-symmetric and fading channels, and/or channels with memory.

A number of embodiments have been described. Nevertheless, it will be understood that various modifications may be made without departing from the spirit and scope of the invention. Accordingly, other embodiments are within the scope of the following claims.

The invention claimed is:

1. A method of encoding a signal, comprising:
 - obtaining a block of data in the signal to be encoded;
 - partitioning said data block into a plurality of sub-blocks, each sub-block including a plurality of data elements; first encoding the data block to form a first encoded data block, said first encoding including repeating the data elements in different sub-blocks a different number of times;
 - interleaving the repeated data elements in the first encoded data block; and
 - second encoding said first encoded data block using an encoder that has a rate close to one.
2. The method of claim 1, wherein said second encoding is via a rate 1 linear transformation.
3. The method of claim 1, wherein said first encoding is carried out by a first coder with a variable rate less than one, and said second encoding is carried out by a second coder with a rate substantially close to one.
4. The method of claim 3, wherein the second coder comprises an accumulator.
5. The method of claim 4, wherein the data elements comprises bits.
6. The method of claim 5, wherein the first coder comprises a repeater operable to repeat different sub-blocks a different number of times in response to a selected degree profile.
7. The method of claim 4, wherein the first coder comprises a low-density generator matrix coder and the second coder comprises an accumulator.
8. The method of claim 1, wherein the second encoding uses a transfer function of $1/(1+D)$.
9. The method of claim 1, wherein the second encoding uses a transfer function of $1/(1+D+D^2)$.
10. The method of claim 1, wherein said second encoding utilizes two accumulators.
11. A method of encoding a signal, comprising:
 - receiving a block of data in the signal to be encoded, the data block including a plurality of bits;
 - first encoding the data block such that each bit in the data block is repeated and two or more of said plurality of bits are repeated a different number of times in order to form a first encoded data block; and
 - second encoding the first encoded data block in such a way that bits in the first encoded data block are accumulated.
12. The method of claim 11, wherein the said second encoding is via a rate 1 linear transformation.
13. The method of claim 11, wherein the first encoding is via a low-density generator matrix transformation.
14. The method of claim 11, wherein the signal to be encoded comprises a plurality of data blocks of fixed size.

8

15. A coder comprising:

a first coder having an input configured to receive a stream of bits, said first coder operative to repeat said stream of bits irregularly and scramble the repeated bits; and
 a second coder operative to further encode bits output from the first coder at a rate within 10% of one.

16. The coder of claim 15, wherein the stream of bits includes a data block, and wherein the first coder is operative to apportion said data block into a plurality of sub-blocks and to repeat bits in each sub-block a number of times, wherein bits in different sub-blocks are repeated a different number of times.

17. The coder of claim 16, wherein the second coder comprises a recursive convolutional encoder with a transfer function of $1/(1+D)$.

18. The coder of claim 16, wherein the second coder comprises a recursive convolutional encoder with a transfer function of $1/(1+D+D^2)$.

19. The coder of claim 15, wherein the first coder comprises a repeater having a variable rate and an interleaver.

20. The coder of claim 15, wherein the first coder comprises a low-density generator matrix coder.

21. The coder of claim 15, wherein the second coder comprises a rate 1 linear encoder.

22. The coder of claim 21, wherein the second coder comprises an accumulator.

23. The coder of claim 22, wherein the second coder further comprises a second accumulator.

24. The coder of claim 15, wherein the second coder comprises a coder operative to further encode bits output from the first coder at a rate within 1% of one.

25. A coding system comprising:

a first coder having an input configured to receive a stream of bits, said first coder operative to repeat said stream of bits irregularly and scramble the repeated bits;

a second coder operative to further encode bits output from the first coder at a rate within 10% of one in order to form an encoded data stream; and

a decoder operative to receive the encoded data stream and decode the encoded data stream using an iterative decoding technique.

26. The coding system of claim 25, wherein the first coder comprises a repeater operative to receive a data block including a plurality of bits from said stream of bits and to repeat bits in the data block a different number of times according to a selected degree profile.

27. The coding system of claim 26, wherein the first coder comprises an interleaver.

28. The coding system of claim 25, wherein the first coder comprises a low-density generator matrix coder.

29. The coding system of claim 25, wherein the second coder comprises a rate 1 accumulator.

30. The coding system of claim 25, wherein the decoder is operative to decode the encoded data stream using a posterior decoding techniques.

31. The coding system of claim 25, wherein the decoder is operative to decode the encoded data stream based on a Tanner graph representation.

32. The coding system of claim 25, wherein the decoder is operative to decode the encoded data stream in linear time.

33. The coding system of claim 25, wherein the second coder comprises a coder operative to further encode bits output from the first coder at a rate within 1% of one.

* * * * *

UNITED STATES PATENT AND TRADEMARK OFFICE
CERTIFICATE OF CORRECTION

PATENT NO. : 7,116,710 B1
APPLICATION NO. : 09/861102
DATED : October 3, 2006
INVENTOR(S) : Hui Jin, Aamod Khandekar and Robert J. McEliece

Page 1 of 1


It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

At column 1, line 8, please amend the paragraph as follows:

This application claims the priority ~~[[to]]~~ of U.S. Provisional Application Ser. No. 60/205,095, filed on May 18, 2000, and ~~[[to]]~~ is a continuation-in-part of U.S. application Ser. No. 09/922,852, filed on Aug. 18, 2000 and entitled Interleaved Serial Concatenation Forming Turbo-Like Codes.

Signed and Sealed this

Twenty-second Day of July, 2008

A handwritten signature in black ink that reads "Jon W. Dudas". The signature is written in a cursive style with a large, stylized initial "J".

JON W. DUDAS
Director of the United States Patent and Trademark Office

UNITED STATES PATENT AND TRADEMARK OFFICE
CERTIFICATE OF CORRECTION

PATENT NO. : 7,116,710 B1
APPLICATION NO. : 09/861102
DATED : October 3, 2006
INVENTOR(S) : Hui Jin, Aamod Khandekar and Robert J. McEliece

Page 1 of 1

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

In the Specification

Column 1, Line 8:

“This application claims the priority of U.S. Provisional Application Ser. No. 60/205,095, filed on May 18, 2000, and is a continuation-in-part of U.S. application Ser. No. 09/922,852, filed on Aug. 18, 2000 and entitled Interleaved Serial Concatenation Forming Turbo-Like Codes.”

Should read:

-- This application claims the priority of U.S. Provisional Application Ser. No. 60/205,095, filed on May 18, 2000. --

Signed and Sealed this
Fifth Day of July, 2022



Katherine Kelly Vidal
Director of the United States Patent and Trademark Office

(12) **INTER PARTES REVIEW CERTIFICATE** (1909th)

**United States Patent
Jin et al.**

(10) **Number:** US 7,116,710 K1

(45) **Certificate Issued:** Feb. 16, 2021

(54) **SERIAL CONCATENATION OF
INTERLEAVED CONVOLUTIONAL CODES
FORMING TURBO-LIKE CODES**

(75) **Inventors:** Hui Jin; Aamod Khandekar; Robert
J. McEliece

(73) **Assignee:** CALIFORNIA INSTITUTE OF
TECHNOLOGY

Trial Numbers:

IPR2017-00210 filed Nov. 15, 2016

IPR2017-00219 filed Nov. 15, 2016

Inter Partes Review Certificate for:

Patent No.: 7,116,710

Issued: Oct. 3, 2006

Appl. No.: 09/861,102

Filed: May 18, 2001

The results of IPR2017-00210 and IPR2017-00219 are reflected in this inter partes review certificate under 35 U.S.C. 318(b).

INTER PARTES REVIEW CERTIFICATE

U.S. Patent 7,116,710 K1

Trial No. IPR2017-00210

Certificate Issued Feb. 16, 2021

1

2

AS A RESULT OF THE INTER PARTES
REVIEW PROCEEDING, IT HAS BEEN
DETERMINED THAT:

Claims **1-8, 11-17, 19-22** and **24-33** are found patentable. ⁵

* * * * *

EXHIBIT B



US007421032B2

(12) **United States Patent**
Jin et al.

(10) **Patent No.:** **US 7,421,032 B2**
(45) **Date of Patent:** **Sep. 2, 2008**

(54) **SERIAL CONCATENATION OF INTERLEAVED CONVOLUTIONAL CODES FORMING TURBO-LIKE CODES**

(75) Inventors: **Hui Jin**, Glen Gardner, NJ (US); **Aamod Khandekar**, Pasadena, CA (US); **Robert J. McEliece**, Pasadena, CA (US)

(73) Assignee: **Callifornia Institute of Technology**, Pasadena, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **11/542,950**

(22) Filed: **Oct. 3, 2006**

(65) **Prior Publication Data**

US 2007/0025450 A1 Feb. 1, 2007

Related U.S. Application Data

(63) Continuation of application No. 09/861,102, filed on May 18, 2001, now Pat. No. 7,116,710, and a continuation-in-part of application No. 09/922,852, filed on Aug. 18, 2000, now Pat. No. 7,089,477.

(60) Provisional application No. 60/205,095, filed on May 18, 2000.

(51) **Int. Cl.**
H04L 5/12 (2006.01)

(52) **U.S. Cl.** **375/262; 375/265; 375/348; 714/755; 714/786; 714/792; 341/52; 341/102**

(58) **Field of Classification Search** **375/259, 375/262, 265, 285, 296, 341, 346, 348; 714/746, 714/752, 755, 756, 786, 792, 794-796; 341/51, 341/52, 56, 102, 103**

See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,392,299 A	2/1995	Rhines et al.	
5,530,707 A *	6/1996	Lin	714/792
5,751,739 A	5/1998	Seshadri et al.	
5,802,115 A *	9/1998	Meyer	375/341
5,881,093 A	3/1999	Wang et al.	
6,014,411 A	1/2000	Wang	
6,023,783 A	2/2000	Divsalar et al.	
6,031,874 A	2/2000	Chennakeshu et al.	
6,032,284 A	2/2000	Bliss	
6,044,116 A	3/2000	Wang	
6,094,739 A *	7/2000	Miller et al.	714/792

(Continued)

OTHER PUBLICATIONS

Appendix A.1 "Structure of Parity Check Matrices of Standardized LDPC Codes," Digital Video Broadcasting (DVB) User guidelines for the second generation system for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications (DVB-S2) ETSI TR 102 376 V1.1.1. (Feb. 2005) Technical Report, pp. 64.

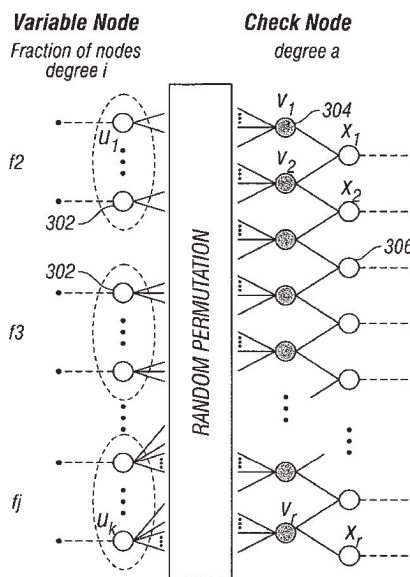
(Continued)

Primary Examiner—Dac V. Ha
(74) *Attorney, Agent, or Firm*—Fish & Richardson P.C.

(57) **ABSTRACT**

A serial concatenated coder includes an outer coder and an inner coder. The outer coder irregularly repeats bits in a data block according to a degree profile and scrambles the repeated bits. The scrambled and repeated bits are input to an inner coder, which has a rate substantially close to one.

23 Claims, 5 Drawing Sheets



US 7,421,032 B2

Page 2

U.S. PATENT DOCUMENTS

6,396,423	B1	5/2002	Laumen et al.	
6,437,714	B1	8/2002	Kim et al.	
6,859,906	B2 *	2/2005	Hammons et al.	714/786
2001/0025358	A1	9/2001	Eidson et al.	

OTHER PUBLICATIONS

Benedetto et al., "A Soft-Input Soft-Output Maximum A Posteriori (MAP) Module to Decode Parallel and Serial Concatenated Codes," The Telecommunications and Data Acquisition (TDA) Progress Report 42-127 for NASA and California Institute of Technology Jet Propulsion Laboratory, Joseph H. Yuen, Ed., pp. 1-20 (Nov. 15, 1996).

Benedetto et al., "Bandwidth efficient parallel concatenated coding schemes," *Electronics Letters* 31(24): 2067-2069 (Nov. 23, 1995).

Benedetto et al., "Design of Serially Concatenated Interleaved Codes," ICC 97, Montreal, Canada, pp. 710-714, (Jun. 1997).

Benedetto et al., "Parallel Concatenated Trellis Coded Modulation," ICC '96, IEEE, pp. 974-978, (Jun. 1996).

Benedetto et al., "Serial Concatenated Trellis Coded Modulation with Iterative Decoding," Proceedings from the IEEE 1997 International Symposium on Information Theory (ISIT), Ulm, Germany, p. 8, Jun. 29-Jul. 4, 1997.

Benedetto et al., "Serial Concatenation of Interleaved Codes: Performance Analysis, Design, and Iterative Decoding," The Telecommunications and Data Acquisition (TDA) Progress Report 42-126 for NASA and California Institute of Technology Jet Propulsion Laboratory, Joseph H. Yuen, Ed., pp. 1-26 (Aug. 15, 1996).

Benedetto et al., "Serial Concatenation of interleaved codes: performance analysis, design, and iterative decoding," Proceedings from the IEEE 1997 International Symposium on Information Theory (ISIT), Ulm, Germany, p. 106, Jun. 29-Jul. 4, 1997.

Benedetto et al., "Soft-output decoding algorithms in iterative decoding of turbo codes," The Telecommunications and Data Acquisition (TDA) Progress Report 42-124 for NASA and California Institute of Technology Jet Propulsion Laboratory, Joseph H. Yuen, Ed., pp. 63-87 (Feb. 15, 1996).

Benedetto, S. et al., "A Soft-Input Soft-Output APP Module for Iterative Decoding of Concatenated Codes," *IEEE Communications Letters* 1(1): 22-24 (Jan. 1997).

Berrou et al., "Near Shannon Limit Error-Correcting Coding and Decoding: Turbo Codes," ICC pp. 1064-1070 (1993).

Digital Video Broadcasting (DVB) User guidelines for the second generation system for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications (DVB-S2) ETSI TR 102 376 V1.1.1. (Feb. 2005) Technical Report, pp. 1-104 (Feb. 15, 2005).

Divsalar et al., "Coding Theorems for 'Turbo-Like' Codes," Proceedings of the 36th Annual Allerton Conference on Communication, Control, and Computing, Sep. 23-25, 1998, Allerton House, Monticello, Illinois, pp. 201-210 (1998).

Divsalar et al., "Effective free distance of turbo codes," *Electronics Letters* 32(5): 445-446 (Feb. 29, 1996).

Divsalar, D. et al., "Hybrid Concatenated Codes and Iterative Decoding," Proceedings from the IEEE 1997 International Symposium on Information Theory (ISIT), Ulm, Germany, p. 10 (Jun. 29-Jul. 4, 1997).

Divsalar, D. et al., "Low-rate turbo codes for Deep Space Communications," Proceedings from the 1995 IEEE International Symposium on Information Theory, Sep. 17-22, 1995, Whistler, British Columbia, Canada, pp. 35.

Divsalar, D. et al., "Multiple Turbo Codes for Deep-Space Communications," The Telecommunications and Data Acquisition (TDA) Progress Report 42-121 for NASA and California Institute of Technology Jet Propulsion Laboratory, Joseph H. Yuen, Ed., pp. 60-77 (May 15, 1995).

Divsalar, D. et al., "Multiple Turbo Codes," MILCOM95, San Diego, CA pp. 279-285 (Nov. 5-6, 1995).

Divsalar, D. et al., "On the Design of Turbo Codes," The Telecommunications and Data Acquisition (TDA) Progress Report 42-123 for NASA and California Institute of Technology Jet Propulsion Laboratory, Joseph H. Yuen, Ed., pp. 99-131 (Nov. 15, 1995).

Divsalar, D. et al., "Serial Turbo Trellis Coded Modulation with Rate-1 Inner Code," Proceedings from the IEEE 2000 International Symposium on Information Theory (ISIT), Italy, pp. 1-14 (Jun. 2000).

Divsalar, D. et al., "Turbo Codes for PCS Applications," ICC 95, IEEE, Seattle, WA, pp. 54-59 (Jun. 1995).

Jin et al., "Irregular Repeat—Accumulate Codes," 2nd International Symposium on Turbo Codes & Related Topics, Sep. 4-7, 2000, Brest, France, 25 slides, (presented on Sep. 4, 2000).

Jin et al., "Irregular Repeat—Accumulate Codes," 2nd International Symposium on Turbo Codes & Related Topics, Sep. 4-7, 2000, Brest, France, pp. 1-8 (2000).

Richardson et al., "Design of capacity approaching irregular low density parity check codes," *IEEE Trans. Inform. Theory* 47: 619-637 (Feb. 2001).

Richardson, T. and R. Urbanke, "Efficient encoding of low-density parity check codes," *IEEE Trans. Inform. Theory* 47: 638-656 (Feb. 2001).

Wilberg, et al., "Codes and Iterative Decoding on General Graphs", 1995 Intl. Symposium on Information Theory, Sep. 1995, p. 468.

* cited by examiner

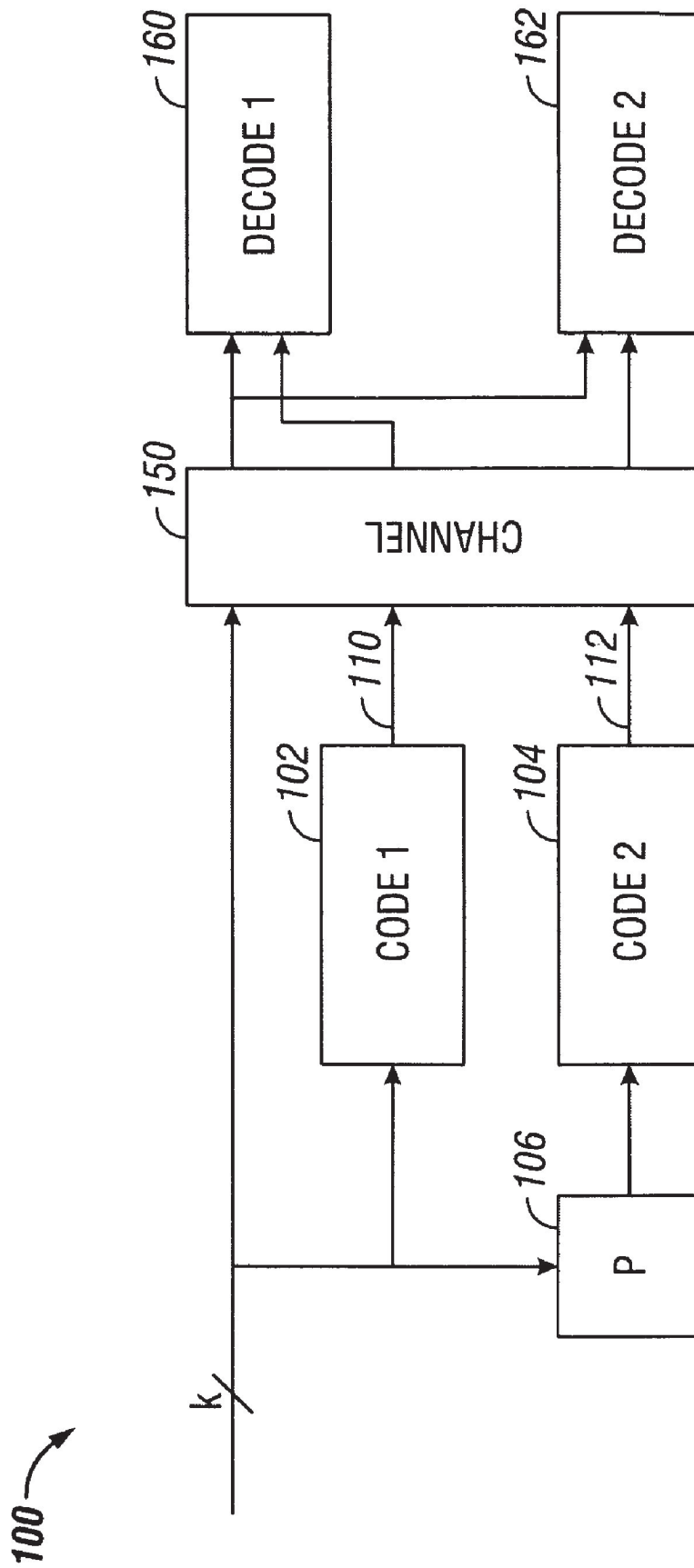


FIG. 1
(Prior Art)

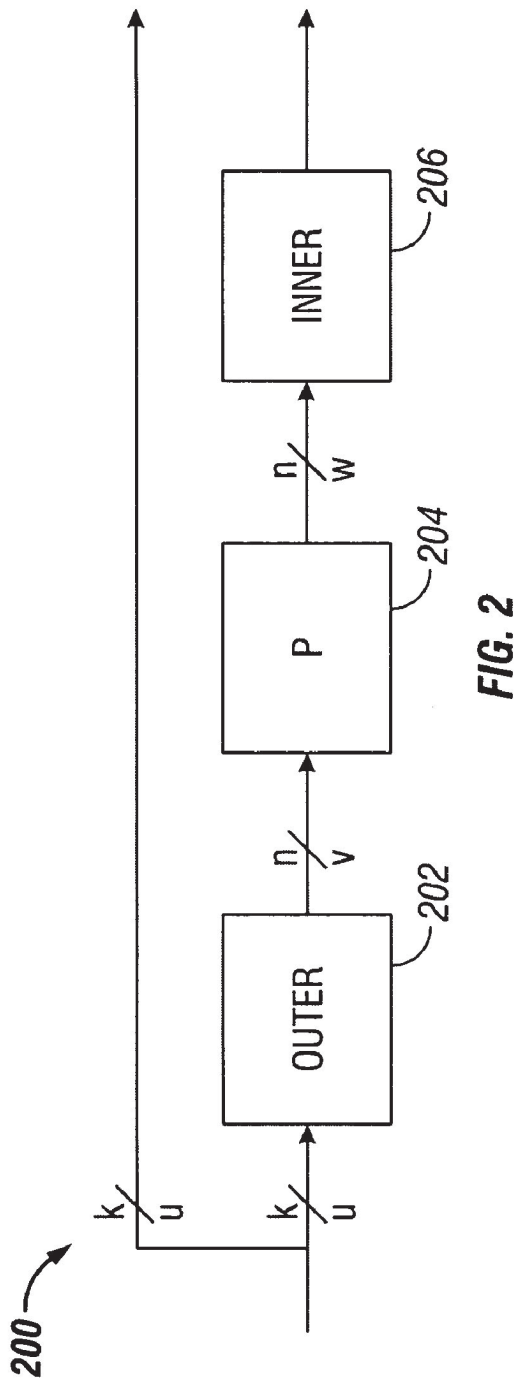


FIG. 2

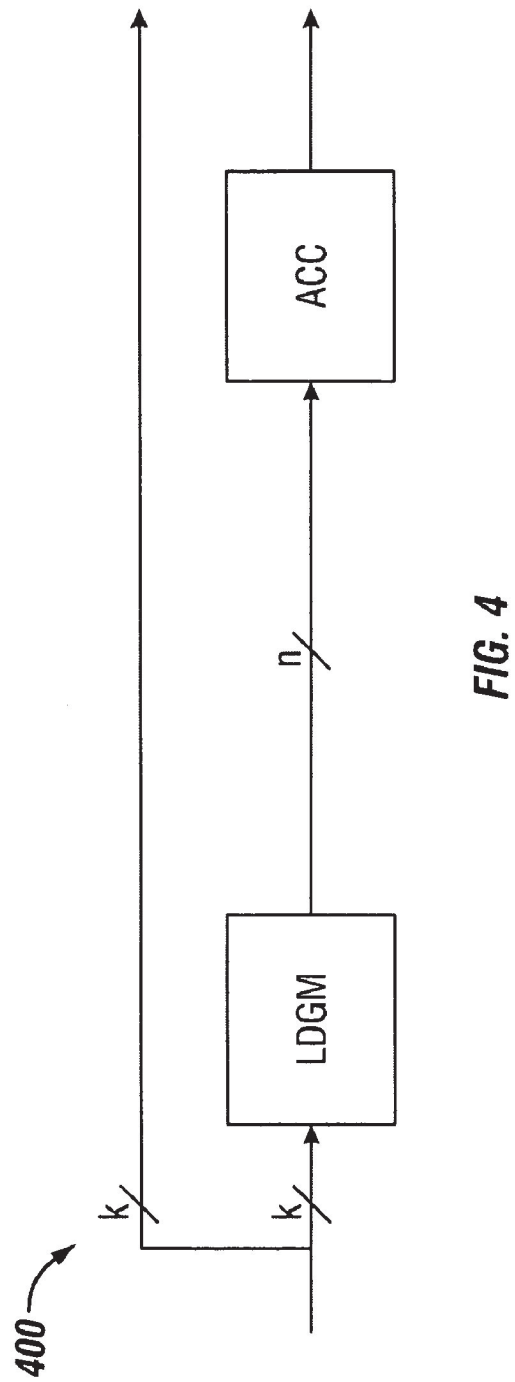


FIG. 4

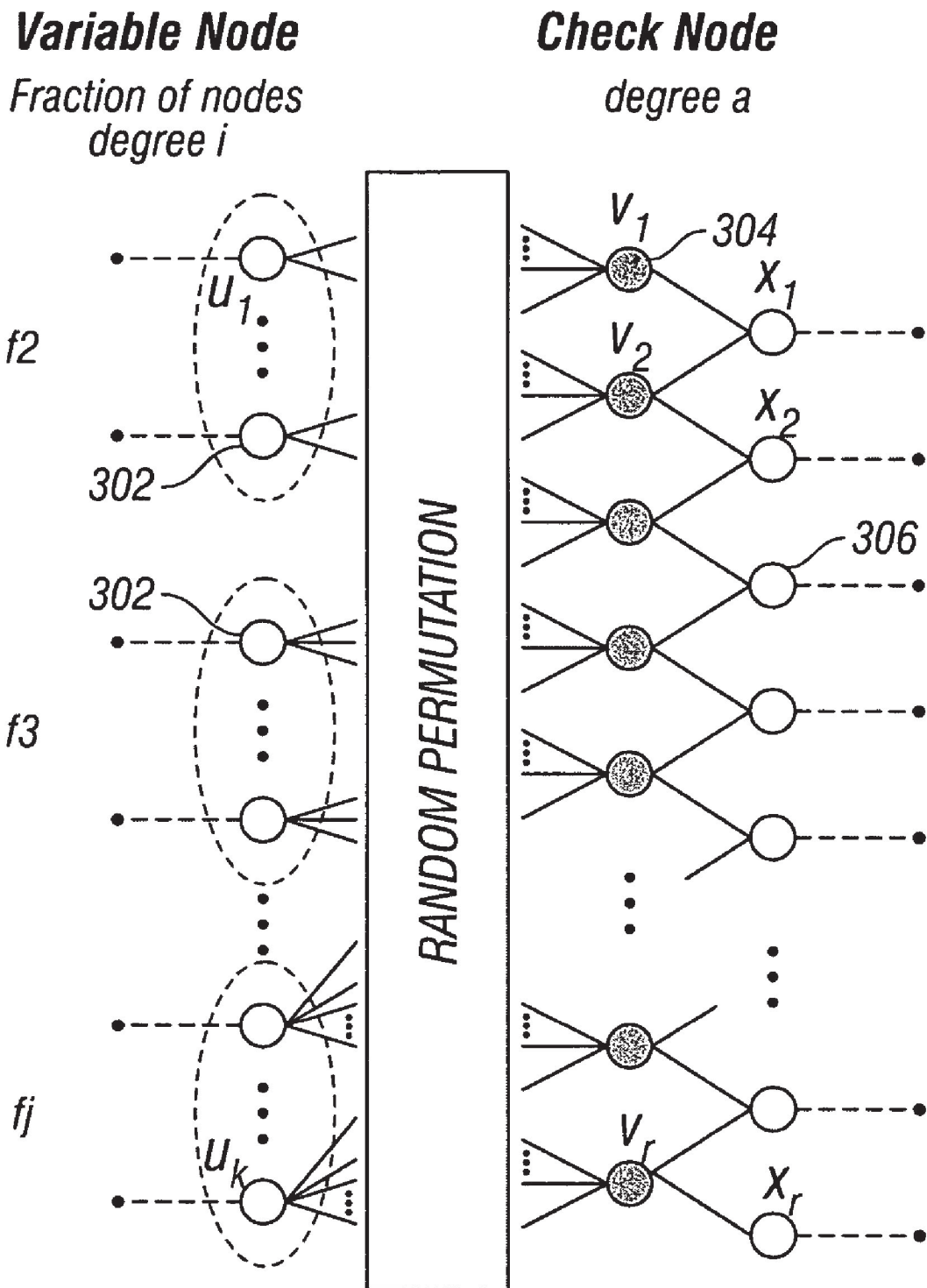


FIG. 3

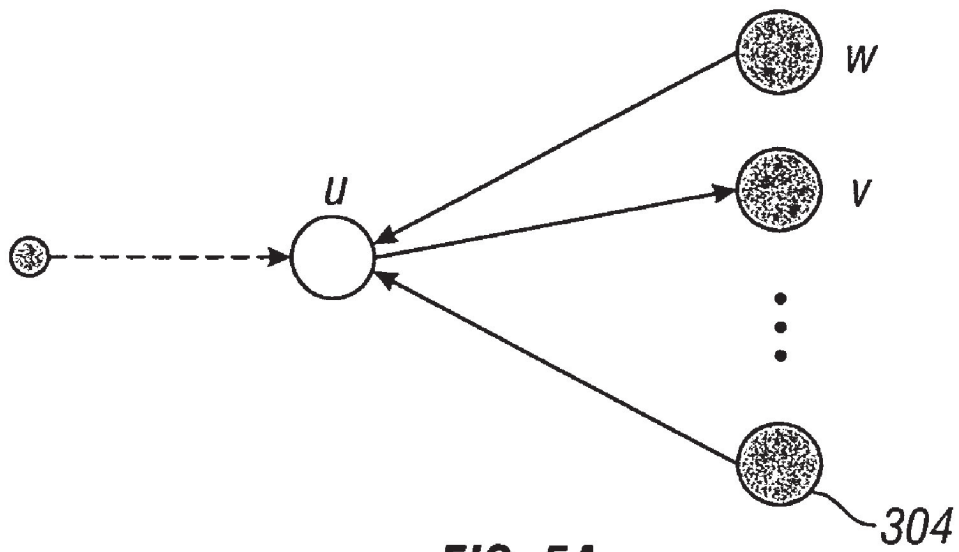


FIG. 5A

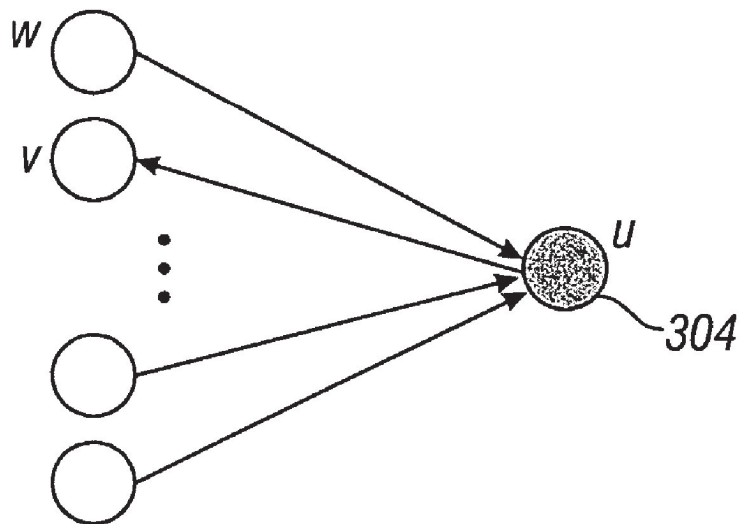


FIG. 5B

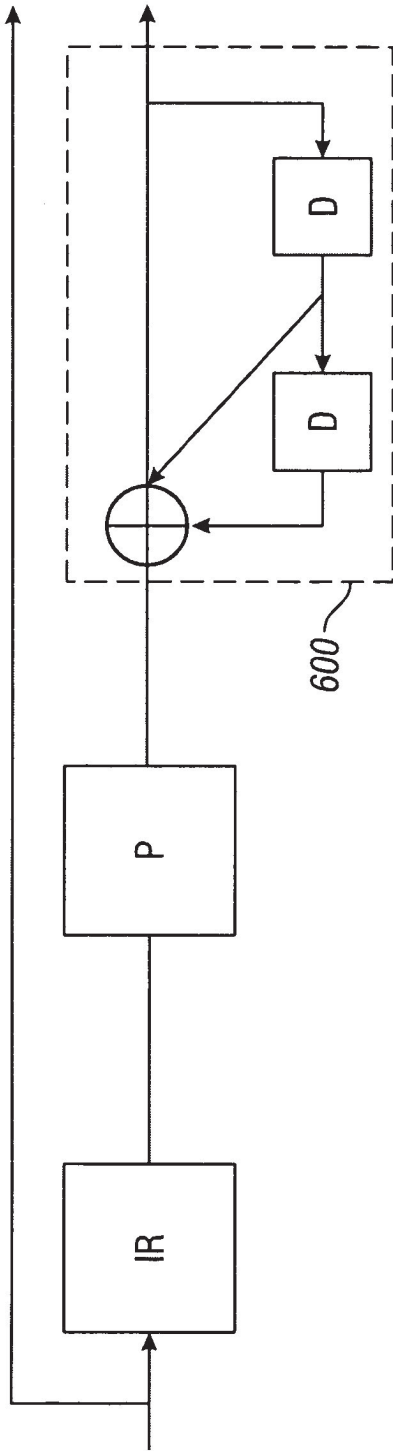


FIG. 6

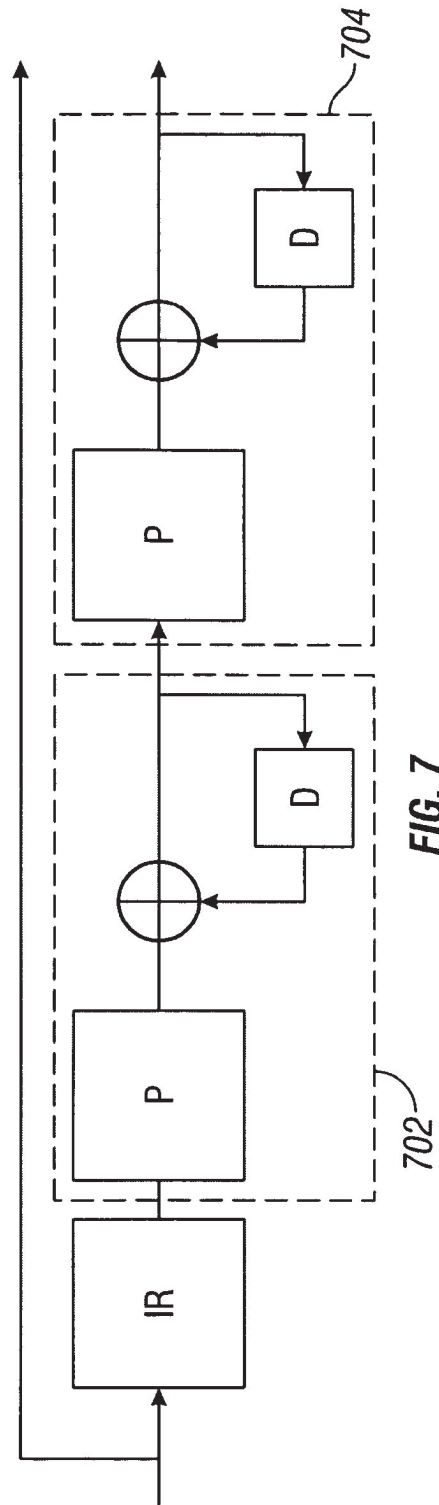


FIG. 7

US 7,421,032 B2

1

SERIAL CONCATENATION OF INTERLEAVED CONVOLUTIONAL CODES FORMING TURBO-LIKE CODES

CROSS-REFERENCE TO RELATED APPLICATIONS

This application is a continuation of U.S. application Ser. No. 09/861,102, filed May 18, 2001, now U.S. Pat. No. 7,116,710, which claims the priority of U.S. provisional application Ser. No. 60/205,095, filed May 18, 2000, and is a continuation-in-part of U.S. application Ser. No. 09/922,852, filed Aug. 18, 2000, now U.S. Pat. No. 7,089,477.

GOVERNMENT LICENSE RIGHTS

The U.S. Government has a paid-up license in this invention and the right in limited circumstances to require the patent owner to license others on reasonable terms as provided for by the terms of Grant No. CCR-9804793 awarded by the National Science Foundation.

BACKGROUND

Properties of a channel affect the amount of data that can be handled by the channel. The so-called "Shannon limit" defines the theoretical limit of the amount of data that a channel can carry.

Different techniques have been used to increase the data rate that can be handled by a channel. "Near Shannon Limit Error-Correcting Coding and Decoding: Turbo Codes," by Berrou et al. ICC, pp 1064-1070, (1993), described a new "turbo code" technique that has revolutionized the field of error correcting codes. Turbo codes have sufficient randomness to allow reliable communication over the channel at a high data rate near capacity. However, they still retain sufficient structure to allow practical encoding and decoding algorithms. Still, the technique for encoding and decoding turbo codes can be relatively complex.

A standard turbo coder **100** is shown in FIG. **1**. A block of k information bits is input directly to a first coder **102**. A k bit interleaver **106** also receives the k bits and interleaves them prior to applying them to a second coder **104**. The second coder produces an output that has more bits than its input, that is, it is a coder with rate that is less than 1. The coders **102**, **104** are typically recursive convolutional coders.

Three different items are sent over the channel **150**: the original k bits, first encoded bits **110**, and second encoded bits **112**. At the decoding end, two decoders are used: a first constituent decoder **160** and a second constituent decoder **162**. Each receives both the original k bits, and one of the encoded portions **110**, **112**. Each decoder sends likelihood estimates of the decoded bits to the other decoders. The estimates are used to decode the uncoded information bits as corrupted by the noisy channel.

SUMMARY

A coding system according to an embodiment is configured to receive a portion of a signal to be encoded, for example, a data block including a fixed number of bits. The coding system includes an outer coder, which repeats and scrambles bits in the data block. The data block is apportioned into two or more sub-blocks, and bits in different sub-blocks are repeated a different number of times according to a selected degree profile. The outer coder may include a

2

repeater with a variable rate and an interleaver. Alternatively, the outer coder may be a low-density generator matrix (LDGM) coder.

The repeated and scrambled bits are input to an inner coder that has a rate substantially close to one. The inner coder may include one or more accumulators that perform recursive modulo two addition operations on the input bit stream.

The encoded data output from the inner coder may be transmitted on a channel and decoded in linear time at a destination using iterative decoding techniques. The decoding techniques may be based on a Tanner graph representation of the code.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** is a schematic diagram of a prior "turbo code" system.

FIG. **2** is a schematic diagram of a coder according to an embodiment.

FIG. **3** is a Tanner graph for an irregular repeat and accumulate (IRA) coder.

FIG. **4** is a schematic diagram of an IRA coder according to an embodiment.

FIG. **5A** illustrates a message from a variable node to a check node on the Tanner graph of FIG. **3**.

FIG. **5B** illustrates a message from a check node to a variable node on the Tanner graph of FIG. **3**.

FIG. **6** is a schematic diagram of a coder according to an alternate embodiment.

FIG. **7** is a schematic diagram of a coder according to another alternate embodiment.

DETAILED DESCRIPTION

FIG. **2** illustrates a coder **200** according to an embodiment. The coder **200** may include an outer coder **202**, an interleaver **204**, and inner coder **206**. The coder may be used to format blocks of data for transmission, introducing redundancy into the stream of data to protect the data from loss due to transmission errors. The encoded data may then be decoded at a destination in linear time at rates that may approach the channel capacity.

The outer coder **202** receives the uncoded data. The data may be partitioned into blocks of fixed size, say k bits. The outer coder may be an (n,k) binary linear block coder, where $n > k$. The coder accepts as input a block u of k data bits and produces an output block v of n data bits. The mathematical relationship between u and v is $v = T_0 u$, where T_0 is an $n \times k$ matrix, and the rate of the coder is k/n .

The rate of the coder may be irregular, that is, the value of T_0 is not constant, and may differ for sub-blocks of bits in the data block. In an embodiment, the outer coder **202** is a repeater that repeats the k bits in a block a number of times q to produce a block with n bits, where $n = qk$. Since the repeater has an irregular output, different bits in the block may be repeated a different number of times. For example, a fraction of the bits in the block may be repeated two times, a fraction of bits may be repeated three times, and the remainder of bits may be repeated four times. These fractions define a degree sequence, or degree profile, of the code.

The inner coder **206** may be a linear rate-1 coder, which means that the n -bit output block x can be written as $x = T_1 w$, where T_1 is a nonsingular $n \times n$ matrix. The inner coder **210** can have a rate that is close to 1, e.g., within 50%, more preferably 10% and perhaps even more preferably within 1% of 1.

In an embodiment, the inner coder **206** is an accumulator, which produces outputs that are the modulo two (mod-2)

3

partial sums of its inputs. The accumulator may be a truncated rate-1 recursive convolutional coder with the transfer function $1/(1+D)$. Such an accumulator may be considered a block coder whose input block $[x_1, \dots, x_n]$ and output block $[y_1, \dots, y_n]$ are related by the formula

$$\begin{aligned}
 y_1 &= x_1 \\
 y_2 &= x_1 \oplus x_2 \\
 y_3 &= x_1 \oplus x_2 \oplus x_3 \\
 &\vdots \\
 &\vdots \\
 &\vdots \\
 y_n &= x_1 \oplus x_2 \oplus x_3 \oplus \dots \oplus x_n.
 \end{aligned}$$

where “ \oplus ” denotes mod-2, or exclusive-OR (XOR), addition. An advantage of this system is that only mod-2 addition is necessary for the accumulator. The accumulator may be embodied using only XOR gates, which may simplify the design.

The bits output from the outer coder 202 are scrambled before they are input to the inner coder 206. This scrambling may be performed by the interleaver 204, which performs a pseudo-random permutation of an input block v , yielding an output block w having the same length as v .

The serial concatenation of the interleaved irregular repeat code and the accumulate code produces an irregular repeat and accumulate (IRA) code. An IRA code is a linear code, and as such, may be represented as a set of parity checks. The set of parity checks may be represented in a bipartite graph, called the Tanner graph, of the code. FIG. 3 shows a Tanner graph 300 of an IRA code with parameters $(f_1, \dots, f_j; a)$, where $f_i \geq 0$, $\sum_i f_i = 1$ and “ a ” is a positive integer. The Tanner graph includes two kinds of nodes: variable nodes (open circles) and check nodes (filled circles). There are k variable nodes 302 on the left, called information nodes. There are r variable nodes 306 on the right, called parity nodes. There are $r = (k \sum_i f_i) / a$ check nodes 304 connected between the information nodes and the parity nodes. Each information node 302 is connected to a number of check nodes 304. The fraction of information nodes connected to exactly i check nodes is f_i . For example, in the Tanner graph 300, each of the f_2 information nodes are connected to two check nodes, corresponding to a repeat of $q=2$, and each of the f_3 information nodes are connected to three check nodes, corresponding to $q=3$.

Each check node 304 is connected to exactly “ a ” information nodes 302. In FIG. 3, $a=3$. These connections can be made in many ways, as indicated by the arbitrary permutation of the ra edges joining information nodes 302 and check nodes 304 in permutation block 310. These connections correspond to the scrambling performed by the interleaver 204.

In an alternate embodiment, the outer coder 202 may be a low-density generator matrix (LDGM) coder that performs an irregular repeat of the k bits in the block, as shown in FIG. 4. As the name implies, an LDGM code has a sparse (low-density) generator matrix. The IRA code produced by the coder 400 is a serial concatenation of the LDGM code and the accumulator code. The interleaver 204 in FIG. 2 may be excluded due to the randomness already present in the structure of the LDGM code.

If the permutation performed in permutation block 310 is fixed, the Tanner graph represents a binary linear block code with k information bits (u_1, \dots, u_k) and r parity bits

4

(x_1, \dots, x_r) , as follows. Each of the information bits is associated with one of the information nodes 302, and each of the parity bits is associated with one of the parity nodes 306. The value of a parity bit is determined uniquely by the condition that the mod-2 sum of the values of the variable nodes connected to each of the check nodes 304 is zero. To see this, set $x_0=0$. Then if the values of the bits on the edges coming out the permutation box are (v_1, \dots, v_{ra}) , then we have the recursive formula

$$x_j = x_{j-1} + \sum_{i=1}^{\lambda} v_{(j-1)\lambda+i}$$

for $j=1, 2, \dots, r$. This is in effect the encoding algorithm.

Two types of IRA codes are represented in FIG. 3, a nonsystematic version and a systematic version. The nonsystematic version is an (r,k) code, in which the codeword corresponding to the information bits (u_1, \dots, u_k) is (x_1, \dots, x_r) . The systematic version is a $(k+r, k)$ code, in which the codeword is $(u_1, \dots, u_k; x_1, \dots, x_r)$.

The rate of the nonsystematic code is

$$R_{n\text{sys}} = \frac{a}{\sum_i f_i}$$

The rate of the systematic code is

$$R_{\text{sys}} = \frac{a}{a + \sum_i f_i}$$

For example, regular repeat and accumulate (RA) codes can be considered nonsystematic IRA codes with $a=1$ and exactly one f_i equal to 1, say $f_q=1$, and the rest zero, in which case $R_{n\text{sys}}$ simplifies to $R=1/q$.

The IRA code may be represented using an alternate notation. Let λ_i be the fraction of edges between the information nodes 302 and the check nodes 304 that are adjacent to an information node of degree i , and let ρ_i be the fraction of such edges that are adjacent to a check node of degree $i+2$ (i.e., one that is adjacent to i information nodes). These edge fractions may be used to represent the IRA code rather than the corresponding node fractions. Define $\lambda(x) = \sum_i \lambda_i x^{i-1}$ and $\rho(x) = \sum_i \rho_i x^{i-1}$ to be the generating functions of these sequences. The pair (λ, ρ) is called a degree distribution. For $L(x) = \sum_i f_i x_i$,

$$f_i = \frac{\lambda_i / i}{\sum_j \lambda_j / j}$$

$$L(x) = \int_0^x \lambda(t) dt / \int_0^1 \lambda(t) dt$$

US 7,421,032 B2

5

The rate of the systematic IRA code given by the degree distribution is given by

$$\text{Rate} = \left(1 + \frac{\sum_j \rho_j / j}{\sum_j \lambda_j / j} \right)^{-1}$$

“Belief propagation” on the Tanner Graph realization may be used to decode IRA codes. Roughly speaking, the belief propagation decoding technique allows the messages passed on an edge to represent posterior densities on the bit associated with the variable node. A probability density on a bit is a pair of non-negative real numbers $p(0)$, $p(1)$ satisfying $p(0) + p(1) = 1$, where $p(0)$ denotes the probability of the bit being 0, $p(1)$ the probability of it being 1. Such a pair can be represented by its log likelihood ratio, $m = \log(p(0)/p(1))$. The outgoing message from a variable node u to a check node v represents information about u , and a message from a check node u to a variable node v represents information about u , as shown in FIGS. 5A and 5B, respectively.

The outgoing message from a node u to a node v depends on the incoming messages from all neighbors w of u except v . If u is a variable message node, this outgoing message is

$$m(u \rightarrow v) = \sum_{w \neq v} m(w \rightarrow u) + m_0(u)$$

where $m_0(u)$ is the log-likelihood message associated with u . If u is a check node, the corresponding formula is

$$\tanh \frac{m(u \rightarrow v)}{2} = \prod_{w \neq v} \tanh \frac{m(w \rightarrow u)}{2}$$

Before decoding, the messages $m(w \rightarrow u)$ and $m(u \rightarrow v)$ are initialized to be zero, and $m_0(u)$ is initialized to be the log-likelihood ratio based on the channel received information. If the channel is memoryless, i.e., each channel output only

6

relies on its input, and y is the output of the channel code bit u , then $m_0(u) = \log(p(u=0|y)/p(u=1|y))$. After this initialization, the decoding process may run in a fully parallel and local manner. In each iteration, every variable/check node receives messages from its neighbors, and sends back updated messages. Decoding is terminated after a fixed number of iterations or detecting that all the constraints are satisfied. Upon termination, the decoder outputs a decoded sequence based on the messages $m(u) = \sum_w m(w \rightarrow u)$.

Thus, on various channels, iterative decoding only differs in the initial messages $m_0(u)$. For example, consider three memoryless channel models: a binary erasure channel (BEC); a binary symmetric channel (BSC); and an additive white Gaussian noise (AWGN) channel.

In the BEC, there are two inputs and three outputs. When 0 is transmitted, the receiver can receive either 0 or an erasure E. An erasure E output means that the receiver does not know how to demodulate the output. Similarly, when 1 is transmitted, the receiver can receive either 1 or E. Thus, for the BEC, $y \in \{0, E, 1\}$, and

$$m_0(u) = \begin{cases} +\infty & \text{if } y = 0 \\ 0 & \text{if } y = E \\ -\infty & \text{if } y = 1 \end{cases}$$

In the BSC, there are two possible inputs (0,1) and two possible outputs (0, 1). The BSC is characterized by a set of conditional probabilities relating all possible outputs to possible inputs. Thus, for the BSC $y \in \{0, 1\}$,

$$m_0(u) = \begin{cases} \log \frac{1-p}{p} & \text{if } y = 0 \\ -\log \frac{1-p}{p} & \text{if } y = 1 \end{cases}$$

and

In the AWGN, the discrete-time input symbols X take their values in a finite alphabet while channel output symbols Y can take any values along the real line. There is assumed to be no distortion or other effects other than the addition of white Gaussian noise. In an AWGN with a Binary Phase Shift Keying (BPSK) signaling which maps 0 to the symbol with amplitude $\sqrt{E_s}$ and 1 to the symbol with amplitude $-\sqrt{E_s}$, output $y \in \mathbb{R}$, then

$$m_0(u) = 4y\sqrt{E_s}/N_0$$

where $N_0/2$ is the noise power spectral density.

The selection of a degree profile for use in a particular transmission channel is a design parameter, which may be affected by various attributes of the channel. The criteria for selecting a particular degree profile may include, for example, the type of channel and the data rate on the channel. For example, Table 1 shows degree profiles that have been found to produce good results for an AWGN channel model.

TABLE 1

a	2	3	4
λ_2	0.139025	0.078194	0.054485
λ_3	0.2221555	0.128085	0.104315
λ_5		0.160813	
λ_6	0.638820	0.036178	0.126755
λ_{10}			0.229816
λ_{11}			0.016484
λ_{12}		0.108828	
λ_{13}		0.487902	
λ_{14}			
λ_{16}			
λ_{27}			0.450302
λ_{28}			0.017842
Rate	0.333364	0.333223	0.333218
σ_{GA}	1.1840	1.2415	1.2615
σ^*	1.1981	1.2607	1.2780
(Eb/No) * (dB)	0.190	-0.250	-0.371
S.L. (dB)	-0.4953	-0.4958	-0.4958

Table 1 shows degree profiles yielding codes of rate approximately 1/3 for the AWGN channel and with a=2, 3, 4. For each sequence, the Gaussian approximation noise threshold, the actual sum-product decoding threshold and the corresponding energy per bit (E_b)-noise power (N₀) ratio in dB are given. Also listed is the Shannon limit (S.L.).

As the parameter “a” is increased, the performance improves. For example, for a=4, the best code found has an iterative decoding threshold of E_b/N₀=-0.371 dB, which is only 0.12 dB above the Shannon limit.

The accumulator component of the coder may be replaced by a “double accumulator” 600 as shown in FIG. 6. The double accumulator can be viewed as a truncated rate 1 convolutional coder with transfer function 1/(1+D+D²).

Alternatively, a pair of accumulators may be added, as shown in FIG. 7. There are three component codes: the “outer” code 700, the “middle” code 702, and the “inner” code 704. The outer code is an irregular repetition code, and the middle and inner codes are both accumulators.

IRA codes may be implemented in a variety of channels, including memoryless channels, such as the BEC, BSC, and AWGN, as well as channels having non-binary input, non-symmetric and fading channels, and/or channels with memory.

A number of embodiments have been described. Nevertheless, it will be understood that various modifications may be made without departing from the spirit and scope of the invention. Accordingly, other embodiments are within the scope of the following claims.

The invention claimed is:

1. A method comprising:

receiving a collection of message bits having a first sequence in a source data stream;

generating a sequence of parity bits, wherein each parity bit “x_j” in the sequence is in accordance with the formula

$$x_j = x_{j-1} + \sum_{i=1}^{\lambda} v_{(j-1)\lambda+i},$$

where

“x_{j-1}” is the value of a parity bit “j-1,” and

$$v = \sum_{i=1}^a v_{(j-1)a+i}''$$

is the value of a sum of “a” randomly chosen irregular repeats of the message bits; and

making the sequence of parity bits available for transmission in a transmission data stream.

2. The method of claim 1, wherein the sequence of parity bits is generated is in accordance with “a” being constant.

3. The method of claim 1, wherein the sequence of parity bits is generated is in accordance with “a” varying for different parity bits.

4. The method of claim 1, wherein generating the sequence of parity bits comprises performing recursive modulo two addition operations on the random sequence of bits.

5. The method of claim 1, wherein generating the sequence of parity bits comprises:

generating a random sequence of bits that repeats each of the message bits one or more times with the repeats of the message bits being distributed in a random sequence, wherein different fractions of the message bits are each repeated a different number of times and the number of repeats for each message bit is irregular; and

XOR summing in linear sequential fashion a predecessor parity bit and “a” bits of the random sequence of bits.

6. The method of claim 5, wherein generating the random sequence of bits comprises coding the collection of message bits using a low-density generator matrix (LDGM) coder.

7. The method of claim 5, wherein generating the random sequence of bits comprises:

producing a block of data bits, wherein different message bits are each repeated a different number of times in a sequence that matches the first sequence; and

randomly permuting the different bits to generate the random sequence.

8. The method of claim 1, further comprising transmitting the sequence of parity bits.

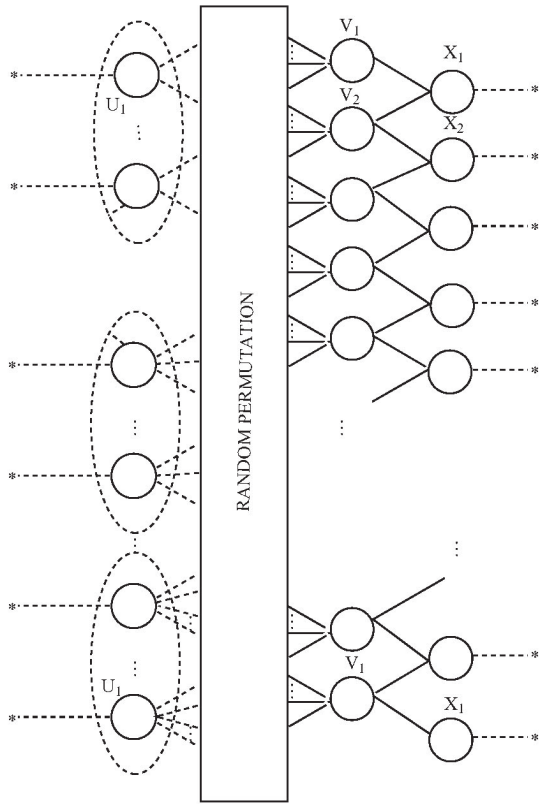
9. The method of claim 8, wherein transmitting the sequence of parity bits comprises transmitting the sequence of parity bits as part of a nonsystematic code.

10. The method of claim 8, wherein transmitting the sequence of parity bits comprises transmitting the sequence of parity bits as part of a systematic code.

11. A device comprising:

an encoder configured to receive a collection of message bits and encode the message bits to generate a collection of parity bits in accordance with the following Tanner graph:

9



12. The device of claim 11, wherein the encoder is configured to generate the collection of parity bits as if a number of inputs into nodes v_i was not constant.

13. The device of claim 11, wherein the encoder comprises: a low-density generator matrix (LDGM) coder configured to perform an irregular repeat on message bits having a first sequence in a source data stream to output a random sequence of repeats of the message bits; and an accumulator configured to XOR sum in linear sequential fashion a predecessor parity bit and "a" bits of the random sequence of repeats of the message bits.

14. The device of claim 12, wherein the accumulator comprises a recursive convolutional coder.

15. The device of claim 14, wherein the recursive convolutional coder comprises a truncated rate-1 recursive convolutional coder.

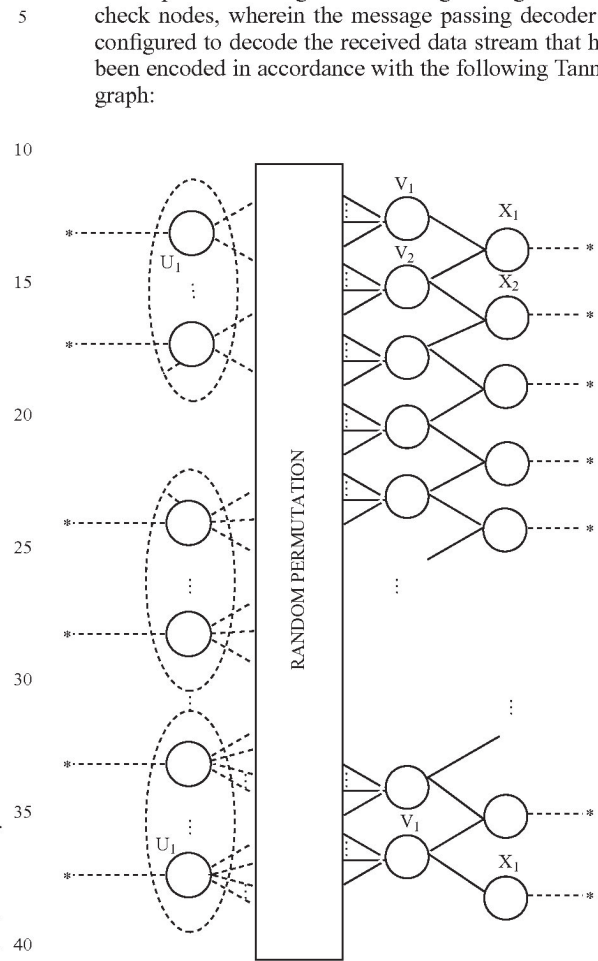
16. The device of claim 14, wherein the recursive convolutional coder has a transfer function of $1/(1+D)$.

17. The device of claim 12, further comprising a second accumulator configured to determine a second sequence of parity bits that defines a second condition that constrains the random sequence of repeats of the message bits.

18. A device comprising:
a message passing decoder configured to decode a received data stream that includes a collection of parity bits, the

10

message passing decoder comprising two or more check/variable nodes operating in parallel to receive messages from neighboring check/variable nodes and send updated messages to the neighboring variable/check nodes, wherein the message passing decoder is configured to decode the received data stream that has been encoded in accordance with the following Tanner graph:



19. The device of claim 18, wherein the message passing decoder is configured to decode the received data stream that includes the message bits.

20. The device of claim 18, wherein the message passing decoder is configured to decode the received data stream as if a number of inputs into nodes v_i was not constant.

21. The device of claim 18, wherein the message passing decoder is configured to decode in linear time at rates that approach a capacity of a channel.

22. The device of claim 18, wherein the message passing decoder comprises a belief propagation decoder.

23. The device of claim 18, wherein the message passing decoder is configured to decode the received data stream without the message bits.

* * * * *

UNITED STATES PATENT AND TRADEMARK OFFICE
CERTIFICATE OF CORRECTION

PATENT NO. : 7,421,032 B2
APPLICATION NO. : 11/542950
DATED : September 2, 2008
INVENTOR(S) : Hui Jin, Aamod Khandekar and Robert J. McEliece

Page 1 of 1

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

Title Page, item [73] (Assignee), line 1, please delete "Callifornia" and insert --California--, therefor.

Claim 11, Column 9, line 28, delete "V₁" and insert --V_r--, therefor.

Claim 11, Column 9, line 29, delete "U₁" and insert --U_k--, therefor.

Claim 11, Column 9, line 29, delete "X₁" and insert --X_r--, therefor.

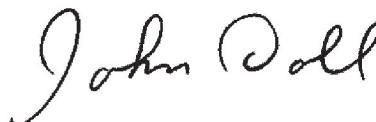
Claim 18, Column 10, line 35, delete "V₁" and insert --V_r--, therefor.

Claim 18, Column 10, line 36, delete "U₁" and insert --U_k--, therefor.

Claim 18, Column 10, line 37, delete "X₁" and insert --X_r--, therefor.

Signed and Sealed this

Seventeenth Day of February, 2009



JOHN DOLL
Acting Director of the United States Patent and Trademark Office

UNITED STATES PATENT AND TRADEMARK OFFICE
CERTIFICATE OF CORRECTION

PATENT NO. : 7,421,032 B2
 APPLICATION NO. : 11/542950
 DATED : September 2, 2008
 INVENTOR(S) : Hui Jin, Aamod Khandekar and Robert J. McEliece

Page 1 of 1

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

At column 4, line 14, please delete “ $x_j = x_{j-1} + \sum_{i=1}^{\lambda} v_{(j-1)\lambda+i}$ ” and insert

$$x_j = x_{j-1} + \sum_{i=1}^a v_{(j-1)a+i}$$

In claim 1, column 8, line 4, please delete “ $x_j = x_{j-1} + \sum_{i=1}^{\lambda} v_{(j-1)\lambda+i}$,” and insert

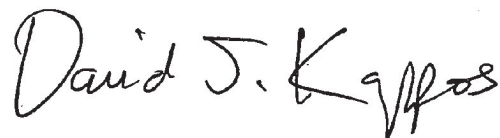
$$x_j = x_{j-1} + \sum_{i=1}^a v_{(j-1)a+i},$$

In claim 1, column 8, line 13, please delete “ $\sum_{i=1}^a v_{(j-1)a+1}$ ” and insert

$$\sum_{i=1}^a v_{(j-1)a+i}$$

Signed and Sealed this

Twenty-seventh Day of July, 2010



David J. Kappos
 Director of the United States Patent and Trademark Office

UNITED STATES PATENT AND TRADEMARK OFFICE
CERTIFICATE OF CORRECTION

PATENT NO. : 7,421,032 B2
APPLICATION NO. : 11/542950
DATED : September 2, 2008
INVENTOR(S) : Hui Jin, Aamod Khandekar and Robert J. McEliece

Page 1 of 1

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

On the Title Page

Item [63], delete:

“Continuation of application No. 09/861,102, filed on May 18, 2001, now Pat. No. 7,116,710, and a continuation-in-part of application No. 09/922,852, filed on Aug. 18, 2000, now Pat. No. 7,089,477.”

And insert:

-- Continuation of application No. 09/861,102, filed on May 18, 2001, now Pat. No. 7,116,710. --

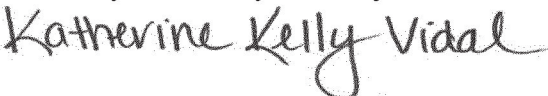
In the Specification

Column 1, Line 8, delete:

“This application is a continuation of U.S. application Ser. No. 09/861,102, filed May 18, 2001, now U.S. Pat. No. 7,116,710, which claims the priority of U.S. provisional application Ser. No. 60/205,095, filed May 18, 2000, and is a continuation-in-part of U.S. application Ser. No. 09/922,852, filed Aug. 18, 2000, now U.S. Pat. No. 7,089,477.”

And insert:

-- This application is a continuation of U.S. application Ser. No. 09/861,102, filed May 18, 2001, now U.S. Pat. No. 7,116,710, which claims the priority of U.S. provisional application Ser. No. 60/205,095, filed May 18, 2000. --

Signed and Sealed this
Thirty-first Day of May, 2022


Katherine Kelly Vidal
Director of the United States Patent and Trademark Office

(12) **INTER PARTES REVIEW CERTIFICATE** (1754th)

**United States Patent
Jin et al.**

(10) **Number:** US 7,421,032 K1
(45) **Certificate Issued:** May 11, 2020

(54) **SERIAL CONCATENATION OF
INTERLEAVED CONVOLUTIONAL CODES
FORMING TURBO-LIKE CODES**

(75) **Inventors: Hui Jin; Aamod Khandekar; Robert
J. McEliece**

(73) **Assignee: CALIFORNIA INSTITUTE OF
TECHNOLOGY**

Trial Numbers:

IPR2017-00700 filed Jan. 20, 2017
IPR2017-00701 filed Jan. 20, 2017
IPR2017-00728 filed Jan. 20, 2017

Inter Partes Review Certificate for:

Patent No.: **7,421,032**
Issued: **Sep. 2, 2008**
Appl. No.: **11/542,950**
Filed: **Oct. 3, 2006**

The results of IPR2017-00700; IPR2017-00701;
IPR2017-00728 are reflected in this inter partes review
certificate under 35 U.S.C. 318(b).

INTER PARTES REVIEW CERTIFICATE

U.S. Patent 7,421,032 K1

Trial No. IPR2017-00700

Certificate Issued May 11, 2020

1

2

AS A RESULT OF THE INTER PARTES
REVIEW PROCEEDING, IT HAS BEEN
DETERMINED THAT:

Claims **1, 4-16** and **18-23** are found patentable.

5

* * * * *

EXHIBIT C



US007916781B2

(12) **United States Patent**
Jin et al.

(10) **Patent No.:** **US 7,916,781 B2**
(45) **Date of Patent:** **Mar. 29, 2011**

(54) **SERIAL CONCATENATION OF INTERLEAVED CONVOLUTIONAL CODES FORMING TURBO-LIKE CODES**

(75) Inventors: **Hui Jin**, Glen Gardner, NJ (US); **Aamod Khandekar**, Pasadena, CA (US); **Robert J. McEliece**, Pasadena, CA (US)

(73) Assignee: **California Institute of Technology**, Pasadena, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 424 days.

(21) Appl. No.: **12/165,606**

(22) Filed: **Jun. 30, 2008**

Prior Publication Data

US 2008/0294964 A1 Nov. 27, 2008

Related U.S. Application Data

(63) Continuation of application No. 11/542,950, filed on Oct. 3, 2006, now Pat. No. 7,421,032, which is a continuation of application No. 09/861,102, filed on May 18, 2001, now Pat. No. 7,116,710, which is a continuation-in-part of application No. 09/922,852, filed on Aug. 18, 2000, now Pat. No. 7,089,477.

(60) Provisional application No. 60/205,095, filed on May 18, 2000.

(51) **Int. Cl.**
H04B 1/66 (2006.01)

(52) **U.S. Cl.** **375/240; 375/285; 375/296; 714/801; 714/804**

(58) **Field of Classification Search** **375/240, 375/240.24, 254, 285, 295, 296, 260; 714/755, 714/758, 800, 801, 804, 805**

See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,181,207 A *	1/1993	Chapman	714/755
5,392,299 A	2/1995	Rhines et al.	
5,530,707 A	6/1996	Lin	
5,751,739 A	5/1998	Seshadri et al.	
5,802,115 A	9/1998	Meyer	
5,881,093 A	3/1999	Wang et al.	
6,014,411 A	1/2000	Wang	
6,023,783 A	2/2000	Divsalar et al.	
6,031,874 A	2/2000	Chennakeshu et al.	
6,032,284 A	2/2000	Bliss	
6,044,116 A	3/2000	Wang	
6,094,739 A	7/2000	Miller et al.	
6,195,396 B1 *	2/2001	Fang et al.	375/261
6,396,423 B1	5/2002	Laumen et al.	
6,437,714 B1	8/2002	Kim et al.	
6,732,328 B1 *	5/2004	McEwen et al.	714/795
6,859,906 B2	2/2005	Hammons et al.	
7,089,477 B1	8/2006	Divsalar et al.	

(Continued)

OTHER PUBLICATIONS

Benedetto, S., et al., "A Soft-Input Soft-Output APP Module for Iterative Decoding of Concatenated Codes," *IEEE Communications Letters*, 1(1):22-24, Jan. 1997.

(Continued)

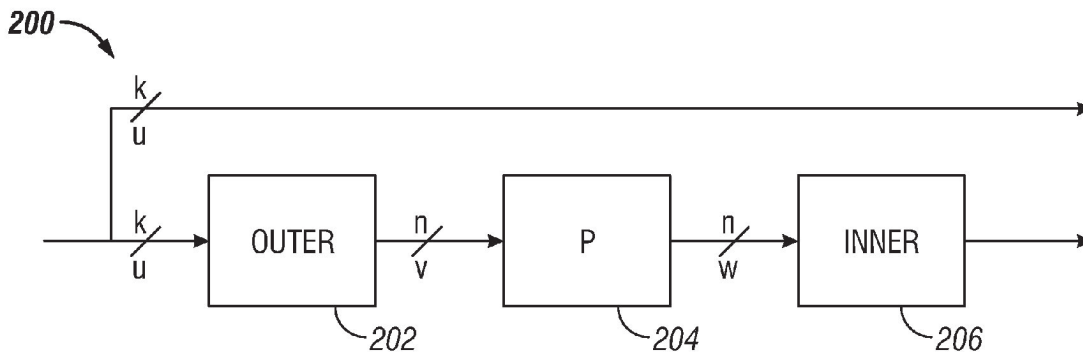
Primary Examiner — Dac V Ha

(74) Attorney, Agent, or Firm — Perkins Coie LLP

(57) **ABSTRACT**

A serial concatenated coder includes an outer coder and an inner coder. The outer coder irregularly repeats bits in a data block according to a degree profile and scrambles the repeated bits. The scrambled and repeated bits are input to an inner coder, which has a rate substantially close to one.

22 Claims, 5 Drawing Sheets



US 7,916,781 B2

Page 2

U.S. PATENT DOCUMENTS

2001/0025358 A1 9/2001 Eidson et al.

OTHER PUBLICATIONS

Benedetto, S., et al., "A Soft-Input Soft-Output Maximum A Posteriori (MAP) Module to Decode Parallel and Serial Concatenated Codes," *The Telecommunications and Data Acquisition Progress Report (TDA PR 42-127)*, pp. 1-20, Nov. 1996.

Benedetto, S., et al., "Bandwidth efficient parallel concatenated coding schemes," *Electronics Letters*, 31(24):2067-2069, Nov. 1995.

Benedetto, S., et al., "Design of Serially Concatenated Interleaved Codes," *ICC 97*, vol. 2, pp. 710-714, Jun. 1997.

Benedetto, S., et al., "Parallel Concatenated Trellis Coded Modulation," *ICC 96*, vol. 2, pp. 974-978, Jun. 1996.

Benedetto, S., et al., "Serial Concatenated Trellis Coded Modulation with Iterative Decoding," *Proceedings 1997 IEEE International Symposium on Information Theory (ISIT)*, Ulm, Germany, p. 8, Jun. 29-Jul. 4, 1997.

Benedetto, S., et al., "Serial Concatenation of Interleaved Codes: Performance Analysis, Design, and Iterative Decoding," *The Telecommunications and Data Acquisition Progress Report (TDAPR 42126)*, pp. 1-26, Aug. 1996.

Benedetto, S., et al., "Serial concatenation of interleaved codes: performance analysis, design, and iterative decoding," *Proceedings 1997 IEEE International Symposium on Information Theory (ISIT)*, Ulm, Germany, p. 106, Jun. 29-Jul. 4, 1997.

Benedetto, S., et al., "Soft-Output Decoding Algorithms in Iterative Decoding of Turbo Codes," *The Telecommunications and Data Acquisition Progress Report (TDA PR 42-124)*, pp. 63-87, Feb. 1996.

Berrou, C., et al., "Near Shannon Limit Error—Correcting Coding and Decoding: Turbo Codes," *ICC 93*, vol. 2, pp. 1064-1070, May 1993.

Digital Video Broadcasting (DVB)—User guidelines for the second generation system for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications (DVB-S2), ETSI TR 102 376 V1.1.1 Technical Report, pp. 1-104 (p. 64), Feb. 2005.

Divsalar, D., et al., "Coding Theorems for 'Turbo-Like' Codes," *Proceedings of the 36th Annual Allerton Conference on Communication, Control, and Computing*, Monticello, Illinois, pp. 201-210, Sep. 1998.

Divsalar, D., et al., "Effective free distance of turbo codes," *Electronics Letters*, 32(5):445-446, Feb. 1996.

Divsalar, D., et al., "Hybrid Concatenated Codes and Iterative Decoding," *Proceedings 1997 IEEE International Symposium on Information Theory (ISIT)*, Ulm, Germany, p. 10, Jun. 29-Jul. 4, 1997.

Divsalar, D., et al., "Low-Rate Turbo Codes for Deep-Space Communications," *Proceedings 1995 IEEE International Symposium on Information Theory (ISIT)*, Whistler, BC, Canada, p. 35, Sep. 1995.

Divsalar, D., et al., "Multiple Turbo Codes for Deep-Space Communications," *The Telecommunications and Data Acquisition Progress Report (TDA PR 42-121)*, pp. 66-77, May 1995.

Divsalar, D., et al., "Multiple Turbo Codes," *MILCOM '95*, vol. 1, pp. 279-285, Nov. 1995.

Divsalar, D., et al., "On the Design of Turbo Codes," *The Telecommunications and Data Acquisition Progress Report (TDA PR 42-123)*, pp. 99-121, Nov. 1995.

Divsalar, D., et al., "Serial Turbo Trellis Coded Modulation with Rate-1 Inner Code," *Proceedings 2000 IEEE International Symposium on Information Theory (ISIT)*, Sorrento, Italy, pp. 194, Jun. 2000.

Divsalar, D., et al., "Turbo Codes for PCS Applications," *IEEE ICC '95*, Seattle, WA, USA, vol. 1, pp. 54-59, Jun. 1995.

Jin, H., et al., "Irregular Repeat—Accumulate Codes," *2nd International Symposium on Turbo Codes*, Brest, France, 25 pages, Sep. 2000.

Jin, H., et al., "Irregular Repeat—Accumulate Codes," *2nd International Symposium on Turbo Codes & Related Topics*, Brest, France, p. 1-8, Sep. 2000.

Richardson, T.J., et al., "Design of Capacity-Approaching Irregular Low-Density Parity-Check Codes," *IEEE Transactions on Information Theory*, 47(2):619-637, Feb. 2001.

Richardson, T.J., et al., "Efficient Encoding of Low-Density Parity-Check Codes," *IEEE Transactions on Information Theory*, 47(2):638-656, Feb. 2001.

Wiberg, N., et al., "Codes and Iterative Decoding on General Graphs," *Proceedings 1995 IEEE International Symposium on Information Theory (ISIT)*, Whistler, BC, Canada, p. 468, Sep. 1995.

Aji, S.M., et al., "The Generalized Distributive Law," *IEEE Transactions on Information Theory*, 46(2):325-343, Mar. 2000.

Tanner, R.M., "A Recursive Approach to Low Complexity Codes," *IEEE Transactions on Information Theory*, 27(5):533-547, Sep. 1981.

* cited by examiner

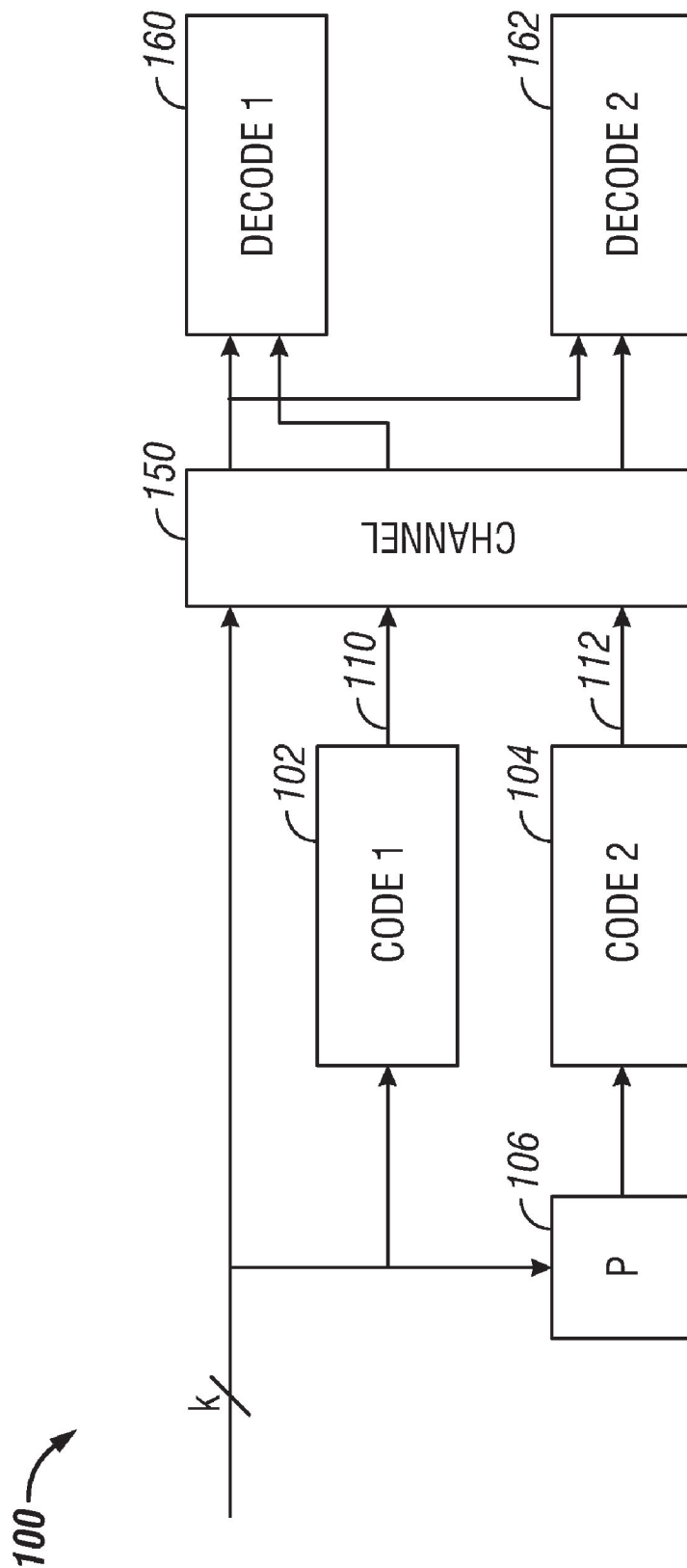


FIG. 1
(Prior Art)

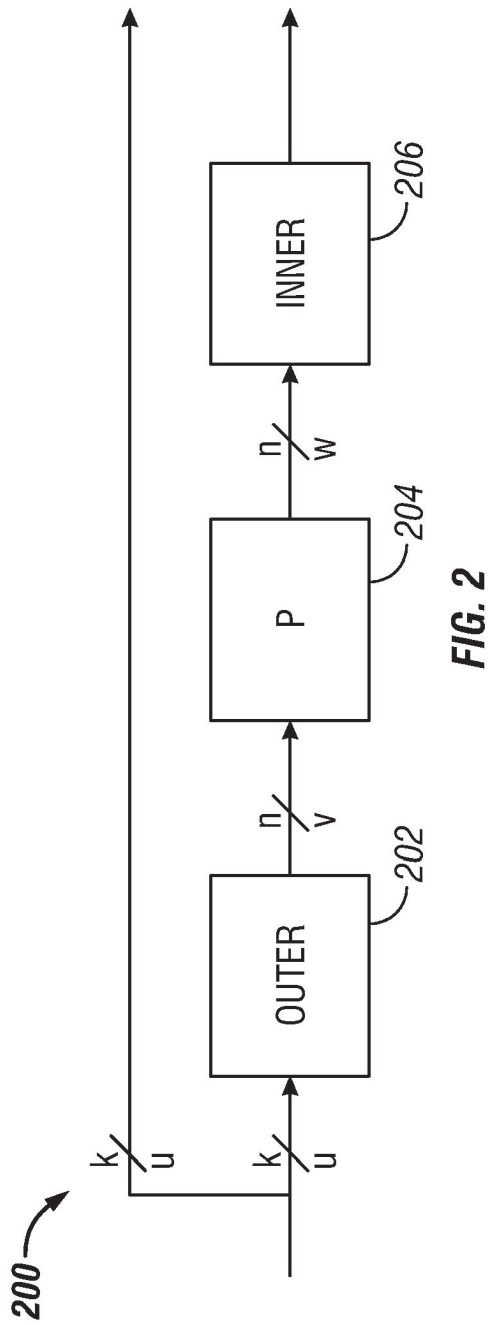


FIG. 2

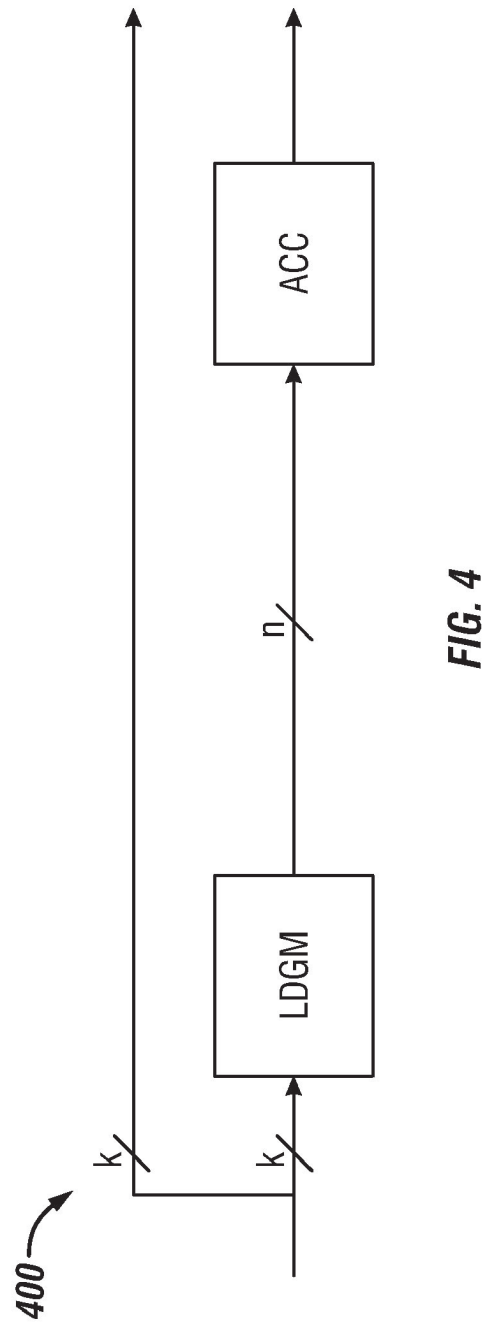


FIG. 4

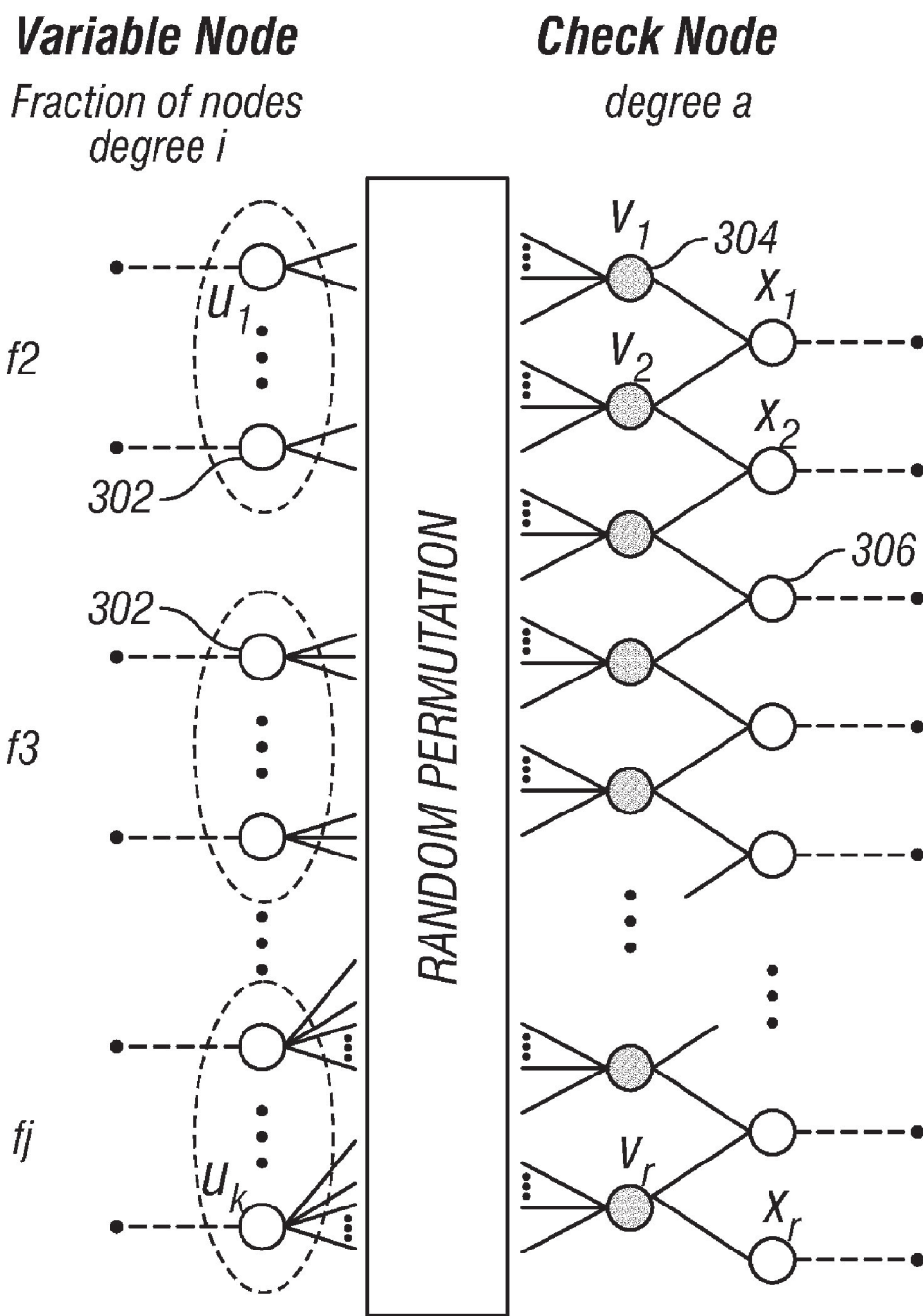


FIG. 3

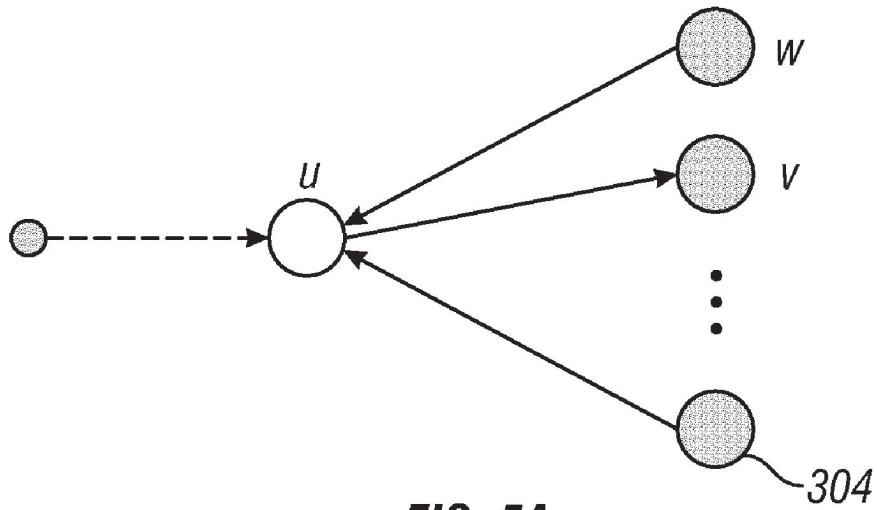


FIG. 5A

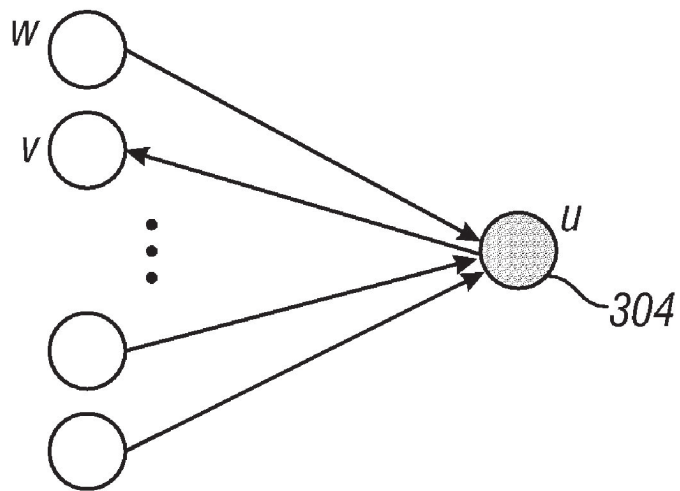


FIG. 5B

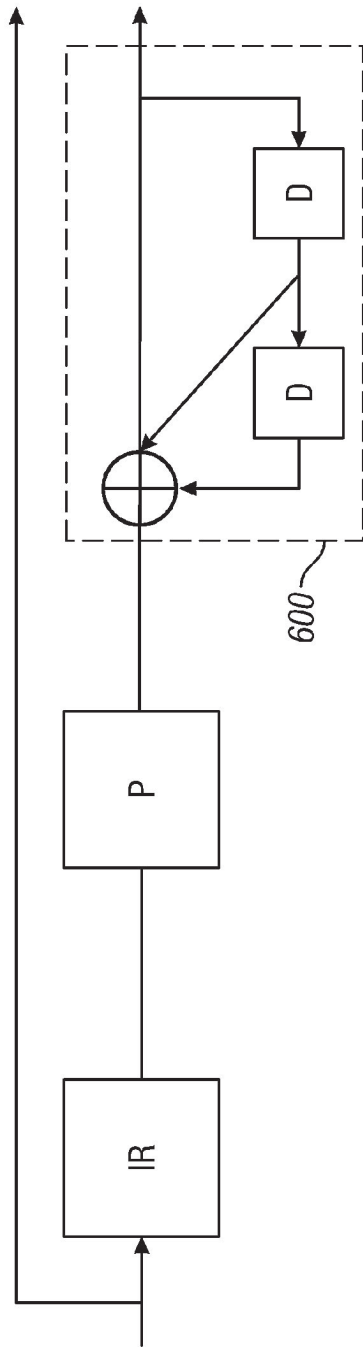


FIG. 6

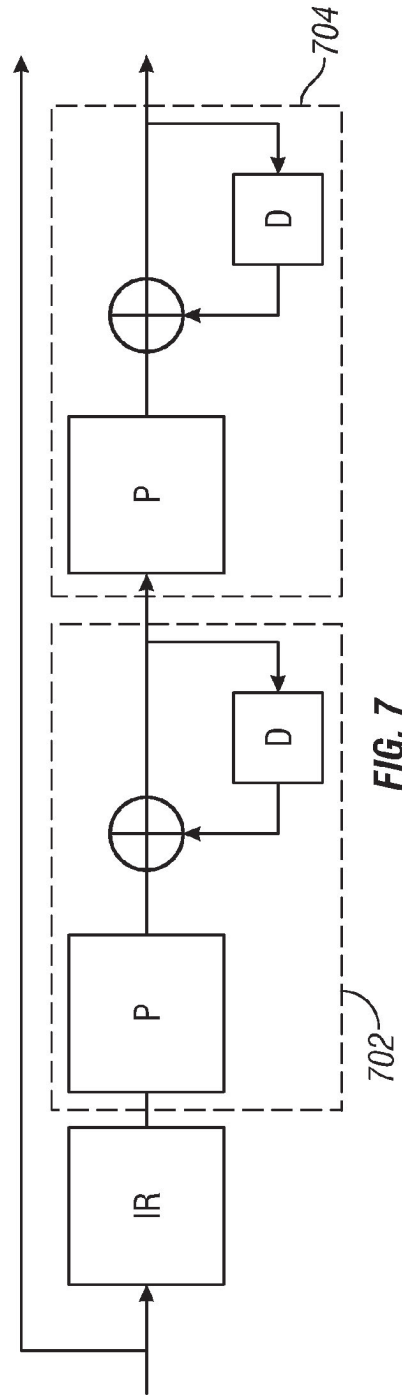


FIG. 7

US 7,916,781 B2

1

SERIAL CONCATENATION OF INTERLEAVED CONVOLUTIONAL CODES FORMING TURBO-LIKE CODES

CROSS-REFERENCE TO RELATED APPLICATIONS

This application is a continuation of U.S. application Ser. No. 11/542,950, filed Oct. 3, 2006 now U.S. Pat. No. 7,421,032, which is a continuation of U.S. application Ser. No. 09/861,102, filed May 18, 2001, now U.S. Pat. No. 7,116,710, which claims the priority of U.S. Provisional Application Ser. No. 60/205,095, filed May 18, 2000, and is a continuation-in-part of U.S. application Ser. No. 09/922,852, filed Aug. 18, 2000, now U.S. Pat. No. 7,089,477. The disclosure of the prior applications are considered part of (and are incorporated by reference in) the disclosure of this application.

GOVERNMENT LICENSE RIGHTS

The U.S. Government has a paid-up license in this invention and the right in limited circumstances to require the patent owner to license others on reasonable terms as provided for by the terms of Grant No. CCR-9804793 awarded by the National Science Foundation.

BACKGROUND

Properties of a channel affect the amount of data that can be handled by the channel. The so-called "Shannon limit" defines the theoretical limit of the amount of data that a channel can carry.

Different techniques have been used to increase the data rate that can be handled by a channel. "Near Shannon Limit Error-Correcting Coding and Decoding: Turbo Codes," by Berrou et al. ICC, pp 1064-1070, (1993), described a new "turbo code" technique that has revolutionized the field of error correcting codes. Turbo codes have sufficient randomness to allow reliable communication over the channel at a high data rate near capacity. However, they still retain sufficient structure to allow practical encoding and decoding algorithms. Still, the technique for encoding and decoding turbo codes can be relatively complex.

A standard turbo coder **100** is shown in FIG. **1**. A block of k information bits is input directly to a first coder **102**. A k bit interleaver **106** also receives the k bits and interleaves them prior to applying them to a second coder **104**. The second coder produces an output that has more bits than its input, that is, it is a coder with rate that is less than 1. The coders **102**, **104** are typically recursive convolutional coders.

Three different items are sent over the channel **150**: the original k bits, first encoded bits **110**, and second encoded bits **112**. At the decoding end, two decoders are used: a first constituent decoder **160** and a second constituent decoder **162**. Each receives both the original k bits, and one of the encoded portions **110**, **112**. Each decoder sends likelihood estimates of the decoded bits to the other decoders. The estimates are used to decode the uncoded information bits as corrupted by the noisy channel.

SUMMARY

A coding system according to an embodiment is configured to receive a portion of a signal to be encoded, for example, a data block including a fixed number of bits. The coding system includes an outer coder, which repeats and scrambles bits in the data block. The data block is apportioned

2

into two or more sub-blocks, and bits in different sub-blocks are repeated a different number of times according to a selected degree profile. The outer coder may include a repeater with a variable rate and an interleaver. Alternatively, the outer coder may be a low-density generator matrix (LDGM) coder.

The repeated and scrambled bits are input to an inner coder that has a rate substantially close to one. The inner coder may include one or more accumulators that perform recursive modulo two addition operations on the input bit stream.

The encoded data output from the inner coder may be transmitted on a channel and decoded in linear time at a destination using iterative decoding techniques. The decoding techniques may be based on a Tanner graph representation of the code.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** is a schematic diagram of a prior "turbo code" system.

FIG. **2** is a schematic diagram of a coder according to an embodiment.

FIG. **3** is a Tanner graph for an irregular repeat and accumulate (IRA) coder.

FIG. **4** is a schematic diagram of an IRA coder according to an embodiment.

FIG. **5A** illustrates a message from a variable node to a check node on the Tanner graph of FIG. **3**.

FIG. **5B** illustrates a message from a check node to a variable node on the Tanner graph of FIG. **3**.

FIG. **6** is a schematic diagram of a coder according to an alternate embodiment.

FIG. **7** is a schematic diagram of a coder according to another alternate embodiment.

DETAILED DESCRIPTION

FIG. **2** illustrates a coder **200** according to an embodiment. The coder **200** may include an outer coder **202**, an interleaver **204**, and inner coder **206**. The coder may be used to format blocks of data for transmission, introducing redundancy into the stream of data to protect the data from loss due to transmission errors. The encoded data may then be decoded at a destination in linear time at rates that may approach the channel capacity.

The outer coder **202** receives the uncoded data. The data may be partitioned into blocks of fixed size, say k bits. The outer coder may be an (n,k) binary linear block coder, where $n > k$. The coder accepts as input a block u of k data bits and produces an output block v of n data bits. The mathematical relationship between u and v is $v = T_o u$, where T_o is an $n \times k$ matrix, and the rate of the coder is k/n .

The rate of the coder may be irregular, that is, the value of T_o is not constant, and may differ for sub-blocks of bits in the data block. In an embodiment, the outer coder **202** is a repeater that repeats the k bits in a block a number of times q to produce a block with n bits, where $n = qk$. Since the repeater has an irregular output, different bits in the block may be repeated a different number of times. For example, a fraction of the bits in the block may be repeated two times, a fraction of bits may be repeated three times, and the remainder of bits may be repeated four times. These fractions define a degree sequence, or degree profile, of the code.

The inner coder **206** may be a linear rate-1 coder, which means that the n -bit output block x can be written as $x = T_I w$, where T_I is a nonsingular $n \times n$ matrix. The inner coder **210** can

3

have a rate that is close to 1, e.g., within 50%, more preferably 10% and perhaps even more preferably within 1% of 1.

In an embodiment, the inner coder **206** is an accumulator, which produces outputs that are the modulo two (mod-2) partial sums of its inputs. The accumulator may be a truncated rate-1 recursive convolutional coder with the transfer function $1/(1+D)$. Such an accumulator may be considered a block coder whose input block $[x_1, \dots, x_n]$ and output block $[y_1, \dots, y_n]$ are related by the formula

$$\begin{aligned} y_1 &= x_1 \\ y_2 &= x_1 \oplus x_2 \\ y_3 &= x_1 \oplus x_2 \oplus x_3 \\ &\vdots \\ &\vdots \\ &\vdots \\ y_n &= x_1 \oplus x_2 \oplus x_3 \oplus \dots \oplus x_n \end{aligned}$$

where “ \oplus ” denotes mod-2, or exclusive-OR (XOR), addition. An advantage of this system is that only mod-2 addition is necessary for the accumulator. The accumulator may be embodied using only XOR gates, which may simplify the design.

The bits output from the outer coder **202** are scrambled before they are input to the inner coder **206**. This scrambling may be performed by the interleaver **204**, which performs a pseudo-random permutation of an input block v , yielding an output block w having the same length as v .

The serial concatenation of the interleaved irregular repeat code and the accumulate code produces an irregular repeat and accumulate (IRA) code. An IRA code is a linear code, and as such, may be represented as a set of parity checks. The set of parity checks may be represented in a bipartite graph, called the Tanner graph, of the code. FIG. 3 shows a Tanner graph **300** of an IRA code with parameters $(f_1, \dots, f_j; a)$, where $f_i \geq 0$, $\sum_i f_i = 1$ and “ a ” is a positive integer. The Tanner graph includes two kinds of nodes: variable nodes (open circles) and check nodes (filled circles). There are k variable nodes **302** on the left, called information nodes. There are r variable nodes **306** on the right, called parity nodes. There are $r = (k \sum_i f_i) / a$ check nodes **304** connected between the information nodes and the parity nodes. Each information node **302** is connected to a number of check nodes **304**. The fraction of information nodes connected to exactly i check nodes is f_i . For example, in the Tanner graph **300**, each of the f_2 information nodes are connected to two check nodes, corresponding to a repeat of $q=2$, and each of the f_3 information nodes are connected to three check nodes, corresponding to $q=3$.

Each check node **304** is connected to exactly “ a ” information nodes **302**. In FIG. 3, $a=3$. These connections can be made in many ways, as indicated by the arbitrary permutation of the ra edges joining information nodes **302** and check nodes **304** in permutation block **310**. These connections correspond to the scrambling performed by the interleaver **204**.

In an alternate embodiment, the outer coder **202** may be a low-density generator matrix (LDGM) coder that performs an irregular repeat of the k bits in the block, as shown in FIG. 4. As the name implies, an LDGM code has a sparse (low-density) generator matrix. The IRA code produced by the coder **400** is a serial concatenation of the LDGM code and the

4

accumulator code. The interleaver **204** in FIG. 2 may be excluded due to the randomness already present in the structure of the LDGM code.

If the permutation performed in permutation block **310** is fixed, the Tanner graph represents a binary linear block code with k information bits (u_1, \dots, u_k) and r parity bits (x_1, \dots, x_r) , as follows. Each of the information bits is associated with one of the information nodes **302**, and each of the parity bits is associated with one of the parity nodes **306**. The value of a parity bit is determined uniquely by the condition that the mod-2 sum of the values of the variable nodes connected to each of the check nodes **304** is zero. To see this, set $x_0=0$. Then if the values of the bits on the ra edges coming out the permutation box are

$$x_j = x_{j-1} + \sum_{i=1}^a v_{(j-1)a+i}$$

(v_1, \dots, v_{ra}) , then we have the recursive formula for $j=1, 2, \dots, r$. This is in effect the encoding algorithm.

Two types of IRA codes are represented in FIG. 3, a non-systematic version and a systematic version. The nonsystematic version is an (r,k) code, in which the codeword corresponding to the information bits (u_1, \dots, u_k) is (x_1, \dots, x_r) . The systematic version is a $(k+r, k)$ code, in which the codeword is $(u_1, \dots, u_k; x_1, \dots, x_r)$.

The rate of the nonsystematic code is

$$R_{n,sys} = \frac{a}{\sum_i f_i}$$

The rate of the systematic code is

$$R_{sys} = \frac{a}{a + \sum_i f_i}$$

For example, regular repeat and accumulate (RA) codes can be considered nonsystematic IRA codes with $a=1$ and exactly one f_i equal to 1, say $f_q=1$, and the rest zero, in which case $R_{n,sys}$ simplifies to $R=1/q$.

The IRA code may be represented using an alternate notation. Let λ_i be the fraction of edges between the information nodes **302** and the check nodes **304** that are adjacent to an information node of degree i , and let ρ_i be the fraction of such edges that are adjacent to a check node of degree $i+2$ (i.e., one that is adjacent to i information nodes). These edge fractions may be used to represent the IRA code rather than the corresponding node fractions. Define $\lambda(x) = \sum_i \lambda_i x^{i-1}$ and $\rho(x) = \sum_i \rho_i x^{i-1}$ to be

$$f_i = \frac{\lambda_i / i}{\sum_j \lambda_j / j}$$

the generating functions of these sequences. The pair (λ, ρ) is called a degree distribution. For $L(x) = \sum_i f_i x^i$,

US 7,916,781 B2

5

The rate of the systematic IRA code given by the

$$L(x) = \int_0^x \lambda(t) dt / \int_0^1 \lambda(t) dt$$

$$\text{Rate} = \left(1 + \frac{\sum_j \rho_j / j}{\sum_j \lambda_j / j} \right)^{-1}$$

degree distribution is given by

“Belief propagation” on the Tanner Graph realization may be used to decode IRA codes. Roughly speaking, the belief propagation decoding technique allows the messages passed on an edge to represent posterior densities on the bit associated with the variable node. A probability density on a bit is a pair of non-negative real numbers $p(0)$, $p(1)$ satisfying $p(0)+p(1)=1$, where $p(0)$ denotes the probability of the bit being 0, $p(1)$ the probability of it being 1. Such a pair can be represented by its log likelihood ratio, $m=\log(p(0)/p(1))$. The outgoing message from a variable node u to a check node v represents information about u , and a message from a check node u to a variable node v represents information about u , as shown in FIGS. 5A and 5B, respectively.

The outgoing message from a node u to a node v depends on the incoming messages from all neighbors w of u except v . If u is a variable message node, this outgoing message is

$$m(u \rightarrow v) = \sum_{w \neq v} m(w \rightarrow u) + m_0(u)$$

where $m_0(u)$ is the log-likelihood message associated with u . If u is a check node, the corresponding formula is

$$\tanh \frac{m(u \rightarrow v)}{2} = \prod_{w \neq v} \tanh \frac{m(w \rightarrow u)}{2}$$

Before decoding, the messages $m(w \rightarrow u)$ and $m(u \rightarrow v)$ are initialized to be zero, and $m_0(u)$ is initialized to be the log-likelihood ratio based on the channel received information. If the channel is memoryless, i.e., each channel output only relies on its input, and y is the output of the channel code bit u , then $m_0(u)=\log(p(u=0|y)/p(u=1|y))$. After this initialization, the decoding process may run in a fully parallel and local manner. In each iteration, every variable/check node receives messages from its neighbors, and sends back updated messages. Decoding is terminated after a fixed number of iterations or detecting that all the constraints are satisfied. Upon termination, the decoder outputs a decoded sequence based on the messages

$$m(u) = \sum w_m(w \rightarrow u).$$

Thus, on various channels, iterative decoding only differs in the initial messages $m_0(u)$. For example, consider three memoryless channel models: a binary erasure channel (BEC); a binary symmetric channel (BSC); and an additive white Gaussian noise (AGWN) channel.

In the BEC, there are two inputs and three outputs. When 0 is transmitted, the receiver can receive either 0 or an erasure E.

6

An erasure E output means that the receiver does not know how to demodulate the output. Similarly, when 1 is transmitted, the receiver can receive either 1 or E. Thus, for the BEC, $y \in \{0, E, 1\}$, and

$$m_0(u) = \begin{cases} +\infty & \text{if } y = 0 \\ 0 & \text{if } y = E \\ -\infty & \text{if } y = 1 \end{cases}$$

In the BSC, there are two possible inputs (0,1) and two possible outputs (0, 1). The BSC is characterized by a set of conditional probabilities relating all possible outputs to possible inputs. Thus, for the BSC $y \in \{0, 1\}$,

$$m_0(u) = \begin{cases} \log \frac{1-p}{p} & \text{if } y = 0 \\ -\log \frac{1-p}{p} & \text{if } y = 1 \end{cases}$$

In the AWGN, the discrete-time input symbols X take their values in a finite alphabet while channel output symbols Y can take any values along the real line. There is assumed to be no distortion or other effects other than the addition of white Gaussian noise. In an AWGN with a Binary Phase Shift Keying (BPSK) signaling which maps 0 to the symbol with amplitude $\sqrt{E_s}$ and 1 to the symbol with amplitude $-\sqrt{E_s}$, output $y \in \mathbb{R}$, then

$$m_0(u) = 4y\sqrt{E_s}N_0$$

where $N_0/2$ is the noise power spectral density.

The selection of a degree profile for use in a particular transmission channel is a design parameter, which may be affected by various attributes of the channel. The criteria for selecting a particular degree profile may include, for example, the type of channel and the data rate on the channel. For example, Table 1 shows degree profiles that have been found to produce good results for an AWGN channel model.

TABLE 1

a	2	3	4
λ_2	0.139025	0.078194	0.054485
λ_3	0.2221555	0.128085	0.104315
λ_5		0.160813	
λ_6	0.638820	0.036178	0.126755
λ_{10}			0.229816
λ_{11}			0.016484
λ_{12}		0.108828	
λ_{13}		0.487902	
λ_{14}			
λ_{16}			
λ_{27}			0.450302
λ_{28}			0.017842
Rate	0.333364	0.333223	0.333218
σ_A	1.1840	1.2415	1.2615
σ^*	1.1981	1.2607	1.2780
(Eb/N0) * (dB)	0.190	-0.250	-0.371
S.L. (dB)	-0.4953	-0.4958	-0.4958

Table 1 shows degree profiles yielding codes of rate approximately 1/3 for the AWGN channel and with $a=2, 3, 4$. For each sequence, the Gaussian approximation noise threshold, the actual sum-product decoding threshold and the corresponding energy per bit (E_b)-noise power (N_0) ratio in dB are given. Also listed is the Shannon limit (S.L.).

US 7,916,781 B2

7

As the parameter “a” is increased, the performance improves. For example, for a=4, the best code found has an iterative decoding threshold of $E_b/N_0 = -0.371$ dB, which is only 0.12 dB above the Shannon limit.

The accumulator component of the coder may be replaced by a “double accumulator” **600** as shown in FIG. 6. The double accumulator can be viewed as a truncated rate 1 convolutional coder with transfer function $1/(1+D+D^2)$.

Alternatively, a pair of accumulators may be added, as shown in FIG. 7. There are three component codes: the “outer” code **700**, the “middle” code **702**, and the “inner” code **704**. The outer code is an irregular repetition code, and the middle and inner codes are both accumulators.

IRA codes may be implemented in a variety of channels, including memoryless channels, such as the BEC, BSC, and AWGN, as well as channels having non-binary input, non-symmetric and fading channels, and/or channels with memory.

A number of embodiments have been described. Nevertheless, it will be understood that various modifications may be made without departing from the spirit and scope of the invention. Accordingly, other embodiments are within the scope of the following claims.

What is claimed is:

1. A method of encoding a signal, comprising:
 - receiving a block of data in the signal to be encoded, the block of data including information bits;
 - performing a first encoding operation on at least some of the information bits, the first encoding operation being a linear transform operation that generates L transformed bits; and
 - performing a second encoding operation using the L transformed bits as an input, the second encoding operation including an accumulation operation in which the L transformed bits generated by the first encoding operation are accumulated, said second encoding operation producing at least a portion of a codeword, wherein L is two or more.
2. The method of claim 1, further comprising:
 - outputting the codeword, wherein the codeword comprises parity bits.
3. The method of claim 2, wherein outputting the codeword comprises:
 - outputting the parity bits; and
 - outputting at least some of the information bits.
4. The method of claim 3, wherein outputting the codeword comprises:
 - outputting the parity bits following the information bits.
5. The method of claim 2, wherein performing the first encoding operation comprises transforming the at least some of the information bits via a low density generator matrix transformation.
6. The method of claim 5, wherein generating each of the L transformed bits comprises mod-2 or exclusive-OR summing of bits in a subset of the information bits.
7. The method of claim 6, wherein each of the subsets of the information bits includes a same number of the information bits.
8. The method of claim 6, wherein at least two of the information bits appear in three subsets of the information bits.
9. The method of claim 6, wherein the information bits appear in a variable number of subsets.
10. The method of claim 2, wherein performing the second encoding operation comprises using a first of the parity bits in the accumulation operation to produce a second of the parity bits.

8

11. The method of claim 10, wherein outputting the codeword comprises outputting the second of the parity bits immediately following the first of the parity bits.

12. The method of claim 2, wherein performing the second encoding operation comprises performing one of a mod-2 addition and an exclusive-OR operation.

13. A method of encoding a signal, comprising:

- receiving a block of data in the signal to be encoded, the block of data including information bits; and
- performing an encoding operation using the information bits as an input, the encoding operation including an accumulation of mod-2 or exclusive-OR sums of bits in subsets of the information bits, the encoding operation generating at least a portion of a codeword, wherein the information bits appear in a variable number of subsets.

14. The method of claim 13, further comprising:

- outputting the codeword, wherein the codeword comprises parity bits.

15. The method of claim 14, wherein outputting the codeword comprises:

outputting the parity bits; and
outputting at least some of the information bits.

16. The method of claim 15, wherein the parity bits follow the information bits in the codeword.

17. The method of claim 13, wherein each of the subsets of the information bits includes a constant number of the information bits.

18. The method of claim 13, wherein performing the encoding operation further comprises:

- performing one of the mod-2 addition and the exclusive-OR summing of the bits in the subsets.

19. A method of encoding a signal, comprising:

- receiving a block of data in the signal to be encoded, the block of data including information bits; and
- performing an encoding operation using the information bits as an input, the encoding operation including an accumulation of mod-2 or exclusive-OR sums of bits in subsets of the information bits, the encoding operation generating at least a portion of a codeword, wherein at least two of the information bits appear in three subsets of the information bits.

20. A method of encoding a signal, comprising:

- receiving a block of data in the signal to be encoded, the block of data including information bits; and
- performing an encoding operation using the information bits as an input, the encoding operation including an accumulation of mod-2 or exclusive-OR sums of bits in subsets of the information bits, the encoding operation generating at least a portion of a codeword, wherein performing the encoding operation comprises:
 - mod-2 or exclusive-OR adding a first subset of information bits in the collection to yield a first sum;
 - mod-2 or exclusive-OR adding a second subset of information bits in the collection and the first sum to yield a second sum.

21. A method comprising:

- receiving a collection of information bits;
- mod-2 or exclusive-OR adding a first subset of information bits in the collection to yield a first parity bit;
- mod-2 or exclusive-OR adding a second subset of information bits in the collection and the first parity bit to yield a second parity bit; and
- outputting a codeword that includes the first parity bit and the second parity bit.

US 7,916,781 B2

9

22. The method of claim 21, wherein:
the method further comprises mod-2 or exclusive-OR adding additional subsets of information bits in the collection and parity bits to yield additional parity bits; and

10

the information bits in the collection appear in a variable number of subsets.

* * * * *

UNITED STATES PATENT AND TRADEMARK OFFICE
CERTIFICATE OF CORRECTION

PATENT NO. : 7,916,781 B2
APPLICATION NO. : 12/165606
DATED : March 29, 2011
INVENTOR(S) : Hui Jin, Aamod Khandekar and Robert J. McEliece

Page 1 of 1

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

On the Title Page

Item [63], the sentence reading:

“Continuation of application No. 11/542,950, filed on Oct. 3, 2006, now Pat. No. 7,421,032, which is a continuation of application No. 09/861,102, filed on May 18, 2001, now Pat. No. 7,116,710, which is a continuation-in-part of application No. 09/922,852, filed on Aug. 18, 2000, now Pat. No. 7,089,477.”

Should read:

-- Continuation of application No. 11/542,950, filed on Oct. 3, 2006, now Pat. No. 7,421,032, which is a continuation of application No. 09/861,102, filed on May 18, 2001, now Pat. No. 7,116,710. --

In the Specification

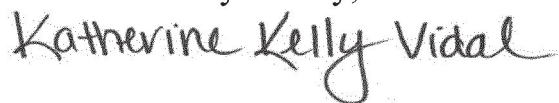
Column 1, Line 8, the sentence reading:

“This application is a continuation of U.S. application Ser. No. 11/542,950, filed Oct. 3, 2006 now U.S. Pat. No. 7,421,032, which is a continuation of U.S. application Ser. No. 09/861,102, filed May 18, 2001, now U.S. Pat. No. 7,116,710, which claims the priority of U.S. Provisional Application Ser. No. 60/205,095, filed May 18, 2000, and is a continuation-in-part of U.S. application Ser. No. 09/922,852, filed Aug. 18, 2000, now U.S. Pat. No. 7,089,477.”

Should read:

-- This application is a continuation of U.S. application Ser. No. 11/542,950, filed Oct. 3, 2006, now U.S. Pat. No. 7,421,032, which is a continuation of U.S. application Ser. No. 09/861,102, filed May 18, 2001, now U.S. Pat. No. 7,116,710, which claims the priority of U.S. Provisional Application Ser. No. 60/205,095, filed May 18, 2000. --

Signed and Sealed this
Fifth Day of July, 2022



Katherine Kelly Vidal
Director of the United States Patent and Trademark Office

(12) **INTER PARTES REVIEW CERTIFICATE** (668th)

**United States Patent
Jin et al.**

(10) **Number:** US 7,916,781 K1
(45) **Certificate Issued:** Feb. 14, 2018

(54) **SERIAL CONCATENATION OF
INTERLEAVED CONVOLUTIONAL CODES
FORMING TURBO-LIKE CODES**

(75) **Inventors: Hui Jin; Aamod Khandekar; Robert
J. McEliece**

(73) **Assignee: CALIFORNIA INSTITUTE OF
TECHNOLOGY**

Trial Number:

IPR2015-00059 filed Oct. 14, 2014

Inter Partes Review Certificate for:

Patent No.: **7,916,781**
Issued: **Mar. 29, 2011**
Appl. No.: **12/165,606**
Filed: **Jun. 30, 2008**

The results of IPR2015-00059 are reflected in this inter partes review certificate under 35 U.S.C. 318(b).

INTER PARTES REVIEW CERTIFICATE

U.S. Patent 7,916,781 K1

Trial No. IPR2015-00059

Certificate Issued Feb. 14, 2018

1

2

AS A RESULT OF THE INTER PARTES
REVIEW PROCEEDING, IT HAS BEEN
DETERMINED THAT:

Claims 1 and 2 are cancelled.

5

* * * * *

(12) **INTER PARTES REVIEW CERTIFICATE** (1736th)

**United States Patent
Jin et al.**

(10) **Number:** US 7,916,781 K2
(45) **Certificate Issued:** May 1, 2020

(54) **SERIAL CONCATENATION OF
INTERLEAVED CONVOLUTIONAL CODES
FORMING TURBO-LIKE CODES**

(75) **Inventors: Hui Jin; Aamod Khandekar; Robert
J. McEliece**

(73) **Assignee: CALIFORNIA INSTITUTE OF
TECHNOLOGY**

Trial Numbers:

IPR2017-00297 filed Dec. 12, 2016
IPR2017-00423 filed Dec. 12, 2016

Inter Partes Review Certificate for:

Patent No.: **7,916,781**
Issued: **Mar. 29, 2011**
Appl. No.: **12/165,606**
Filed: **Jun. 30, 2008**

The results of IPR2017-00297; IPR2017-00423 are reflected in this inter partes review certificate under 35 U.S.C. 318(b).

INTER PARTES REVIEW CERTIFICATE

U.S. Patent 7,916,781 K2

Trial No. IPR2017-00297

Certificate Issued May 1, 2020

1

2

AS A RESULT OF THE INTER PARTES
REVIEW PROCEEDING, IT HAS BEEN
DETERMINED THAT:

Claims **13-16, 18, 22** are found patentable.

5

Claims **19-21** are cancelled.

* * * * *

EXHIBIT D



US008284833B2

(12) **United States Patent**
Jin et al.

(10) **Patent No.:** **US 8,284,833 B2**
(45) **Date of Patent:** **Oct. 9, 2012**

(54) **SERIAL CONCATENATION OF INTERLEAVED CONVOLUTIONAL CODES FORMING TURBO-LIKE CODES**

(58) **Field of Classification Search** 375/240, 375/240.24, 254, 285, 295, 296, 260; 714/755, 714/758, 800, 801, 804, 805
See application file for complete search history.

(75) Inventors: **Hui Jin**, Glen Gardner, NJ (US); **Aamod Khandekar**, Pasadena, CA (US); **Robert J. McEliece**, Pasadena, CA (US)

(56) **References Cited**

(73) Assignee: **California Institute of Technology**, Pasadena, CA (US)

U.S. PATENT DOCUMENTS

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

5,181,207	A	1/1993	Chapman	
5,392,299	A	2/1995	Rhines et al.	
5,530,707	A	6/1996	Lin	
5,751,739	A	5/1998	Seshadri et al.	
5,802,115	A	9/1998	Meyer	
5,881,093	A	3/1999	Wang et al.	
5,956,351	A *	9/1999	Bossen et al.	714/757
6,014,411	A	1/2000	Wang	
6,023,783	A	2/2000	Divsalar et al.	
6,031,874	A	2/2000	Chennakeshu et al.	
6,032,284	A	2/2000	Bliss	
6,044,116	A	3/2000	Wang	
6,094,739	A	7/2000	Miller et al.	
6,195,396	B1	2/2001	Fang et al.	
6,396,423	B1	5/2002	Laumen et al.	
6,437,714	B1	8/2002	Kim et al.	
6,732,328	B1	5/2004	McEwen et al.	

(21) Appl. No.: **13/073,947**

(22) Filed: **Mar. 28, 2011**

(65) **Prior Publication Data**

US 2011/0264985 A1 Oct. 27, 2011

Related U.S. Application Data

(63) Continuation of application No. 12/165,606, filed on Jun. 30, 2008, now Pat. No. 7,916,781, which is a continuation of application No. 11/542,950, filed on Oct. 3, 2006, now Pat. No. 7,421,032, which is a continuation of application No. 09/861,102, filed on May 18, 2001, now Pat. No. 7,116,710, which is a continuation-in-part of application No. 09/922,852, filed on Aug. 18, 2000, now Pat. No. 7,089,477.

(60) Provisional application No. 60/205,095, filed on May 18, 2000.

(51) **Int. Cl.**
H04B 1/66 (2006.01)

(52) **U.S. Cl.** **375/240; 375/285; 375/296; 714/801; 714/804**

(Continued)

OTHER PUBLICATIONS

Aji, S.M., et al., "The Generalized Distributive Law," IEEE Transactions on Information Theory, 46(2):325-343, Mar. 2000.

(Continued)

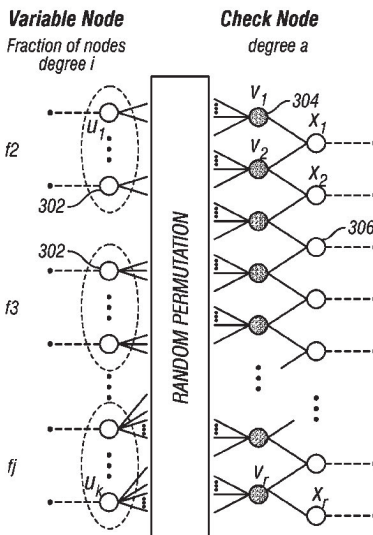
Primary Examiner — Dac Ha

(74) Attorney, Agent, or Firm — Perkins Coie LLP

(57) **ABSTRACT**

A serial concatenated coder includes an outer coder and an inner coder. The outer coder irregularly repeats bits in a data block according to a degree profile and scrambles the repeated bits. The scrambled and repeated bits are input to an inner coder, which has a rate substantially close to one.

14 Claims, 5 Drawing Sheets



US 8,284,833 B2

Page 2

U.S. PATENT DOCUMENTS

6,859,906	B2	2/2005	Hammons et al.	
7,089,477	B1	8/2006	Divsalar et al.	
7,116,710	B1	10/2006	Jin et al.	
7,421,032	B2	9/2008	Jin et al.	
7,916,781	B2	3/2011	Jin et al.	
7,934,146	B2*	4/2011	Stolpman	714/800
2001/0025358	A1	9/2001	Eidson et al.	
2007/0025450	A1	2/2007	Jin et al.	
2008/0263425	A1*	10/2008	Lakkis	714/752
2008/0294964	A1	11/2008	Jin et al.	

OTHER PUBLICATIONS

Benedetto, S., et al., "A Soft-Input Soft-Output APP Module for Iterative Decoding of Concatenated Codes," IEEE Communications Letters, 1(1):22-24, Jan. 1997.

Benedetto, S., et al., "A Soft-Input Soft-Output Maximum a Posteriori (MAP) Module to Decode Parallel and Serial Concatenated Codes," The Telecommunications and Data Acquisition Progress Report (TDA PR 42-127), pp. 1-20, Nov. 1996.

Benedetto, S., et al., "Bandwidth efficient parallel concatenated coding schemes," Electronics Letters, 31 (24):2067-2069, Nov. 1995.

Benedetto, S., et al., "Design of Serially Concatenated Interleaved Codes," ICC 97, vol. 2, pp. 710-714, Jun. 1997.

Benedetto, S., et al., "Parallel Concatenated Trellis Coded Modulation," ICC 96, vol. 2, pp. 974-978, Jun. 1996.

Benedetto, S., et al., "Serial Concatenated Trellis Coded Modulation with Iterative Decoding," Proceedings 1997 IEEE International Symposium on Information Theory (ISIT), Ulm, Germany, p. 8, Jun. 29-Jul. 4, 1997.

Benedetto, S., et al., "Serial Concatenation of Interleaved Codes: Performance Analysis, Design, and Iterative Decoding," The Telecommunications and Data Acquisition Progress Report (TDA PR 42-126), pp. 1-26, Aug. 1996.

Benedetto, S., et al., "Serial concatenation of interleaved codes: performance analysis, design, and iterative decoding," Proceedings 1997 IEEE International Symposium on Information Theory (ISIT), Ulm, Germany, p. 106, Jun. 29-Jul. 4, 1997.

Benedetto, S., et al., "Soft-Output Decoding Algorithms in Iterative Decoding of Turbo Codes," The Telecommunications and Data Acquisition Progress Report (TDA PR 42-124), pp. 63-87, Feb. 1996.

Berrou, C., et al., "Near Shannon Limit Error—Correcting Coding and Decoding: Turbo Codes," ICC 93, vol. 2, pp. 1064-1070, May 1993.

Digital Video Broadcasting (DVB)—User guidelines for the second generation system for Broadcasting, Interactive Services, News

Gathering and other broadband satellite applications (DVB-S2), ETSI TR 102 376 V1.1.1 Technical Report, pp. 1-104 (p. 64), Feb. 2005.

Divsalar, D., et al., "Coding Theorems for 'Turbo-Like' Codes," Proceedings of the 36th Annual Allerton Conference on Communication, Control, and Computing, Monticello, Illinois, pp. 201-210, Sep. 1998.

Divsalar, D., et al., "Effective free distance of turbo codes," Electronics Letters, 32(5):445-446, Feb. 1996.

Divsalar, D., et al., "Hybrid Concatenated Codes and Iterative Decoding," Proceedings 1997 IEEE International Symposium on Information Theory (ISIT), Ulm, Germany, p. 10, Jun. 29-Jul. 4, 1997.

Divsalar, D., et al., "Low-Rate Turbo Codes for Deep-Space Communications," Proceedings 1995 IEEE International Symposium on Information Theory (ISIT), Whistler, BC, Canada, p. 35, Sep. 1995.

Divsalar, D., et al., "Multiple Turbo Codes for Deep-Space Communications," The Telecommunications and Data Acquisition Progress Report (TDA PR 42-121), pp. 66-77, May 1995.

Divsalar, D., et al., "Multiple Turbo Codes," MILCOM '95, vol. 1, pp. 279-285, Nov. 1995.

Divsalar, D., et al., "On the Design of Turbo Codes," The Telecommunications and Data Acquisition Progress Report (TDA PR 42-123), pp. 99-121, Nov. 1995.

Divsalar, D., et al., "Serial Turbo Trellis Coded Modulation with Rate-1 Inner Code," Proceedings 2000 IEEE International Symposium on Information Theory (ISIT), Sorrento, Italy, pp. 194, Jun. 2000.

Divsalar, D., et al., "Turbo Codes for PCS Applications," IEEE ICC '95, Seattle, WA, USA, vol. 1, pp. 54-59, Jun. 1995.

Jin, H., et al., "Irregular Repeat—Accumulate Codes," 2nd International Symposium on Turbo Codes, Brest, France, 25 pages, Sep. 2000.

Jin, H., et al., "Irregular Repeat—Accumulate Codes," 2nd International Symposium on Turbo Codes & Related Topics, Brest, France, p. 1-8, Sep. 2000.

Richardson, T.J., et al., "Design of Capacity-Approaching Irregular Low-Density Parity-Check Codes," IEEE Transactions on Information Theory, 47(2):619-637, Feb. 2001.

Richardson, T.J., et al., "Efficient Encoding of Low-Density Parity-Check Codes," IEEE Transactions on Information Theory, 47(2):638-656, Feb. 2001.

Tanner, R.M., "A Recursive Approach to Low Complexity Codes," IEEE Transactions on Information Theory, 27 (5):533-547, Sep. 1981.

Wiberg, N., et al., "Codes and Iterative Decoding on General Graphs," Proceedings 1995 IEEE International Symposium on Information Theory (ISIT), Whistler, BC, Canada, p. 468, Sep. 1995.

* cited by examiner

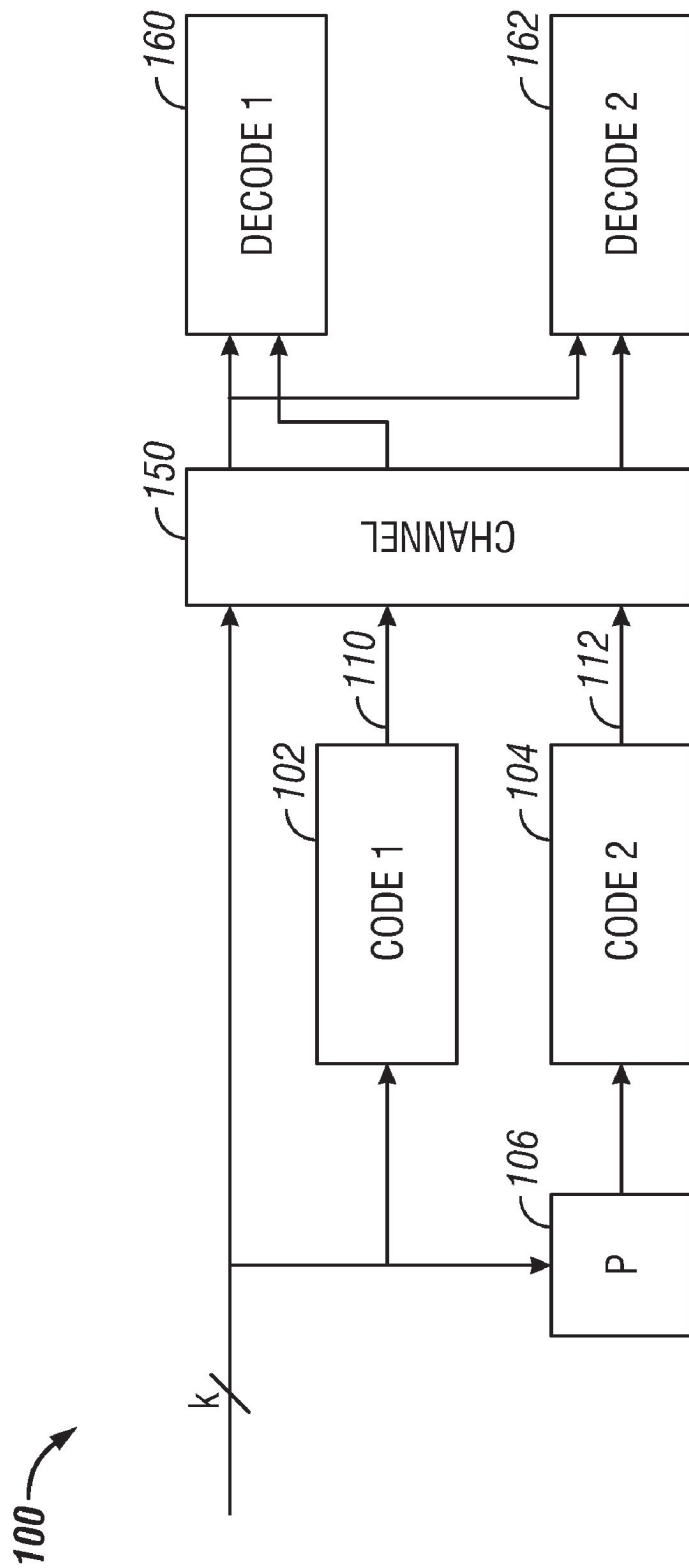


FIG. 1
(Prior Art)

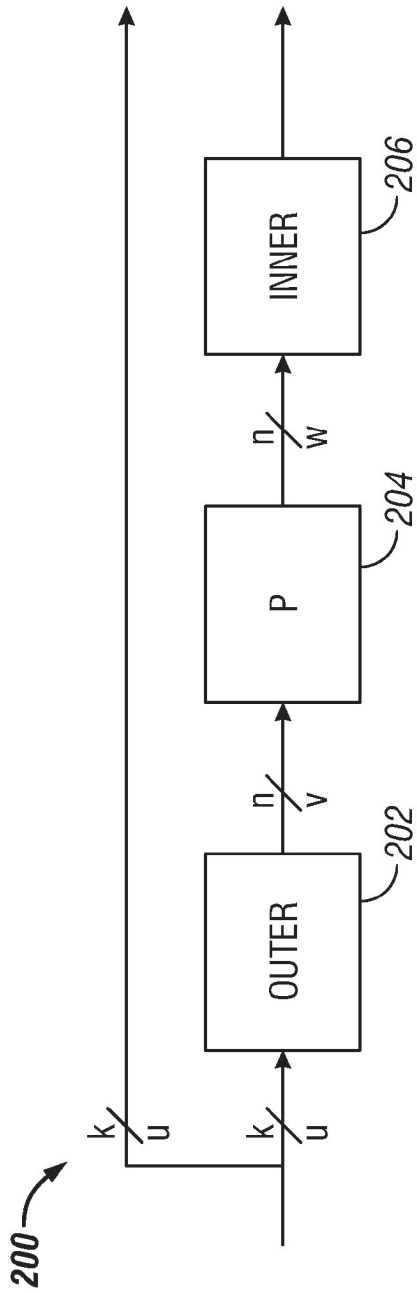


FIG. 2

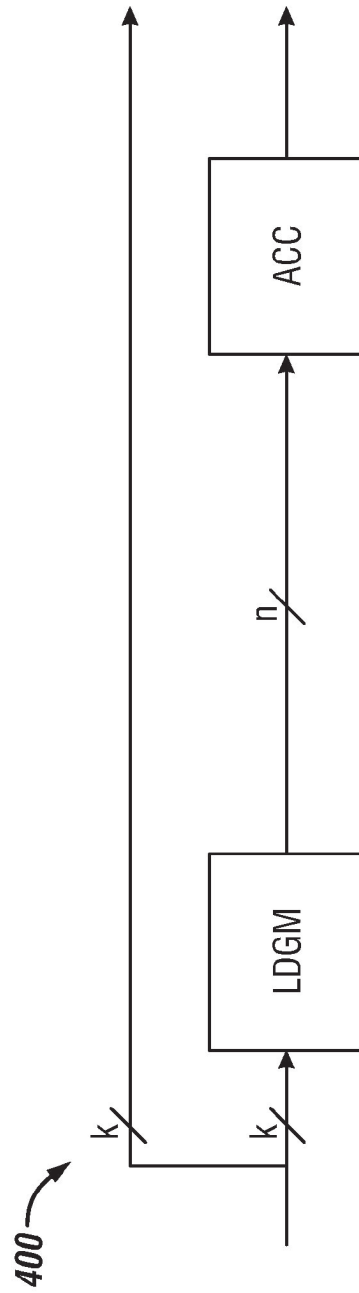


FIG. 4

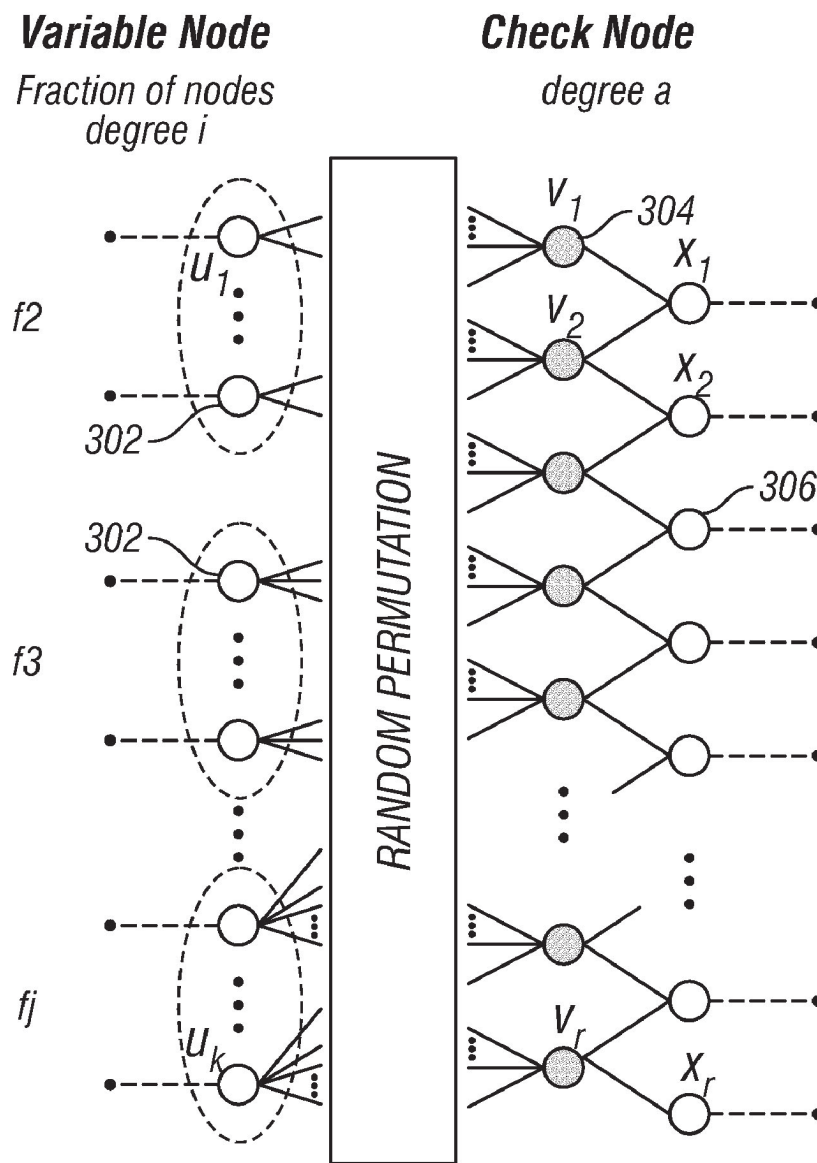


FIG. 3

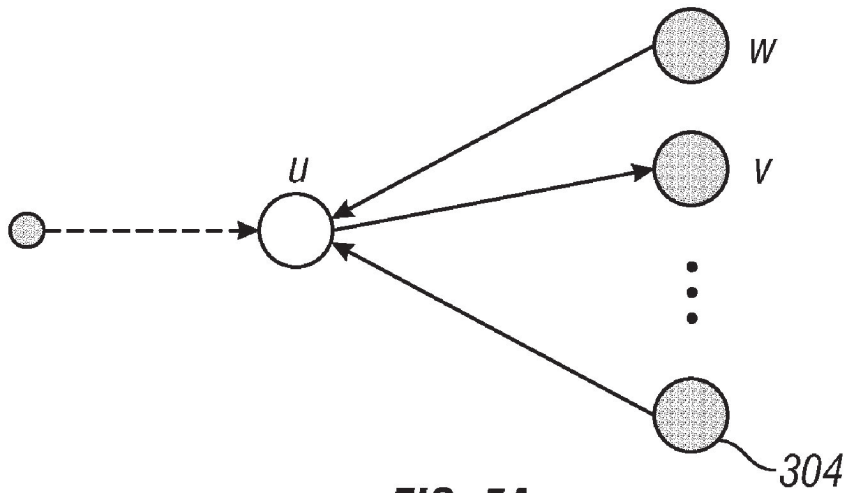


FIG. 5A

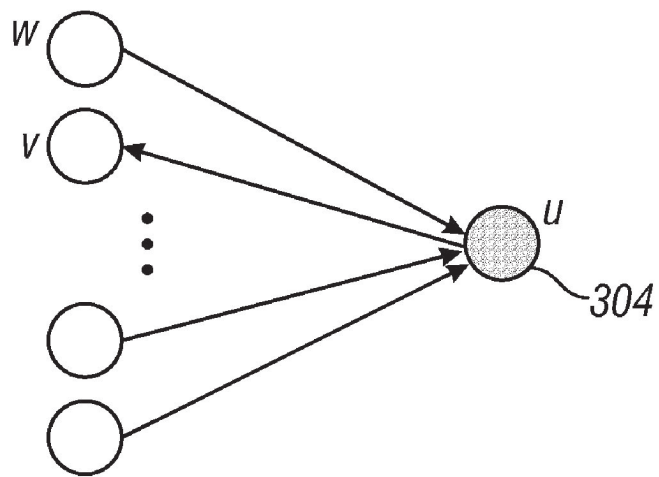


FIG. 5B

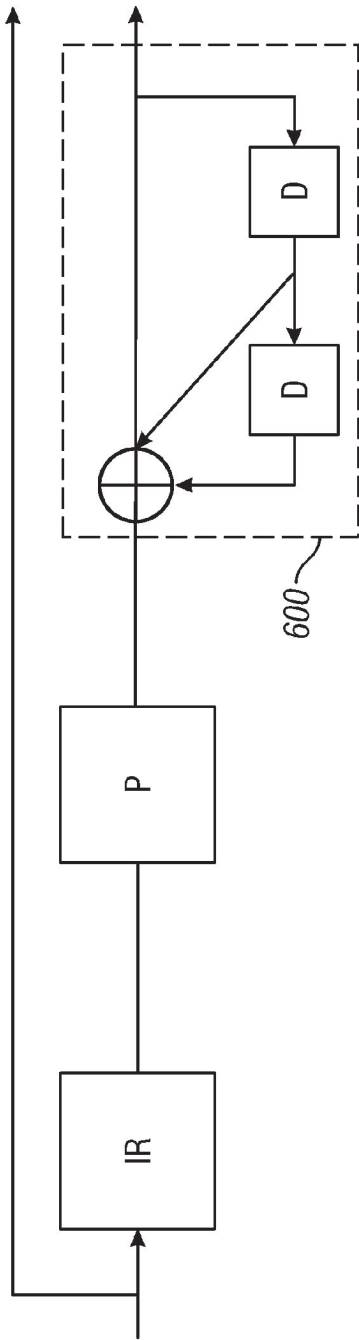


FIG. 6

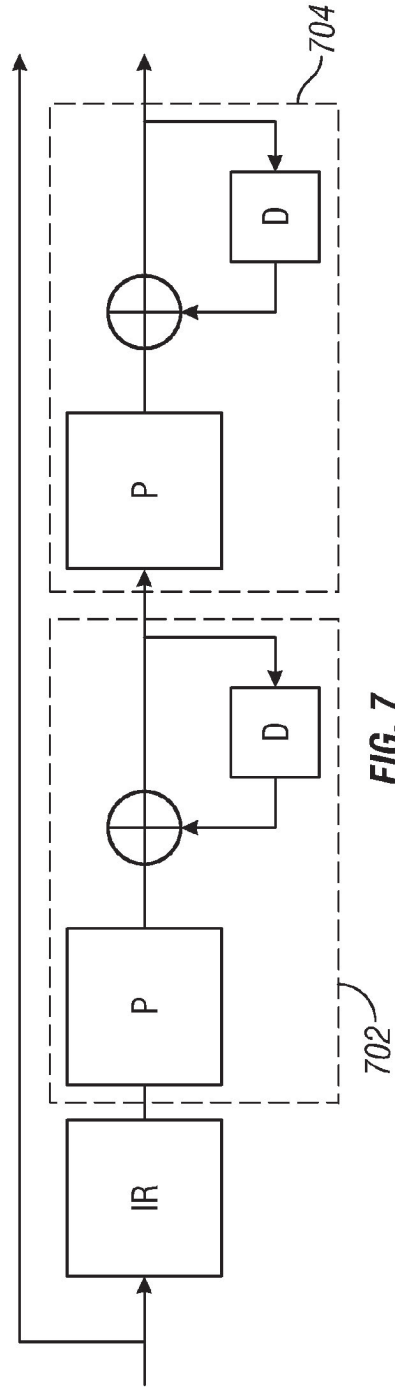


FIG. 7

US 8,284,833 B2

1

SERIAL CONCATENATION OF INTERLEAVED CONVOLUTIONAL CODES FORMING TURBO-LIKE CODES

CROSS-REFERENCE TO RELATED APPLICATIONS

This application is a continuation of U.S. application Ser. No. 12/165,606, filed Jun. 30, 2008 now U.S. Pat. No. 7,916,781, which is a continuation of U.S. application Ser. No. 11/542,950, filed Oct. 3, 2006, now U.S. Pat. No. 7,421,032, which is a continuation of U.S. application Ser. No. 09/861,102, filed May 18, 2001, now U.S. Pat. No. 7,116,710, which claims the priority of U.S. Provisional Application Ser. No. 60/205,095, filed May 18, 2000, and is a continuation-in-part of U.S. application Ser. No. 09/922,852, filed Aug. 18, 2000, now U.S. Pat. No. 7,089,477. The disclosures of the prior applications are considered part of (and are incorporated by reference in) the disclosure of this application.

GOVERNMENT LICENSE RIGHTS

The U.S. Government has a paid-up license in this invention and the right in limited circumstances to require the patent owner to license others on reasonable terms as provided for by the terms of Grant No. CCR-9804793 awarded by the National Science Foundation.

BACKGROUND

Properties of a channel affect the amount of data that can be handled by the channel. The so-called "Shannon limit" defines the theoretical limit of the amount of data that a channel can carry.

Different techniques have been used to increase the data rate that can be handled by a channel. "Near Shannon Limit Error-Correcting Coding and Decoding: Turbo Codes," by Berrou et al. ICC, pp 1064-1070, (1993), described a new "turbo code" technique that has revolutionized the field of error correcting codes. Turbo codes have sufficient randomness to allow reliable communication over the channel at a high data rate near capacity. However, they still retain sufficient structure to allow practical encoding and decoding algorithms. Still, the technique for encoding and decoding turbo codes can be relatively complex.

A standard turbo coder **100** is shown in FIG. 1. A block of k information bits is input directly to a first coder **102**. A k bit interleaver **106** also receives the k bits and interleaves them prior to applying them to a second coder **104**. The second coder produces an output that has more bits than its input, that is, it is a coder with rate that is less than 1. The coders **102**, **104** are typically recursive convolutional coders.

Three different items are sent over the channel **150**: the original k bits, first encoded bits **110**, and second encoded bits **112**. At the decoding end, two decoders are used: a first constituent decoder **160** and a second constituent decoder **162**. Each receives both the original k bits, and one of the encoded portions **110**, **112**. Each decoder sends likelihood estimates of the decoded bits to the other decoders. The estimates are used to decode the uncoded information bits as corrupted by the noisy channel.

SUMMARY

A coding system according to an embodiment is configured to receive a portion of a signal to be encoded, for example, a data block including a fixed number of bits. The

2

coding system includes an outer coder, which repeats and scrambles bits in the data block. The data block is apportioned into two or more sub-blocks, and bits in different sub-blocks are repeated a different number of times according to a selected degree profile. The outer coder may include a repeater with a variable rate and an interleaver. Alternatively, the outer coder may be a low-density generator matrix (LDGM) coder.

The repeated and scrambled bits are input to an inner coder that has a rate substantially close to one. The inner coder may include one or more accumulators that perform recursive modulo two addition operations on the input bit stream.

The encoded data output from the inner coder may be transmitted on a channel and decoded in linear time at a destination using iterative decoding techniques. The decoding techniques may be based on a Tanner graph representation of the code.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a schematic diagram of a prior "turbo code" system.

FIG. 2 is a schematic diagram of a coder according to an embodiment.

FIG. 3 is a Tanner graph for an irregular repeat and accumulate (IRA) coder.

FIG. 4 is a schematic diagram of an IRA coder according to an embodiment.

FIG. 5A illustrates a message from a variable node to a check node on the Tanner graph of FIG. 3.

FIG. 5B illustrates a message from a check node to a variable node on the Tanner graph of FIG. 3.

FIG. 6 is a schematic diagram of a coder according to an alternate embodiment.

FIG. 7 is a schematic diagram of a coder according to another alternate embodiment.

DETAILED DESCRIPTION

FIG. 2 illustrates a coder **200** according to an embodiment. The coder **200** may include an outer coder **202**, an interleaver **204**, and inner coder **206**. The coder may be used to format blocks of data for transmission, introducing redundancy into the stream of data to protect the data from loss due to transmission errors. The encoded data may then be decoded at a destination in linear time at rates that may approach the channel capacity.

The outer coder **202** receives the uncoded data. The data may be partitioned into blocks of fixed size, say k bits. The outer coder may be an (n,k) binary linear block coder, where $n > k$. The coder accepts as input a block u of k data bits and produces an output block v of n data bits. The mathematical relationship between u and v is $v = T_o u$, where T_o is an $n \times k$ matrix, and the rate of the coder is k/n .

The rate of the coder may be irregular, that is, the value of T_o is not constant, and may differ for sub-blocks of bits in the data block. In an embodiment, the outer coder **202** is a repeater that repeats the k bits in a block a number of times q to produce a block with n bits, where $n = qk$. Since the repeater has an irregular output, different bits in the block may be repeated a different number of times. For example, a fraction of the bits in the block may be repeated two times, a fraction of bits may be repeated three times, and the remainder of bits may be repeated four times. These fractions define a degree sequence, or degree profile, of the code.

The inner coder **206** may be a linear rate-1 coder, which means that the n -bit output block x can be written as $x = T_w w$,

3

where T_r is a nonsingular $n \times n$ matrix. The inner coder **210** can have a rate that is close to 1, e.g., within 50%, more preferably 10% and perhaps even more preferably within 1% of 1.

In an embodiment, the inner coder **206** is an accumulator, which produces outputs that are the modulo two (mod-2) partial sums of its inputs. The accumulator may be a truncated rate-1 recursive convolutional coder with the transfer function $1/(1+D)$. Such an accumulator may be considered a block coder whose input block $[x_1, \dots, x_n]$ and output block $[y_1, \dots, y_n]$ are related by the formula

$$\begin{aligned} y_1 &= x_1 \\ y_2 &= x_1 \oplus x_2 \\ y_3 &= x_1 \oplus x_2 \oplus x_3 \\ &\vdots \\ y_n &= x_1 \oplus x_2 \oplus x_3 \oplus \dots \oplus x_n \end{aligned}$$

where “ \oplus ” denotes mod-2, or exclusive-OR (XOR), addition. An advantage of this system is that only mod-2 addition is necessary for the accumulator. The accumulator may be embodied using only XOR gates, which may simplify the design.

The bits output from the outer coder **202** are scrambled before they are input to the inner coder **206**. This scrambling may be performed by the interleaver **204**, which performs a pseudo-random permutation of an input block v , yielding an output block w having the same length as v .

The serial concatenation of the interleaved irregular repeat code and the accumulate code produces an irregular repeat and accumulate (IRA) code. An IRA code is a linear code, and as such, may be represented as a set of parity checks. The set of parity checks may be represented in a bipartite graph, called the Tanner graph, of the code. FIG. 3 shows a Tanner graph **300** of an IRA code with parameters $(f_1, \dots, f_j; a)$, where $f_i \geq 0$, $\sum_i f_i = 1$ and “ a ” is a positive integer. The Tanner graph includes two kinds of nodes: variable nodes (open circles) and check nodes (filled circles). There are k variable nodes **302** on the left, called information nodes. There are r variable nodes **306** on the right, called parity nodes. There are $r = (k \sum_i f_i) / a$ check nodes **304** connected between the information nodes and the parity nodes. Each information node **302** is connected to a number of check nodes **304**. The fraction of information nodes connected to exactly i check nodes is f_i . For example, in the Tanner graph **300**, each of the f_2 information nodes are connected to two check nodes, corresponding to a repeat of $q=2$, and each of the f_3 information nodes are connected to three check nodes, corresponding to $q=3$.

Each check node **304** is connected to exactly “ a ” information nodes **302**. In FIG. 3, $a=3$. These connections can be made in many ways, as indicated by the arbitrary permutation of the ra edges joining information nodes **302** and check nodes **304** in permutation block **310**. These connections correspond to the scrambling performed by the interleaver **204**.

In an alternate embodiment, the outer coder **202** may be a low-density generator matrix (LDGM) coder that performs an irregular repeat of the k bits in the block, as shown in FIG. 4. As the name implies, an LDGM code has a sparse (low-density) generator matrix. The IRA code produced by the coder **400** is a serial concatenation of the LDGM code and the accumulator code. The interleaver **204** in FIG. 2 may be excluded due to the randomness already present in the structure of the LDGM code.

If the permutation performed in permutation block **310** is fixed, the Tanner graph represents a binary linear block code with k information bits (u_1, \dots, u_k) and r parity bits

4

(x_1, \dots, x_r) , as follows. Each of the information bits is associated with one of the information nodes **302**, and each of the parity bits is associated with one of the parity nodes **306**. The value of a parity bit is determined uniquely by the condition that the mod-2 sum of the values of the variable nodes connected to each of the check nodes **304** is zero. To see this, set $x_0=0$. Then if the values of the bits on the ra edges coming out the permutation box are (v_1, \dots, v_{ra}) , then we have the recursive formula for $j=1, 2, \dots, r$. This is in effect the encoding algorithm.

Two types of IRA codes are represented in FIG. 3, a non-systematic version and

$$x_j = x_{j-1} + \sum_{i=1}^a v_{(j-1)a+i}$$

a systematic version. The nonsystematic version is an (r,k) code, in which the codeword corresponding to the information bits (u_1, \dots, u_k) is (x_1, \dots, x_r) . The systematic version is a $(k+r, k)$ code, in which the codeword is $(u_1, \dots, u_k; x_1, \dots, x_r)$.

$$R_{n\text{sys}} = \frac{a}{\sum_i f_i}$$

The rate of the nonsystematic code is

The rate of the systematic code is

$$R_{\text{sys}} = \frac{a}{a + \sum_i f_i}$$

For example, regular repeat and accumulate (RA) codes can be considered nonsystematic IRA codes with $a=1$ and exactly one f_i equal to 1, say $f_q=1$, and the rest zero, in which case $R_{n\text{sys}}$ simplifies to $R=1/q$.

The IRA code may be represented using an alternate notation. Let λ_j be the fraction of edges between the information nodes **302** and the check nodes **304** that are

$$f_i = \frac{\lambda_i / i}{\sum_j \lambda_j / j}$$

adjacent to an information node of degree i , and let ρ_i be the fraction of such edges that are adjacent to a check node of degree $i+2$ (i.e., one that is adjacent to i information nodes). These edge fractions may be used to represent the IRA code rather than the corresponding node fractions. Define $\lambda(x) = \sum_i \lambda_i x^{i-1}$ and $\rho(x) = \sum_i \rho_i x^{i-1}$ to be the generating functions of these sequences. The pair (λ, ρ) is called a degree distribution. For $L(x) = \sum_i f_i x^i$,

The rate of the systematic IRA code given by the degree distribution is given by

$$L(x) = \int_0^x \lambda(t) dt / \int_0^1 \lambda(t) dt$$

-continued

$$\text{Rate} = \left(1 + \frac{\sum_j \rho_j / j}{\sum_j \lambda_j / j} \right)^{-1}$$

“Belief propagation” on the Tanner Graph realization may be used to decode IRA codes. Roughly speaking, the belief propagation decoding technique allows the messages passed on an edge to represent posterior densities on the bit associated with the variable node. A probability density on a bit is a pair of non-negative real numbers $p(0)$, $p(1)$ satisfying $p(0)+p(1)=1$, where $p(0)$ denotes the probability of the bit being 0, $p(1)$ the probability of it being 1. Such a pair can be represented by its log likelihood ratio, $m=\log(p(0)/p(1))$. The outgoing message from a variable node u to a check node v represents information about u , and a message from a check node u to a variable node v represents information about u , as shown in FIGS. 5A and 5B, respectively.

The outgoing message from a node u to a node v depends on the incoming messages from all neighbors w of u except v . If u is a variable message node, this outgoing message is

$$m(u \rightarrow v) = \sum_{w \neq v} m(w \rightarrow u) + m_0(u)$$

where $m_0(u)$ is the log-likelihood message associated with u . If u is a check node, the

$$\tanh \frac{m(u \rightarrow v)}{2} = \prod_{w \neq v} \tanh \frac{m(w \rightarrow u)}{2}$$

corresponding formula is

Before decoding, the messages $m(w \rightarrow u)$ and $m(u \rightarrow v)$ are initialized to be zero, and $m_0(u)$ is initialized to be the log-likelihood ratio based on the channel received information. If the channel is memoryless, i.e., each channel output only relies on its input, and y is the output of the channel code bit u , then $m_0(u)=\log(p(u=0|y)/p(u=1|y))$. After this initialization, the decoding process may run in a fully parallel and local manner. In each iteration, every variable/check node receives messages from its neighbors, and sends back updated messages. Decoding is terminated after a fixed number of iterations or detecting that all the constraints are satisfied. Upon termination, the decoder outputs a decoded sequence based on the messages $m(u)=\sum w_m(w \rightarrow u)$.

Thus, on various channels, iterative decoding only differs in the initial messages $m_0(u)$. For example, consider three memoryless channel models: a binary erasure channel (BEC); a binary symmetric channel (BSC); and an additive white Gaussian noise (AGWN) channel.

In the BEC, there are two inputs and three outputs. When 0 is transmitted, the receiver can receive either 0 or an erasure E. An erasure E output means that the receiver does not know how to demodulate the output. Similarly, when 1 is transmitted, the receiver can receive either 1 or E. Thus, for the BEC, $y \in \{0, E, 1\}$, and

In the BSC, there are two possible inputs (0,1) and two possible outputs (0, 1).

$$m_0(u) = \begin{cases} +\infty & \text{if } y = 0 \\ 0 & \text{if } y = E \\ -\infty & \text{if } y = 1 \end{cases}$$

The BSC is characterized by a set of conditional probabilities relating all possible outputs to possible inputs. Thus, for the BSC $y \in \{0, 1\}$, and

$$m_0(u) = \begin{cases} \log \frac{1-p}{p} & \text{if } y = 0 \\ -\log \frac{1-p}{p} & \text{if } y = 1 \end{cases}$$

In the AWGN, the discrete-time input symbols X take their values in a finite alphabet while channel output symbols Y can take any values along the real line. There is assumed to be no distortion or other effects other than the addition of white Gaussian noise. In an AWGN with a Binary Phase Shift Keying (BPSK) signaling which maps 0 to the symbol with amplitude $\sqrt{E_s}$ and 1 to the symbol with amplitude $-\sqrt{E_s}$, output $y \in \mathbb{R}$, then

$$m_0(u) = 4y\sqrt{E_s}N_0$$

where $N_0/2$ is the noise power spectral density.

The selection of a degree profile for use in a particular transmission channel is a design parameter, which may be affected by various attributes of the channel. The criteria for selecting a particular degree profile may include, for example, the type of channel and the data rate on the channel. For example, Table 1 shows degree profiles that have been found to produce good results for an AWGN channel model.

TABLE 1

a	2	3	4
λ_2	0.139025	0.078194	0.054485
λ_3	0.2221555	0.128085	0.104315
λ_5		0.160813	
λ_6	0.638820	0.036178	0.126755
λ_{10}			0.229816
λ_{11}			0.016484
λ_{12}		0.108828	
λ_{13}		0.487902	
λ_{14}			
λ_{16}			
λ_{27}			0.450302
λ_{28}			0.017842
Rate	0.333364	0.333223	0.333218
σ_{GA}	1.1840	1.2415	1.2615
σ^*	1.1981	1.2607	1.2780
(E_b/N_0) * (dB)	0.190	-0.250	-0.371
S.L. (dB)	-0.4953	-0.4958	-0.4958

Table 1 shows degree profiles yielding codes of rate approximately $1/3$ for the AWGN channel and with $a=2, 3, 4$. For each sequence, the Gaussian approximation noise threshold, the actual sum-product decoding threshold and the corresponding energy per bit (E_b)-noise power (N_0) ratio in dB are given. Also listed is the Shannon limit (S.L.).

As the parameter “a” is increased, the performance improves. For example, for $a=4$, the best code found has an iterative decoding threshold of $E_b/N_0=-0.371$ dB, which is only 0.12 dB above the Shannon limit.

The accumulator component of the coder may be replaced by a “double accumulator” 600 as shown in FIG. 6. The

US 8,284,833 B2

7

double accumulator can be viewed as a truncated rate 1 convolutional coder with transfer function $1/(1+D+D^2)$.

Alternatively, a pair of accumulators may be added, as shown in FIG. 7. There are three component codes: the “outer” code **700**, the “middle” code **702**, and the “inner” code **704**. The outer code is an irregular repetition code, and the middle and inner codes are both accumulators.

IRA codes may be implemented in a variety of channels, including memoryless channels, such as the BEC, BSC, and AWGN, as well as channels having non-binary input, non-symmetric and fading channels, and/or channels with memory.

A number of embodiments have been described. Nevertheless, it will be understood that various modifications may be made without departing from the spirit and scope of the invention. Accordingly, other embodiments are within the scope of the following claims.

What is claimed is:

1. An apparatus for performing encoding operations, the apparatus comprising:

a first set of memory locations to store information bits;
a second set of memory locations to store parity bits;
a permutation module to read a bit from the first set of memory locations and combine the read bit to a bit in the second set of memory locations based on a corresponding index of the first set of memory locations and a corresponding index of the second set of memory locations; and

an accumulator to perform accumulation operations on the bits stored in the second set of memory locations, wherein two or more memory locations of the first set of memory locations are read by the permutation module different times from one another.

2. The apparatus of claim **1**, wherein the permutation module is configured to perform the combine operation to include performing mod-2 or exclusive-OR sum.

3. The apparatus of claim **2**, wherein the permutation module is configured to perform the combining operation to further include writing the sum to the second set of memory locations based on a corresponding index.

4. The apparatus of claim **1**, wherein the accumulator is configured to perform the accumulation operation to include a mod-2 or exclusive-OR sum of the bit stored in a prior index to a bit stored in a current index based on a corresponding index of the second set of memory locations.

8

5. The apparatus of claim **4**, wherein the accumulator is configured to perform the accumulation operation to at least 2 consecutive indices of the second set of memory locations.

6. The apparatus of claim **1**, wherein the permutation module further comprises a permutation information module to generate pairs of an index of the first set of memory locations and an index of the second set of memory locations.

7. The apparatus of claim **6**, wherein at least one index of the second set of memory locations is used twice.

8. A method of performing encoding operations, the method comprising:

receiving a sequence of information bits from a first set of memory locations;

performing an encoding operation using the received sequence of information bits as an input, said encoding operation comprising:

reading a bit from the received sequence of information bits, and

combining the read bit to a bit in a second set of memory locations based on a corresponding index of the first set of memory locations for the received sequence of information bits and a corresponding index of the second set of memory locations; and

accumulating the bits in the second set of memory locations,

wherein two or more memory locations of the first set of memory locations are read by the permutation module different times from one another.

9. The method of claim **8**, wherein performing the combine operation comprises performing mod-2 or exclusive-OR sum.

10. The method of claim **9**, wherein performing the combine operation comprises writing the sum to the second set of memory locations based on a corresponding index.

11. The method of claim **8**, wherein performing the accumulation operation comprises performing a mod-2 or exclusive-OR sum of the bit stored in a prior index to a bit stored in a current index based on a corresponding index of the second set of memory locations.

12. The method of claim **8**, wherein the accumulation operation is performed to at least 2 consecutive indices of the second set of memory locations.

13. The method of claim **8**, wherein the combining operation comprises generating pairs of an index of the first set of memory locations and an index of the second set of memory locations.

14. The method of claim **13**, wherein at least one index of the second set of memory locations is used twice.

* * * * *

UNITED STATES PATENT AND TRADEMARK OFFICE
CERTIFICATE OF CORRECTION

PATENT NO. : 8,284,833 B2
APPLICATION NO. : 13/073947
DATED : October 9, 2012
INVENTOR(S) : Hui Jin et al.

Page 1 of 1

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

On the Title Page, in the Figures, insert Referral Tag -- 300 --.

On Title Page 2, Item (56), under "OTHER PUBLICATIONS", Line 19, delete "Performace" and insert -- Performance --, therefor.

In Fig. 3, Sheet 3 of 5, insert Referral Tag -- 300 --.

In Column 1, Line 38, delete "Bcrrou" and insert -- Berrou --, therefor.

In Column 3, Line 3, delete "1% of I." and insert -- 1% of 1. --, therefor.

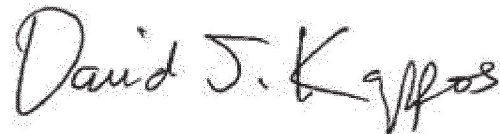
In Column 4, Line 31, delete "The rate of the nonsystematic code is" and insert the same at Line 25 as a new line.

In Column 4, Line 61, delete "The" and insert -- the --, therefor.

In Column 5, Line 39, delete "corresponding formula is" and insert the same in Line 32, after "node, the".

In Column 5, Line 59, delete "(AGWN)" and insert -- (AWGN) --, therefor.

Signed and Sealed this
Eighth Day of January, 2013



David J. Kappos
Director of the United States Patent and Trademark Office

UNITED STATES PATENT AND TRADEMARK OFFICE
CERTIFICATE OF CORRECTION

PATENT NO. : 8,284,833 B2
APPLICATION NO. : 13/073947
DATED : October 9, 2012
INVENTOR(S) : Hui Jin, Aamod Khandekar and Robert J. McEliece

Page 1 of 1

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

On the Title Page

Item [63], delete:

“Continuation of application No. 12/165,606, filed on Jun. 30, 2008, now Pat. No. 7,916,781, which is a continuation of application No. 11/542,950, filed on Oct. 3, 2006, now Pat. No. 7,421,032, which is a continuation of application No. 09/861,102, filed on May 18, 2001, now Pat. No. 7,116,710, which is a continuation-in-part of application No. 09/922,852, filed on Aug. 18, 2000, now Pat. No. 7,089,477.”

And insert:

-- Continuation of application No. 12/165,606, filed on Jun. 30, 2008, now Pat. No. 7,916,781, which is a continuation of application No. 11/542,950, filed on Oct. 3, 2006, now Pat. No. 7,421,032, which is a continuation of application No. 09/861,102, filed on May 18, 2001, now Pat. No. 7,116,710. --

In the Specification

Column 1, Line 8, delete:

“This application is a continuation of U.S. application Ser. No. 12/165,606, filed Jun. 30, 2008 now U.S. Pat. No. 7,916,781, which is a continuation of U.S. application Ser. No. 11/542,950, filed Oct. 3, 2006, now U.S. Pat. No. 7,421,032, which is a continuation of U.S. application Ser. No. 09/861,102, filed May 18, 2001, now U.S. Pat. No. 7,116,710, which claims the priority of U.S. Provisional Application Ser. No. 60/205,095, filed May 18, 2000, and is a continuation-in-part of U.S. application Ser. No. 09/922,852, filed Aug. 18, 2000, now U.S. Pat. No. 7,089,477.”

And insert:

-- This application is a continuation of U.S. application Ser. No. 12/165,606, filed Jun. 30, 2008, now U.S. Pat. No. 7,916,781, which is a continuation of U.S. application Ser. No. 11/542,950, filed Oct. 3, 2006, now U.S. Pat. No. 7,421,032, which is a continuation of U.S. application Ser. No. 09/861,102, filed May 18, 2001, now U.S. Pat. No. 7,116,710, which claims the priority of U.S. Provisional Application Ser. No. 60/205,095, filed May 18, 2000. --

Signed and Sealed this
Third Day of May, 2022



Katherine Kelly Vidal
Director of the United States Patent and Trademark Office

EXHIBIT E

Irregular Repeat–Accumulate Codes ¹

Hui Jin, Aamod Khandekar, and Robert McEliece

Department of Electrical Engineering, California Institute of Technology
Pasadena, CA 91125 USA

E-mail: {hui, aamod, rjm}@systems.caltech.edu

Abstract: *In this paper we will introduce an ensemble of codes called irregular repeat-accumulate (IRA) codes. IRA codes are a generalization of the repeat-accumulate codes introduced in [1], and as such have a natural linear-time encoding algorithm. We shall prove that on the binary erasure channel, IRA codes can be decoded reliably in linear time, using iterative sum-product decoding, at rates arbitrarily close to channel capacity. A similar result appears to be true on the AWGN channel, although we have no proof of this. We illustrate our results with numerical and experimental examples.*

Keywords: repeat-accumulate codes, turbo-codes, low-density parity-check codes, iterative decoding.

1. INTRODUCTION

With the hindsight provided by the past seven years of research in turbo-codes and low-density parity-check codes, one is tempted to propose the following problem as the final problem for channel coding researchers: *For a given channel, find an ensemble of codes with (1) a linear-time encoding algorithm, and (2) which can be decoded reliably in linear time at rates arbitrarily close to channel capacity.* For turbo-codes, both parallel and serial, (1) holds, but according to the recent work by Divsalar, Dolinar, and Pollara [7], on the AWGN channel there appears to be a gap, albeit usually not a large one, between channel capacity and the iterative decoding thresholds for any turbo ensemble. For LDPC codes, the natural encoding algorithm is quadratic in the block length, and from the work of Richardson and Urbanke [2] we know that for regular LDPC codes, on the binary symmetric and AWGN channels there is a gap between capacity and the iterative decoding thresholds. On the positive side, however, Luby, Shokrollahi et al. [3], [4], [8], have established the remarkable fact that on the binary erasure channel *irregular* LDPC codes satisfy (2). Recent work by Richardson, Shokrollahi and Urbanke [5] shows

that on the AWGN channel, irregular LDPC codes are markedly better than regular ones, but whether or not they can reach capacity is not yet known. In summary, as yet there is no known noisy channel for which the final problem has been solved, although researchers are very close on the AWGN channel and extremely close on the binary erasure channel.

In this paper, we will introduce a promising class of codes called *irregular repeat-accumulate* codes, which generalizes the repeat-accumulate codes of [1]. After defining the codes in Section 2, and observing that they have a simple linear-time encoding algorithm, in Section 3, using the powerful Richardson-Urbanke method [2], we will prove rigorously that IRA codes solve the final problem for the binary erasure channel. In Section 4, we will discuss, less rigorously, the performance of IRA codes on the AWGN channel, and show that their performance is remarkably good.

2. DEFINITION OF IRA CODES

Figure 1 shows a Tanner graph of an IRA code with parameters $(f_1, \dots, f_J; a)$, where $f_i \geq 0$, $\sum_i f_i = 1$ and a is a positive integer. The Tanner graph is a bipartite graph with two kinds of nodes: variable nodes (open circles) and check nodes (filled circles). There are k variable nodes on the left, called information nodes; there are $r = (k \sum_i i f_i)/a$ check nodes; and there are r variable nodes on the right, called parity nodes. Each information node is connected to a number of check nodes: the fraction of information nodes connected to exactly i check nodes is f_i . Each check node is connected to exactly a information nodes. These connections can be made in many ways, as indicated in Figure 1 by the “arbitrary permutation” of the ra edges joining information nodes and check nodes. The check nodes are connected to the parity nodes in the simple zigzag pattern shown in the figure.

If the “arbitrary permutation” in Figure 1 is fixed, the Tanner graph represents a binary linear code with k information bits (u_1, \dots, u_k) and r parity bits (x_1, \dots, x_r) , as follows. Each of the information bits is associated with one of the information nodes; and each of the parity bits is associated with one of the

¹This paper is to be presented at the Second International Conference on Turbo Codes, Brest, France, September 2000. This research was supported by NSF grant no. CCR-9804793, and grants from Sony, Qualcomm, and Caltech’s Lee Center for Advanced Networking.

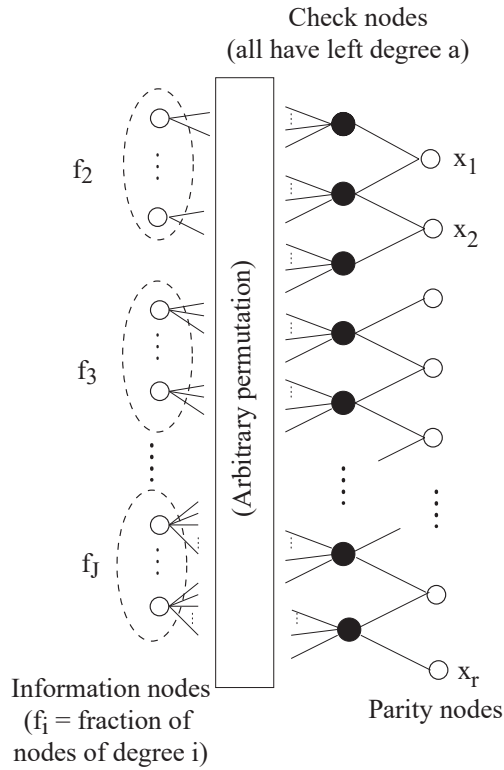


Figure 1: Tanner graph for IRA code with parameters $(f_1, \dots, f_J; a)$.

parity nodes. The value of a parity bit is determined uniquely by the condition that the mod-2 sum of the values of the variable nodes connected to each of the check nodes is zero. To see this, let us conventionally set $x_0 = 0$. Then if the values of the bits on the ra edges coming out of the permutation box are (v_1, \dots, v_{ra}) , we have the recursive formula

$$x_j = x_{j-1} + \sum_{i=1}^a v_{(j-1)a+i}, \quad (1)$$

for $j = 1, 2, \dots, r$. This is in effect the encoding algorithm, and so if a is fixed and $n \rightarrow \infty$, the encoding complexity is $O(n)$.

There are two versions of the IRA code in Figure 1: the *nonsystematic* and the *systematic* versions. The nonsystematic version is an (r, k) code, in which the codeword corresponding to the information bits (u_1, \dots, u_k) is (x_1, \dots, x_r) . The systematic version is a $(k+r, k)$ code, in which the codeword is

$$(u_1, \dots, u_k; x_1, \dots, x_r).$$

The rate of the *nonsystematic* code is easily seen to be

$$R_{\text{nsys}} = \frac{a}{\sum_i i f_i}, \quad (2)$$

whereas for the systematic code the rate is

$$R_{\text{sys}} = \frac{a}{a + \sum_i i f_i} \quad (3)$$

For example, the original RA codes are nonsystematic IRA codes with $a = 1$ and exactly one f_i equal to 1, say $f_q = 1$, and the rest zero, in which case (2) simplifies to $R = 1/q$. (However, in this paper we will be concerned almost exclusively with systematic IRA codes.)

In an iterative sum-product message-passing decoding algorithm, all messages are assumed to be log-likelihood ratios, i.e., of the form $m = \log(p(0)/p(1))$. The outgoing message from a variable node u to a check node v represents information about u , and a message from a check node u to a variable node v represents information about u . Initially, messages are sent from variable nodes which represent transmitted symbols.

The outgoing message from a node u to a node v depends on the incoming messages from all neighbors w of u except v . If u is a variable message node, this outgoing message is

$$m(u \rightarrow v) = \sum_{w \neq v} m(w \rightarrow u) + m_0(u), \quad (4)$$

where $m_0(u)$ is the log-likelihood message associated with u . (If u is not a codeword node, this term is absent.) If u is a check node the corresponding formula is [10]

$$\tanh \frac{m(u \rightarrow v)}{2} = \prod_{w \neq v} \tanh \frac{m(w \rightarrow u)}{2}. \quad (5)$$

3. IRA CODES ON THE BINARY ERASURE CHANNEL

The sum-product algorithm defined in equations (4) and (5) simplifies considerably on the binary erasure channel (BEC). The BEC is a binary input channel with three output symbols, a 0, a 1 and “erasure.” The input symbol is received as an erasure with probability p and is received correctly with probability $1-p$. It is important to note that no errors are ever made on this channel.

It is not difficult to see that the messages defined in (4) and (5) can assume only three values on the BEC, viz. $+\infty$, $-\infty$ or 0, corresponding to a variable value 0, 1, or “unknown.” No errors can occur during the running of the algorithm; if a message is $\pm\infty$, the corresponding variable is guaranteed to be 0 or 1, respectively. The operations at the nodes in the graph given by eqns (4) and (5) can be stated much more simply and intuitively in this case. At a variable node, the outgoing message is equal to any non-erasure incoming message, or an erasure if all incoming messages are erasures. At a check node, the outgoing message is an erasure if any incoming message is an erasure, and otherwise is the binary sum of all incoming messages.

3.1. Notation

In this section and the next, it will be convenient to use a slightly different representation for an IRA code than the one used in Section 2. Firstly, we will begin with the assumption that the degrees of both the information nodes and the check nodes are non-constant, though we will soon restrict attention to the “right-regular” case, in which the check nodes have constant degree.

Secondly, let λ_i be the fraction of *edges* between the information and the check nodes that are adjacent to an information node of degree i , and let ρ_i be the fraction of such edges that are adjacent to a check node of degree $i + 2$ (i.e. one which is adjacent to i information nodes). We will use these edge fractions λ_i and ρ_i to represent the IRA code rather than the corresponding node fractions. We define $\lambda(x) = \sum_i \lambda_i x^{i-1}$ and $\rho(x) = \sum_i \rho_i x^{i-1}$ to be the generating functions of these sequences. The pair (λ, ρ) is called a *degree distribution*. It is quite easy to convert between the two representations. We demonstrate the conversion with the information node degrees. Let the f_i 's be as defined in Section 2 and let $L(x) = \sum_i f_i x^i$. Then we have

$$f_i = \frac{\lambda_i/i}{\sum_j \lambda_j/j}, \quad (6)$$

$$L(x) = \int_0^x \lambda(t)dt / \int_0^1 \lambda(t)dt. \quad (7)$$

The rate of the systematic IRA code (we shall be dealing only with these) given by this degree distribution is given by

$$\text{Rate} = \left(1 + \frac{\sum_j \rho_j/j}{\sum_j \lambda_j/j}\right)^{-1} \quad (8)$$

(This is an easy exercise. For a proof, see [8].)

3.2. Fixed point analysis of iterative decoding

In [2], it was shown that if for a code ensemble, the probability of the *depth- l neighborhood* of an edge (in the Tanner graph) being cycle-free goes to 1 as the length of the code goes to infinity (we will call this condition the *cycle-free condition*), then *density evolution* gives an accurate estimate of the bit error rate after l iterations, again as the length of the codes goes to infinity. In density evolution, we evolve the probability density of the messages being passed according to the operations being performed on them, assuming that all incoming messages are independent (which is true if the depth- l neighbourhood is tree-like). The cycle-free condition does indeed hold

for IRA codes. The proof of this fact is almost exactly the same as in the irregular LDPC codes case, which was done in [2].

Now, in the case of the erasure channel, we have seen that the messages are only of three types, so in effect we have a discrete density function, and the probability of error is merely the probability of erasure. With this in mind, we will now study the evolution of the erasure probability, and derive conditions which guarantee that it goes to zero as the number of iterations goes to infinity. Under these conditions iterative decoding will be successful in the sense of [2], i.e., it will achieve arbitrarily small BERs, given enough iterations and long enough codes.

Let p be the channel probability of erasure. We will iterate the probability of erasure along the edges of the graph during the course of the algorithm. Let x_0 be the probability of erasure on an edge from an information node to a check node, x_1 the probability of erasure on an edge from a check node to a parity node, x_2 the probability of erasure on an edge from a parity node to a check node, and x_3 the probability of erasure on an edge from a check node to an information node. The initial probability of erasure on the message bits is p .

We now assume that we are at a fixed point of the decoding algorithm and solve for x_0 . We get the following equations:

$$x_1 = 1 - (1 - x_2)R(1 - x_0), \quad (9)$$

$$x_2 = px_1, \quad (10)$$

$$x_3 = 1 - (1 - x_2)^2\rho(1 - x_0), \quad (11)$$

$$x_0 = p\lambda(x_3). \quad (12)$$

where $R(x)$ is the polynomial in which the coefficient of x^i denotes the fraction of check nodes of degree i . $R(x)$ is given by (cf. eq. (7))

$$R(x) = \frac{\int_0^x \rho(t)dt}{\int_0^1 \rho(t)dt} \quad (13)$$

We eliminate x_1 from the first two of these equations to get x_2 in terms of x_0 and then keep substituting forwards to get an equation purely in x_0 , henceforth denoted by x . We thereby obtain the following equation for a fixed point of iterative decoding:

$$p\lambda\left(1 - \left[\frac{1-p}{1-pR(1-x)}\right]^2 \rho(1-x)\right) = x. \quad (14)$$

If this equation has no solution in the interval $(0, 1]$, then iterative decoding must converge to probability of erasure zero. Therefore, if we have

$$p\lambda \left(1 - \left[\frac{1-p}{1-pR(1-x)} \right]^2 \rho(1-x) \right) < x, \quad \forall x \neq 0. \quad (15)$$

then in the sense of [2], iterative decoding is successful.

3.3. Capacity-achieving sequences of degree distributions

We will now derive sequences of degree distributions that can be shown to achieve channel capacity. First, we restrict attention to the case $\rho(x) = x^{a-1}$ for some $a \geq 1$, since it turns out that we can achieve capacity even with this restriction. In this case, $R(x) = x^a$, and the condition for convergence to zero BER now becomes

$$p\lambda \left(1 - \left[\frac{1-p}{1-p(1-x)^a} \right]^2 (1-x)^{a-1} \right) < x, \quad \forall x \neq 0 \quad (16)$$

We now make the following new definitions

$$f_p(x) \triangleq 1 - \left[\frac{1-p}{1-p(1-x)^a} \right]^2 (1-x)^{a-1} \quad (17)$$

$$h_p(x) \triangleq 1 - \left[\frac{1-p}{1-p(1-x)^a} \right]^2 (1-x)^a \quad (18)$$

$$g_p(x) \triangleq h_p^{-1}(x) \quad (19)$$

Notice that $f_p(x)$, $h_p(x)$ and $g_p(x)$ are all monotonic functions in $[0, 1]$ and attain the values 0 at 0 and 1 at 1. In addition, $h_p(x)$ can be inverted by hand (by making the substitution $(1-x)^a = y$) and it can be shown that $g_p(x)$ has a power series expansion around 0 with non-negative coefficients. Let this expansion be $g_p(x) = \sum_i g_{p,i} x^i$.

Now, the condition (16) can now be rewritten as

$$p\lambda(f_p(x)) < x, \quad \forall x \neq 0 \quad (20)$$

which can be rewritten as

$$\lambda(x) < \frac{f_p^{-1}(x)}{p} \quad (21)$$

We make the following choice of $\lambda(x)$:

$$\lambda(x) = \frac{1}{p} \left(\sum_{i=1}^{N-1} g_{p,i} x^i + \epsilon x^N \right) \quad (22)$$

where $0 < \epsilon < g_{p,N}$ and $\sum_{i=1}^{N-1} g_{p,i} + \epsilon = p$. Such a choice of N and ϵ exists and is unique since the $g_{p,i}$'s are non-negative and $\sum_{i=1}^{\infty} g_{p,i} = g_p(1) = 1$. For this choice of $\lambda(x)$, we have

$$p\lambda(x) < g_p(x) = h_p^{-1}(x) < f_p^{-1}(x) \quad \forall x \neq 0 \quad (23)$$

where the last inequality follows because $f_p(x) < h_p(x) \quad \forall x \neq 0$.

Thus, the condition (21) for BER going to zero is satisfied and the degree distributions we have thus defined yield codes with thresholds that are greater than or equal to p . We now wish to compute the rate of these codes in the limit as $a \rightarrow \infty$ to show that they achieve channel capacity. The rate of the code is given by eq. (8) which simplifies to $(1 + (a \sum_i \lambda_i/i)^{-1})^{-1}$ in the right-regular case. Now,

$$\lim_{a \rightarrow \infty} a \sum_i \frac{\lambda_i}{i} = \lim_{a \rightarrow \infty} a \left(\sum_{i=1}^{N-1} \frac{g_{p,i}}{i} + \frac{\epsilon}{N} \right) \quad (24)$$

We also have

$$\lim_{a \rightarrow \infty} a \sum_{i=N}^{\infty} \frac{g_{p,i}}{i} \leq \lim_{a \rightarrow \infty} \frac{a}{N} \sum_{i=N}^{\infty} g_{p,i} \leq \lim_{a \rightarrow \infty} \frac{a}{N} = 0 \quad (25)$$

where the last equality is a property of the function $g_p(x)$ and is also proved by manual inversion of $h_p(x)$. We therefore have

$$\begin{aligned} \lim_{a \rightarrow \infty} a \sum_i \frac{\lambda_i}{i} &= \lim_{a \rightarrow \infty} a \sum_{i=1}^{\infty} \frac{g_{p,i}}{i} \\ &= \lim_{a \rightarrow \infty} a \int_0^1 g_p(x) dx \\ &= a \left(1 - \int_0^1 h_p(x) dx \right) \\ &= a \int_0^1 \left(\frac{1-p}{1-px^a} \right)^2 x^a dx. \end{aligned}$$

The integrand on the right can be expanded in a power series with non-negative coefficients, with the first non-zero coefficient being that of x^a . Keeping in mind that we are integrating this power series, it is easy to see that

$$\begin{aligned} &\frac{a}{a+1} \int_0^1 \left(\frac{1-p}{1-px^a} \right)^2 x^{a-1} dx \\ &< 1 - \int_0^1 h_p(x) dx \\ &< \int_0^1 \left(\frac{1-p}{1-px^a} \right)^2 x^{a-1} dx. \end{aligned} \quad (26)$$

Both bounds in the above equation can be computed easily and both tend to $(1-p)/p$ in the limit of large a . Plugging this result into the formula for the rate, we finally get that the rate tends to $1-p$ in the limit of large a , which is indeed the capacity of the BEC.

Thus the sequence of degree distributions given in eq. (22) does indeed achieve channel capacity.

3.4. Some numerical results

We have seen that the condition for BER going to zero at a channel erasure probability of p is $p\lambda(x) < f_p^{-1}(x) \forall x \neq 0$. We later enforced a stronger condition, namely $p\lambda(x) < h_p^{-1}(x) = g_p(x) \forall x \neq 0$ and derived capacity-achieving degree sequences satisfying this condition. The reason we needed to enforce the stronger condition was that $h_p^{-1}(x) = g_p(x)$ has non-negative power-series coefficients, while the same cannot be said for $f_p^{-1}(x)$. However, from (26) we see that enforcing this stronger condition costs us a factor of $1 - a/(a+1) = 1/(a+1)$ in the rate which is very large for values of a that are of interest, and therefore the resulting codes are not very good.

If, however, $f_p^{-1}(x)$ were to have non-negative power series coefficients, then we could use it to define a degree distribution and we would no longer lose this factor of $1/(a+1)$. We have found through direct numerical computation in all cases that we tried, that enough terms in the beginning of this power series are non-negative to enable us to define $\lambda(x)$ by an equation analogous to eq. (22), replacing $g_p(x)$ by $f_p^{-1}(x)$. Of course, the resulting code is not theoretically guaranteed to have a threshold $\geq p$, but numerical computation shows that the threshold is either equal to or very marginally less than p .

This design turns out to yield very powerful codes, in particular codes whose performance is in every way comparable to the irregular LDPC codes listed in [8] as far as decoding performance is concerned. The performance of some of these distributions is listed in Table 1. The threshold values p are the same as those in [8] for corresponding values of a (IRA codes with right degree $a+2$ should be compared to irregular LDPC codes with right degree a , so that the decoding complexity is about the same), so as to make comparison easy. The codes listed in [8] were shown to have certain optimality properties with respect to the tradeoff between $1 - \delta/(1 - R)$ (distance from capacity) and a (decoding complexity), so it is very heartening to note that the codes we have designed are comparable to these.

We end this section with a brief discussion of the case $a = 1$. In this case, it turns out that $f_p^{-1}(x)$ does indeed have non-negative power-series coefficients. The resulting degree sequences yield codes that are better than conventional RA codes at small rates. An entirely similar exercise can be carried out for the case of non-systematic RA codes with $a = 1$ and the codes resulting in this case are significantly better than conventional RA codes for most rates. However, non-systematic RA codes turn out to be useless for higher values of a , as can be seen by manually following the decoding algorithm for one iteration, which shows that decoding does not proceed at all. For this reason all the preceding analysis was

Table 1: Performance of some codes designed using the procedure described in Section 3.4. at rates close to $2/3$ and $1/2$. δ is the code threshold (maximum allowable value of p), N the number of terms in $\lambda(x)$, and R the rate of the code.

a	δ	N	$1 - R$	$\delta/(1 - R)$
4	0.20000	1	0.333333	0.6000
5	0.23611	3	0.317101	0.7448
6	0.28994	6	0.329412	0.8802
7	0.31551	11	0.336876	0.9366
8	0.32024	16	0.333850	0.9592
9	0.32558	26	0.334074	0.9744
4	0.48090	13	0.502141	0.9577
5	0.49287	28	0.502225	0.9814

performed for systematic RA codes.

4. IRA CODES ON THE AWGN CHANNEL

In this section, we will consider the behavior of IRA codes on the AWGN channel. Here there are only two possible inputs, 0 and 1, but the output alphabet is the set of real numbers: if the x is the input, then the output is $y = (-1)^x + z$, where z is a mean zero, variance σ^2 Gaussian random variable. For a given noise variance σ^2 , our objective will be to find a left degree sequence $\lambda(x)$ such that the ensemble message error probability approaches zero, while the rate is as large as possible. Unlike the BEC, where we deal only with probabilities, in the case of the AWGN we must deal with probability densities. This complicates the analysis, and forces us to resort to approximate design methods.

4.1. Gaussian Approximation

Wiberg [9] has shown that the messages passed in iterative decoding on the AWGN channel can be well approximated by Gaussian random variables, provided the messages are in log-likelihood ratio form. In [6], this approximation was used to design good LDPC codes for the AWGN channel.

In this subsection, we use this Gaussian approximation to design good IRA codes for the AWGN channel. Specifically, we approximate the messages from check nodes to variable nodes (both information and parity) as Gaussian at every iteration. For a variable node, if all the incoming messages are Gaussian, then all the outgoing messages are also Gaussian because of (4). A Gaussian distribution $f(x)$ is called *consistent* [5] if $f(x) = f(-x)e^x$ for $\forall x \leq 0$. The consistency condition implies that the mean and variance satisfy $\sigma^2 = 2\mu$. For the sum-product algorithm, it has been shown [2] that consistency is preserved at message updates of both the variable and

check nodes. Thus if we assume Gaussian messages, and require consistency, we only need to keep track of the means. To this end, we define a *consistent Gaussian density* with mean μ to be

$$G_\mu(z) = \frac{1}{\sqrt{4\pi\mu}} e^{-(z-\mu)^2/4\mu}. \quad (27)$$

The expected value of $\tanh \frac{z}{2}$ for a consistent Gaussian distributed random variable z with mean μ is then

$$E[\tanh \frac{z}{2}] = \int_{-\infty}^{+\infty} G_\mu(z) \tanh \frac{z}{2} dz \triangleq \phi(\mu). \quad (28)$$

It is easy to see that $\phi(u)$ is a monotonic increasing function of u ; we denote its inverse function by $\phi^{(-1)}(y)$. Let $\mu_L^{(l)}$ and $\mu_R^{(l)}$ be the means of the message from check nodes to variable nodes on the left (i.e., information nodes) and on the right (i.e., parity nodes) at the l th iteration. We want to obtain expressions for $\mu_L^{(l+1)}$ and $\mu_R^{(l+1)}$ in terms of $\mu_L^{(l)}$ and $\mu_R^{(l)}$. A message from a degree- i information node to a check node at the l th iteration, is Gaussian with mean $(i-1)\mu_L^{(l)} + \mu_o$, where μ_o is the mean of message m_o in (4). Hence if v_L denotes the message on a randomly selected edge from an information node to a check node, the density of v_L is

$$\sum_{i=1}^J \lambda_i G_{(i-1)\mu_L^{(l)} + \mu_o}(z). \quad (29)$$

From (29) and (28) we obtain:

$$E[\tanh \frac{v_L}{2}] = \sum_{i=1}^J \lambda_i \phi((i-1)\mu_L^{(l)} + \mu_o). \quad (30)$$

Similarly, if v_R denotes the message on a randomly selected edge from a parity node to a check node,

$$E[\tanh \frac{v_R}{2}] = \phi(\mu_R^{(l)} + \mu_o). \quad (31)$$

Because of (5) we have

$$E[\tanh \frac{m(u \rightarrow v)}{2}] = \prod_{w \neq v} E[\tanh \frac{m(w \rightarrow u)}{2}]. \quad (32)$$

Denote a message from a check node to an information node, resp. parity node, by u_L , resp. u_R . Replacing $E[\tanh \frac{m(w \rightarrow u)}{2}]$ with the right side of (30) or (31) depending upon whether the message comes from the left or right, (32) implies:

$$\begin{aligned} E[\tanh \frac{u_L}{2}] &= E[\tanh \frac{v_L}{2}]^{a-1} E[\tanh \frac{v_R}{2}]^2 \\ &= \left(\sum_{i=1}^J \lambda_i \phi((i-1)\mu_L^{(l)} + \mu_o) \right)^{a-1} (\phi(\mu_R^{(l)} + \mu_o))^2, \end{aligned}$$

$$\begin{aligned} E[\tanh \frac{u_R}{2}] &= E[\tanh \frac{v_L}{2}]^a E[\tanh \frac{v_R}{2}] \\ &= \left(\sum_{i=1}^J \lambda_i \phi((i-1)\mu_L^{(l)} + \mu_o) \right)^a \phi(\mu_R^{(l)} + \mu_o). \end{aligned}$$

Using the definition of $\phi(\mu)$ in (28), we thus have the following recursion for $\mu_L^{(l)}$ and $\mu_R^{(l)}$:

$$\begin{aligned} \phi(\mu_L^{(l+1)}) &= \left(\sum_{i=1}^J \lambda_i \phi((i-1)\mu_L^{(l)} + \mu_o) \right)^{a-1} \times \\ &\quad (\phi(\mu_R^{(l)} + \mu_o))^2, \end{aligned} \quad (33)$$

$$\begin{aligned} \phi(\mu_R^{(l+1)}) &= \left(\sum_{i=1}^J \lambda_i \phi((i-1)\mu_L^{(l)} + \mu_o) \right)^a \times \\ &\quad \phi(\mu_R^{(l)} + \mu_o). \end{aligned} \quad (34)$$

In order to have arbitrary small bit error probability, the means $\mu_L^{(l)}$ and $\mu_R^{(l)}$ should approach infinity as l approaches infinity. In the next subsection, we derive a sufficient condition for this.

4.2. Fixed point analysis

We now assume that iterative decoding has reached a fixed point of (33) and (34), i.e., $\mu_L^{(l+1)} = \mu_L^{(l)} = \mu_L$ and $\mu_R^{(l+1)} = \mu_R^{(l)} = \mu_R$. Denote $\sum_{i=1}^J \lambda_i \phi((i-1)\mu_L + \mu_o)$ by x . From (30) we can see that $0 < x < 1$ and $x \rightarrow 1$ if and only if $\mu_L \rightarrow \infty$. From (34) it's easy to show that μ_R is a function of x , denoted by f , i.e., $\mu_R = f(x)$. Then, dividing (33) by the square of (34) gives us:

$$\phi(\mu_L) = \phi^2(\mu_R)/x^{a+1} = \phi^2(f(x))/x^{a+1}. \quad (35)$$

Now replacing μ_L with $\phi^{(-1)}(\phi^2(f(x))/x^{a+1})$ into the definition of x , we obtain the following equation for the fixed point x :

$$x = \sum_{i=1}^J \lambda_i \phi(\mu_o + (i-1)\phi^{(-1)}(\frac{\phi^2(f(x))}{x^{a+1}})). \quad (36)$$

If this equation doesn't have a solution in the interval $[0, 1]$, then the decoding bit error probability converges to zero. Therefore, if we have

$$F(x) \triangleq \sum_{i=1}^J \lambda_i \phi(\mu_o + (i-1)\phi^{(-1)}(\frac{\phi^2(f(x))}{x^{a+1}})) > x, \quad (37)$$

for any $x \in [x_0, 1)$, where x_0 is the value of x at the first iteration, then (the Gaussian approximation to) iterative decoding is successful.

Since the rate of the code is given by (cf. (8)):

$$\frac{\sum_i \lambda_i / i}{1/a + \sum_i \lambda_i / i}, \quad (38)$$

to maximize the rate, we should maximize $\sum_i \lambda_i/i$. Thus, under the Gaussian approximation, the problem of finding a good degree sequence for IRA codes is converted to the following linear programming problem:

Linear Programming Problem. Maximize

$$\sum_{i=1}^J \lambda_i/i, \quad (39)$$

under the condition

$$F(x) > x, \quad \forall x \in [x_0, 1]. \quad (40)$$

We have designed some degree sequences for IRA codes using this linear programming methodology. The results are presented in Tables 2 (code rate $\approx 1/3$) and 3 (code rate $\approx 1/2$). After using the heuristic Gaussian approximation method to design the degree sequences, we used exact density evolution to determine the actual noise threshold. (In every case, the true iterative decoding threshold was better than the one predicted by the Gaussian approximation.)

a	2	3	4
λ_2	0.139025	0.078194	0.054485
λ_3	0.222155	0.128085	0.104315
λ_5		0.160813	
λ_6	0.638820	0.036178	0.126755
λ_{10}			0.229816
λ_{11}			0.016484
λ_{12}		0.108828	
λ_{13}		0.487902	
λ_{14}			
λ_{16}			
λ_{27}			0.450302
λ_{28}			0.017842
rate	0.333364	0.333223	0.333218
σ_{GA}	1.1840	1.2415	1.2615
σ^*	1.1981	1.2607	1.2780
$(\frac{E_b}{N_0})^*(dB)$	0.190	-0.250	-0.371
S.L. (dB)	-0.4953	-0.4958	-0.4958

Table 2: Good degree sequences yielding codes of rate approximately $1/3$ for the AWGN channel and with $a = 2, 3, 4$. For each sequence the Gaussian approximation noise threshold, the actual sum-product decoding threshold, and the corresponding $(\frac{E_b}{N_0})^*$ in dB are given. Also listed is the Shannon limit (S.L.)

For example, consider the “ $a = 3$ ” column in Table 2. We adjust Gaussian approximation noise threshold

σ_{GA} to be 1.2415 to have the returned optimal sequence having rate 0.333223. Then applying the exact density evolution program on this code, we obtain the actual sum-product decoding threshold $\sigma^* = 1.2607$, which corresponds to $E_b/N_0 = -0.250$ dB. This should be compared to the Shannon limit for the ensemble of all linear codes of the same rate, which is -0.4958 dB. As we increase the parameter a , the ensemble improves. For $a = 4$, the best code we have found has iterative decoding threshold $E_b/N_0 = -0.371$ dB, which is only 0.12 dB above the Shannon limit.

The above analysis is for *bit* error probability. In order to have zero *word* error probability, it is necessary to have $\lambda_2 = 0$. (This can be proved by the following argument: if $\lambda_2 > 0$, then in the ensemble, as $n \rightarrow \infty$, the average number of weight 2 codewords is bounded away from zero. Hence even a maximum-likelihood decoder would have non-zero decoding error probability.) In Table 3, we compare the noise thresholds of codes with and without $\lambda_2 = 0$.

a	8	8
λ_2		0.0577128
λ_3	0.252744	0.117057
λ_7		0.2189922
λ_8		0.0333844
λ_{11}	0.081476	
λ_{12}	0.327162	
λ_{18}		0.2147221
λ_{20}		0.0752259
λ_{46}	0.184589	
λ_{48}	0.154029	
λ_{55}		0.0808676
λ_{58}		0.202038
rate	0.50227	0.497946
σ^*	0.9589	0.972
$(\frac{E_b}{N_0})^*(dB)$	0.344	0.266
Shannon limit	0.197	0.178

Table 3: Two degree sequences yielding codes of rate $\approx 1/2$ with $a = 8$. For each sequence, the actual sum-product decoding threshold, and the corresponding $(\frac{E_b}{N_0})^*$ in dB are given. Also listed is the Shannon limit.

We chose rate one-half because we wanted to compare our results with the best irregular LDPC codes obtained in [5]. Our best IRA code has threshold 0.266 dB, while the best rate one-half irregular LDPC code found in [5] has threshold 0.25 dB. These two codes have roughly the same decoding complexity, but unlike LDPC codes, IRA codes have a simple linear encoding algorithm.

4.3. Simulation Results

We simulated the rate one-half code with $\lambda_2 = 0$ in Table 3. Figure 2 shows the performance of that particular code, with information block lengths 10^3 , 10^4 , and 10^5 . For comparison, we also show the performance of the best known rate 1/2 turbo code for the same block length.

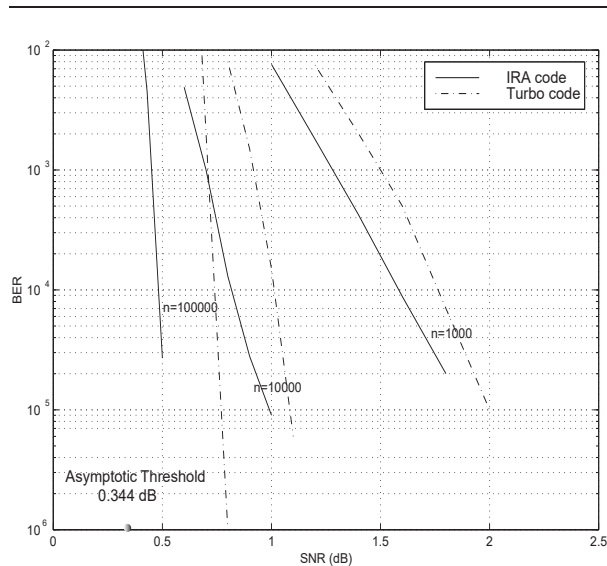


Figure 2: Comparison between turbo codes (dashed curves) and IRA codes (solid curves) of lengths $n = 10^3, 10^4, 10^5$. All codes are of rate one-half.

5. CONCLUSIONS

We have introduced a class of codes, the IRA codes, that combines many of the favorable attributes of turbo codes and LDPC codes. Like turbo codes (and unlike LDPC codes), they can be encoded in linear time. Like LDPC codes (and unlike turbo codes), they are amenable to an exact Richardson-Urbanke style analysis. In simulated performance they appear to be slightly superior to turbo codes of comparable complexity, and just as good as the best known irregular LDPC codes. In our opinion, the important open problem is to prove (or disprove) that IRA codes can be decoded reliably in linear time at rates arbitrarily close to channel capacity. We know this to be true for the binary erasure channel, but for no other channel model. If this should turn out to be true, we would argue that IRA codes definitively solve the problem posed implicitly by Shannon in 1948. If it is not true, then researchers should search for an even better class of code ensembles.

REFERENCES

- [1] D. Divsalar, H. Jin, and R. J. McEliece, “Coding theorems for ‘turbo-like’ codes,” pp. 201-210 in *Proc. 36th Allerton Conf. on Communication, Control, and Computing*. (Allerton, Illinois, Sept. 1998).
- [2] T. J. Richardson and R. Urbanke, “The capacity of low-density parity-check codes under message passing decoding,” submitted to *IEEE Trans. Inform. Theory*.
- [3] M. Luby, M. Mitzenmacher, A. Shokrollahi, D. Spielman, and V. Stemann, “Practical loss-resilient codes,” *Proc. 29th ACM Symp. on the Theory of Computing* (1997), pp. 150-159.
- [4] M. Luby, M. Mitzenmacher, A. Shokrollahi, and D. Spielman, “Analysis of low-density codes and improved designs using irregular graphs,” *Proc. 30th ACM Symp. on the Theory of Computing* (1998), pp. 249-258.
- [5] T. J. Richardson, A. Shokrollahi, and R. Urbanke, “Design of provably good low-density parity-check codes,” submitted to *IEEE Trans. Inform. Theory*.
- [6] S.-Y. Chung, R. Urbanke, and T. J. Richardson, “Analysis of sum-product decoding of low-density parity-check codes using a Gaussian approximation,” submitted to *IEEE Trans. Inform. Theory*.
- [7] D. Divsalar, S. Dolinar, and F. Pollara, “Iterative turbo decoder analysis based on Gaussian density evolution,” submitted to *IEEE J. Selected Areas in Comm.*
- [8] M. A. Shokrollahi, “New sequences of linear time erasure codes approaching channel capacity,” *Proc. 1999 ISITA* (Honolulu, Hawaii, November 1999) pp. 65–76.
- [9] N. Wiberg, “Codes and decoding on general graphs,” dissertation no. 440, Linköping Studies in Science and Technology, Linköping, Sweden, 1996.
- [10] J. Hagenauer, E. Offer, and L. Papke, “Iterative decoding of binary block and convolutional codes,” *IEEE Trans. Inform. Theory*, vol. IT-42, no. 2 (March 1996). pp. 429–445.

EXHIBIT F

Design Methods for Irregular Repeat–Accumulate Codes

Aline Roumy, *Member, IEEE*, Souad Guemghar, *Student Member, IEEE*, Giuseppe Caire, *Senior Member, IEEE*, and Sergio Verdú, *Fellow, IEEE*

Abstract—We optimize the random-like ensemble of irregular repeat–accumulate (IRA) codes for binary-input symmetric channels in the large block-length limit. Our optimization technique is based on approximating the evolution of the densities (DE) of the messages exchanged by the belief-propagation (BP) message-passing decoder by a one-dimensional dynamical system. In this way, the code ensemble optimization can be solved by linear programming. We propose four such DE approximation methods, and compare the performance of the obtained code ensembles over the binary-symmetric channel (BSC) and the binary-antipodal input additive white Gaussian noise channel (BIAWGNC). Our results clearly identify the best among the proposed methods and show that the IRA codes obtained by these methods are competitive with respect to the best known irregular low-density parity-check (LDPC) codes. In view of this and the very simple encoding structure of IRA codes, they emerge as attractive design choices.

Index Terms—Belief propagation (BP), channel capacity, density evolution, low-density parity-check (LDPC) codes, stability, threshold, turbo codes.

I. INTRODUCTION

SINCE the discovery of turbo codes [1], there have been several notable inventions in the field of random-like codes. In particular, the rediscovery of the low-density parity-check (LDPC) codes, originally proposed in [2], the introduction of irregular LDPCs [3], [4], and the introduction of the repeat–accumulate (RA) codes [5].

In [3], [4], irregular LDPCs were shown to asymptotically achieve the capacity of the binary erasure channel (BEC) under iterative message-passing decoding. Although the BEC is the only channel for which such a result currently exists, irregular LDPC codes have been designed for other binary-input channels (e.g., the binary-symmetric channel (BSC), the binary-antipodal input additive white Gaussian noise channel (BIAWGNC) [6], and the binary-input intersymbol interference (ISI) channel [7]–[9]) and have shown to achieve very good performance.

First attempts to optimize irregular LDPC codes ([10] for the BEC and other channels [11]) with the density evolution (DE) technique computes the expected performance for a random-like

code ensemble in the limit of infinite code block length. In order to reduce the computational burden of ensemble optimization based on the DE, faster techniques have been proposed, based on the approximation of the DE by a one-dimensional dynamical system (recursion). These techniques are exact only for the BEC (for which DE is one-dimensional). The most popular techniques proposed so far are based on the Gaussian approximation (GA) of messages exchanged in the message-passing decoder. GA in addition to the symmetry condition of message densities implies that the Gaussian density of messages is expressed by a single parameter. Techniques differ in the parameter to be tracked and in the mapping functions defining the dynamical system [12]–[18].

The introduction of irregular LDPCs motivated other schemes such as irregular RA (IRA) [19], for which similar results exist (achievability of the BEC capacity) and irregular turbo codes [20]. IRA codes are, in fact, special subclasses of both irregular LDPCs and irregular turbo codes. In IRA codes, a fraction f_i of information bits is repeated i times, for $i = 2, 3, \dots$. The distribution

$$\left\{ f_i \geq 0, i = 2, 3, \dots : \sum_{i=2}^{\infty} f_i = 1 \right\}$$

is referred to as the *repetition profile*, and it is kept as a degree of freedom in the optimization of the IRA ensemble. After the repetition stage, the resulting sequence is interleaved and input to a recursive finite-state machine (called accumulator) which outputs one bit for every a input symbols, where a is referred to as *grouping factor* and is also a design parameter.

IRA codes are an appealing choice because the encoder is extremely simple, their performance is quite competitive with that of turbo codes and LDPCs, and they can be decoded with a very-low-complexity iterative decoding scheme.

The only other work that has proposed a method to design IRA codes is [19], [21] where the design focuses on the choice of the grouping factor and the repetition profile. The recursive finite-state machine is the simplest one which gives full freedom to choose any rational number between 0 and 1 as the coding rate. We will also restrict our study to IRAs that use the same simple recursion of [19], although it might be expected that better codes can be obtained by including the finite-state machine as a degree of freedom in the overall ensemble optimization. The method used in [19] to choose the repetition profile was based on the infinite-block-length GA of message-passing decoding proposed in [14]. In this work, we propose and compare four low-complexity ensemble

Manuscript received October 22, 2002; revised April 1, 2004.

A. Roumy is with IRISA-INRIA, 35042 Rennes, France (e-mail: aline.roumy@irisa.fr).

S. Guemghar and G. Caire are with the Eurecom Institute, 06904 Sophia-Antipolis, France (e-mail: Souad.Guemghar@eurecom.fr; Giuseppe.Caire@eurecom.fr).

S. Verdú is with the Department of Electrical Engineering, Princeton University, Princeton, NJ 08544 USA (e-mail: verdu@princeton.edu).

Communicated by R. Urbanke, Associate Editor for Coding Techniques.

Digital Object Identifier 10.1109/TIT.2004.831778

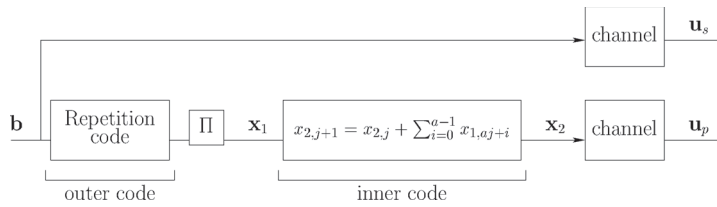


Fig. 1. IRA encoder.

optimization methods. Our approach to design IRAs is based on several tools that have been noticed recently: the EXtrinsic mutual Information Transfer (EXIT) function and its analytical properties [12], [22], [23], reciprocal channel (duality) approximation [22], [24], and the nonstrict convexity of mutual information.

The rest of the paper is organized as follows. Section II presents the systematic IRA encoder and its related decoder: the belief-propagation (BP) message-passing algorithm. Existing results on the analysis of the decoder (i.e., DE technique) are summarized and applied to the IRA code ensemble. This leads to a two-dimensional dynamical system whose state is defined on the space of symmetric distributions, for which we derive a local stability condition. In Section III, we propose a general framework in order to approximate the DE (defined on the space of distributions) by a standard dynamical system defined on the reals. We propose four low-complexity ensemble optimization methods as special cases of our general framework. These methods differ by the way the message densities and the BP transformations are approximated:

- 1) GA, with reciprocal channel (duality) approximation;
- 2) BEC approximation, with reciprocal channel approximation;
- 3) GA, with EXIT function of the inner decoder;
- 4) BEC approximation, with EXIT function of the inner decoder.

All four methods lead to optimization problems solvable by linear programming. In Section IV, we show that the first proposed method yields a one-dimensional DE approximation with the same stability condition as the exact DE, whereas the exact stability condition must be added to the ensemble optimization as an explicit additional constraint for the second method. Then, we show that, in general, the GA methods are optimistic, in the sense that there is no guarantee that the optimized rate is below capacity. On the contrary, we show that for the BEC approximation methods rates below capacity are guaranteed. In Section V, we compare our code optimization methods by evaluating their iterative decoding threshold (evaluated by the exact DE) over the BIAWGNC and the BSC.

II. ENCODING, DECODING, AND DENSITY EVOLUTION

Fig. 1 shows the block diagram of a systematic IRA encoder. A block of information bits $\mathbf{b} = (b_1, \dots, b_k) \in \mathbb{F}_2^k$ is encoded by an (irregular) repetition code of rate k/n . Each bit b_j is repeated r_j times, where (r_1, \dots, r_k) is a sequence of integers such that $2 \leq r_j \leq d$ and $\sum_{j=1}^k r_j = n$ (d is the maximum repetition factor). The block of repeated symbols is interleaved,

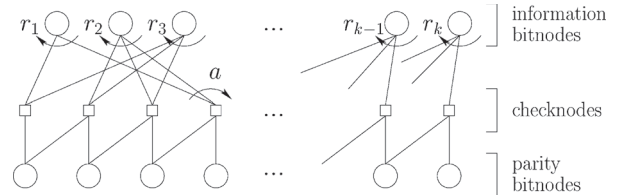


Fig. 2. Tanner graph of an IRA code.

and the resulting block $\mathbf{x}_1 = (x_{1,1}, \dots, x_{1,n}) \in \mathbb{F}_2^n$ is encoded by an *accumulator*, defined by the recursion

$$x_{2,j+1} = x_{2,j} + \sum_{i=0}^{a-1} x_{1,aj+i}, \quad j = 0, \dots, m-1 \quad (1)$$

with initial condition $x_{2,0} = 0$, where $\mathbf{x}_2 = (x_{2,1}, \dots, x_{2,m}) \in \mathbb{F}_2^m$ is the accumulator output block corresponding to the input \mathbf{x}_1 , $a \geq 1$ is a given integer (referred to as *grouping factor*), and we assume that $m = n/a$ is an integer. Finally, the codeword corresponding to the information block \mathbf{b} is given by $\mathbf{x} = (\mathbf{b}, \mathbf{x}_2)$.

The transmission channel is memoryless, binary-input, and symmetric-output, i.e., its transition probability $p_{Y|X}(y|x)$ satisfies

$$p_{Y|X}(y|0) = p_{Y|X}(-y|1) \quad (2)$$

where $y \mapsto -y$ indicates a *reflection* of the output alphabet.¹

IRA codes are best represented by their Tanner graph [25] (see Fig. 2). In general, the Tanner graph of a linear code is a bipartite graph whose node set is partitioned into two subsets: the *bitnodes*, corresponding to the coded symbols, and the *checknodes*, corresponding to the parity-check equations that codewords must satisfy. The graph has an edge between bitnode α and checknode β if the symbol corresponding to α participates in the parity-check equation corresponding to β .

Since the IRA encoder is systematic (see Fig. 1), it is useful to further classify the bitnodes into two subclasses: the information bitnodes, corresponding to information bits, and the parity bitnodes, corresponding to the symbols output by the accumulator. Those information bits that are repeated i times are represented by bitnodes with degree i , as they participate in i parity-check equations. Each checknode is connected to a information bit nodes and to two parity bitnodes and represents one of the equations (for a particular j) (1). The connections between checknodes and information bitnodes are determined by the interleaver and are highly randomized. On the contrary, the connections between checknodes and parity bitnodes are arranged in a regular

¹If the output alphabet is the real line, then $-y$ coincides with ordinary reflection with respect to the origin. Generalizations to other alphabets are immediate.

zig-zag pattern since, according to (1), every pair of consecutive parity bits are involved in one parity-check equation.

A random IRA code ensemble with parameters $(\{\lambda_i\}, a)$ and (information) block length k is formed by all graphs of the form of Fig. 2 with k information bitnodes, grouping factor a , and $\lambda_i n$ edges connected to information bitnodes of degree i , for $i = 2, \dots, d$. The sequence of nonnegative coefficients $\{\lambda_i\}$ such that $\sum_{i=2}^d \lambda_i = 1$ is referred to as the *degree distribution* of the ensemble. The probability distribution over the code ensemble is induced by the uniform probability over all interleavers (permutations) of n elements.

The information bitnodes average degree is given by $\bar{d} \triangleq 1/(\sum_{i=2}^d \lambda_i/i)$. The number of edges connecting information bitnodes to checknodes is $n = k/(\sum_{i=2}^d \lambda_i/i)$. The number of parity bitnodes is $m = k/(a \sum_{i=2}^d \lambda_i/i)$. Finally, the code rate is given by

$$R = \frac{k}{k+m} = \frac{a \sum_{i=2}^d \lambda_i/i}{1 + a \sum_{i=2}^d \lambda_i/i} = \frac{a}{a + \bar{d}}. \quad (3)$$

Under the constraints $0 \leq \lambda_i \leq 1$ and $\sum_{i \geq 2} \lambda_i = 1$, we get $\bar{d} \geq 2$. Therefore, the highest rate with parameter a set to 1 is $1/3$. This motivates the use of $a \geq 2$ in order to get higher rates.

A. Belief Propagation Decoding of IRA Codes

In this work, we consider BP message-passing decoding [26]–[28]. In message-passing decoding algorithms, the graph nodes receive messages from their neighbors, compute new messages, and forward them to their neighbors. The algorithm is defined by the code Tanner graph, by the set on which messages take on values, by the node computation rules, and by the node activation scheduling.

In BP decoding, messages take on values in the extended real line $\mathbb{R} \cup \{-\infty, \infty\}$. The BP decoder is initialized by setting all messages output by the checknodes equal to zero. Each bitnode α is associated with the *channel observation* message (log-likelihood ratio)

$$u_\alpha = \log \frac{p_{Y|X}(y_\alpha | x_\alpha = 0)}{p_{Y|X}(y_\alpha | x_\alpha = 1)} \quad (4)$$

where y_α is the channel output corresponding to the transmission of the code symbol x_α .

The BP node computation rules are given as follows. For a given node, we identify an adjacent edge as *outgoing* and all other adjacent edges as *incoming*. Consider a bitnode α of degree i and let m_1, \dots, m_{i-1} denote the messages received from the $i-1$ incoming edges and u_α the associated channel observation message. The message $m_{o,\alpha}$ passed along the outgoing edge is given by

$$m_{o,\alpha} = m_1 + \dots + m_{i-1} + u_\alpha. \quad (5)$$

Consider a checknode β of degree i and let m_1, \dots, m_{i-1} denote the messages received from the $i-1$ incoming edges. The message $m_{o,\beta}$ passed along the outgoing edge is given by

$$m_{o,\beta} = \gamma^{-1}(\gamma(m_1) + \dots + \gamma(m_{i-1})) \quad (6)$$

where the mapping $\gamma: \mathbb{R} \rightarrow \mathbb{F}_2 \times \mathbb{R}_+$ is defined by [11]

$$\gamma(z) = \left(\text{sign}(z), -\log \tanh \frac{|z|}{2} \right) \quad (7)$$

and where the sign function is defined as [11]

$$\text{sign}(z) = \begin{cases} 0, & \text{if } z > 0 \\ 0, & \text{with probability } 1/2 \text{ if } z = 0 \\ 1, & \text{with probability } 1/2 \text{ if } z = 0 \\ 1, & \text{if } z < 0. \end{cases}$$

Since the code Tanner graph has cycles, different schedulings yield in general nonequivalent BP algorithms. In this work, we shall consider the following ‘‘classical’’ schedulings.

- LDPC-like scheduling [19]. In this case, all bitnodes and all checknodes are activated alternately and in parallel. Every time a node is activated, it sends outgoing messages to all its neighbors. A decoding iteration (or ‘‘round’’ [31]) consists of the activation of all bitnodes and all checknodes.
- Turbo-like scheduling. Following [29], a good decoding scheduling consists of isolating large trellis-like subgraphs (or, more generally, normal realizations in Forney’s terminology) and applying locally the forward-backward Bahl-Cocke-Jelinek-Raviv (BCJR) algorithm [30] (that implements efficiently the BP algorithm on normal cycle-free graphs), as done for turbo codes [1]. A decoding iteration consists of activating all the information bitnodes in parallel (according to (5)) and of running the BCJR algorithm over the entire accumulator trellis. In particular, the checknodes do not send messages to the information bitnodes until the BCJR iteration is completed.

Notice that for both of the above schedulings one decoder iteration corresponds to the activation of all information bitnodes in the graph exactly once.

B. Density Evolution and Stability

The bit-error rate (BER) performance of BP decoding averaged over the IRA code ensemble and over the noise observations can be analyzed, for any finite number ℓ of iterations and in the limit of $k \rightarrow \infty$, by the DE technique [11]. The usefulness of the DE method stems from the *Concentration Theorem* [31], [10] which guarantees that, with high probability, the BER after ℓ iterations of the BP decoder applied to a randomly selected code in the ensemble and to a randomly generated channel noise sequence is close to the BER computed by DE, for sufficiently large block length.

Next, we formulate the DE for IRA codes and we study the stability condition of the fixed-point corresponding to zero BER. As in [11, Sec. III-B], we introduce the space of *distributions* whose elements are nonnegative nondecreasing right-continuous functions with range in $[0, 1]$ and domain the extended real line.

It can be shown that, for a binary-input symmetric-output channel, the distributions of messages at any iteration of the DE satisfy the symmetry condition

$$\int h(x)dF(x) = \int e^{-x}h(-x)dF(x) \quad (8)$$

for any function h for which the integral exists. If F has density f , (8) is equivalent to

$$f(x) = e^x f(-x). \quad (9)$$

With some abuse of terminology, distributions satisfying (8) are said to be *symmetric*. The space of symmetric distributions will be denoted by \mathcal{F}_{sym} .

The BER operator $\text{Pe}: \mathcal{F}_{\text{sym}} \rightarrow [0, 1/2]$ is defined by

$$\text{Pe}(F) = \frac{1}{2}(F^-(0) + F(0))$$

where $F^-(z)$ is the left-continuous version of $F(z)$. We introduce the ‘‘delta at zero’’ distribution, denoted by Δ_0 , for which $\text{Pe}(\Delta_0) = 1/2$, and the ‘‘delta at infinity’’ distribution, denoted by Δ_∞ , for which $\text{Pe}(\Delta_\infty) = 0$.

The symmetry property (8) implies that a sequence of symmetric distributions $\{F^{(\ell)}\}_{\ell=0}^\infty$ converges to Δ_∞ if and only if $\lim_{\ell \rightarrow \infty} \text{Pe}(F^{(\ell)}) = 0$, where convergence of distributions is in the sense given in [11, Sec. III-F].

The DE for IRA code ensembles is given by the following proposition whose derivation is omitted as it is completely analogous to the derivation of DE in [11] for irregular LDPC codes.

Proposition 1: Let P_ℓ (respectively, \tilde{P}_ℓ) denote the average distribution of messages passed from an information bitnode (respectively, parity bitnode) to a checknode, at iteration ℓ . Let Q_ℓ (respectively, \tilde{Q}_ℓ) denote the average distribution of messages passed from a checknode to an information bitnode (respectively, parity bitnode), at iteration ℓ .

Under the cycle-free condition, $P_\ell, \tilde{P}_\ell, Q_\ell, \tilde{Q}_\ell$ satisfy the following recursion:

$$P_\ell = F_u \otimes \lambda(Q_\ell) \quad (10)$$

$$\tilde{P}_\ell = F_u \otimes \tilde{Q}_\ell \quad (11)$$

$$Q_\ell = \Gamma^{-1} \left(\Gamma(\tilde{P}_{\ell-1})^{\otimes 2} \otimes \Gamma(P_{\ell-1})^{\otimes (a-1)} \right) \quad (12)$$

$$\tilde{Q}_\ell = \Gamma^{-1} \left(\Gamma(\tilde{P}_{\ell-1}) \otimes \Gamma(P_{\ell-1})^{\otimes a} \right) \quad (13)$$

for $\ell = 1, 2, \dots$, with initial condition $P_0 = \tilde{P}_0 = \Delta_0$, where F_u denotes the distribution of the channel observation messages (4), \otimes denotes convolution of distributions, defined by

$$(F \otimes G)(z) = \int F(z-t)dG(t) \quad (14)$$

where \otimes^m denotes m -fold convolution,

$$\lambda(F) \triangleq \sum_{i=2}^d \lambda_i F^{\otimes (i-1)},$$

$\Gamma(F_x)$ is the distribution of $y = \gamma(x)$ (defined on $\mathbb{F}_2 \times \mathbb{R}_2$), when $x \sim F_x$, and Γ^{-1} denotes the inverse mapping of Γ , i.e., $\Gamma^{-1}(G_y)$ is the distribution of $x = \gamma^{-1}(y)$ when $y \sim G_y$. \square

The DE recursion (10)–(13) is a two-dimensional nonlinear dynamical system with state space $\mathcal{F}_{\text{sym}}^2$ (i.e., the state trajecto-

ries of (10)–(13) are sequences of pairs of symmetric distributions (P_ℓ, \tilde{P}_ℓ)). For this system, the BER at iteration ℓ is given by $\text{Pe}(P_\ell)$.

It is easy to see that $(\Delta_\infty, \Delta_\infty)$ is a fixed point of (10)–(13). The local stability of this fixed point is given by the following result.

Theorem 1: The fixed point $(\Delta_\infty, \Delta_\infty)$ for the DE is locally stable if and only if

$$\lambda_2 < \frac{e^r(e^r - 1)}{a + 1 + e^r(a - 1)} \quad (15)$$

where $r = -\log(\int e^{-z/2} dF_u(z))$.

Proof: See Appendix I. \square

Here necessity and sufficiency are used in the sense of [11]. By following steps analogous to [11], it can be shown that if (15) holds, then there exists $\xi > 0$ such that if for some $\ell \in \mathbb{N}$

$$\text{Pe}(RP_\ell(P_0, \tilde{P}_0) + (1 - R)\tilde{P}_\ell(P_0, \tilde{P}_0)) < \xi$$

then $\text{Pe}(RP_\ell + (1 - R)\tilde{P}_\ell)$ converges to zero as ℓ tends to infinity. On the contrary, if λ_2 is strictly larger than the right-hand side (RHS) of (15), then there exists $\xi > 0$ such that for all $\ell \in \mathbb{N}$

$$\text{Pe}(RP_\ell(P_0, \tilde{P}_0) + (1 - R)\tilde{P}_\ell(P_0, \tilde{P}_0)) > \xi.$$

III. IRA ENSEMBLE OPTIMIZATION

In this section, we tackle the problem of optimizing the IRA code ensemble parameters for a broad class of binary-input symmetric-output channels.

A property of DE given in Proposition 1 is that $\text{Pe}(P_\ell)$ for $\ell = 1, 2, \dots$ is a nonincreasing nonnegative sequence. Hence, the limit $\lim_{\ell \rightarrow \infty} \text{Pe}(P_\ell)$ exists. Consider a family of channels

$$\mathcal{C}(\nu) = \{p_{Y|X}^\nu : \nu \in \mathbb{R}_+\}$$

where the channel parameter ν is, for example, an indicator of the noise level in the channel. Following [31], we say that $\mathcal{C}(\nu)$ is monotone with respect to the IRA code ensemble $(\{\lambda_i\}, a)$ under BP decoding if, for any finite ℓ

$$\nu \leq \nu' \Leftrightarrow \text{Pe}(P_\ell) \leq \text{Pe}(P'_\ell)$$

where P_ℓ and P'_ℓ are the message distributions at iteration ℓ of DE applied to channels $p_{Y|X}^\nu$ and $p_{Y|X}^{\nu'}$, respectively.

Let $\text{BER}(\nu) = \lim_{\ell \rightarrow \infty} \text{Pe}(P_\ell)$, where $\{P_\ell\}$ is the trajectory of DE applied to the channel $p_{Y|X}^\nu$. The *threshold* ν^* of the ensemble $(\{\lambda_i\}, a)$ over the monotone family $\mathcal{C}(\nu)$ is the worst case channel parameter for which the limiting BER is zero, i.e.,

$$\nu^* = \sup\{\nu \geq 0 : \text{BER}(\nu) = 0\}. \quad (16)$$

Thus, for every value of ν , the optimal IRA ensemble parameters a and $\{\lambda_i\}$ maximize R subject to vanishing $\text{BER}(\nu) = 0$, i.e., are solution of the optimization problem

$$\begin{cases} \text{maximize} & a \sum_{i=2}^d \lambda_i / i \\ \text{subject to} & \sum_{i=2}^d \lambda_i = 1, \lambda_i \geq 0 \quad \forall i \\ \text{and to} & \text{BER}(\nu) = 0 \end{cases} \quad (17)$$

the solution of which can be found by some numerical techniques, as in [11]. However, the constraint $\text{BER}(\nu) = 0$ is given directly in terms of the fixed point of the DE recursion, and makes optimization very computationally intensive.

A variety of methods have been developed in order to simplify the code ensemble optimization [19], [24], [14], [32]. They consist of replacing the DE with a dynamical system defined over the reals (rather than over the space of distributions), whose trajectories and fixed points are related in some way to the trajectories and the fixed point of the DE. Essentially, all proposed approximated DE methods can be formalized as follows. Let $\Phi: \mathcal{F}_{\text{sym}} \rightarrow \mathbb{R}$ and $\Psi: \mathbb{R} \rightarrow \mathcal{F}_{\text{sym}}$ be mappings of the set of symmetric distributions to the real numbers and *vice versa*. Then, a dynamical system with state space \mathbb{R}^2 can be derived from (10)–(13) as

$$x_\ell = \Phi(F_u \otimes \lambda(Q_\ell)) \quad (18)$$

$$\tilde{x}_\ell = \Phi(F_u \otimes \tilde{Q}_\ell) \quad (19)$$

$$Q_\ell = \Gamma^{-1} \left(\Gamma(\Psi(\tilde{x}_{\ell-1}))^{\otimes 2} \otimes \Gamma(\Psi(x_{\ell-1}))^{\otimes (a-1)} \right) \quad (20)$$

$$\tilde{Q}_\ell = \Gamma^{-1} \left(\Gamma(\Psi(\tilde{x}_{\ell-1})) \otimes \Gamma(\Psi(x_{\ell-1}))^{\otimes a} \right) \quad (21)$$

for $\ell = 1, 2, \dots$, with initial condition $x_0 = \tilde{x}_0 = \Phi(\Delta_0)$, and where (x_ℓ, \tilde{x}_ℓ) are the system state variables.

By eliminating the intermediate distributions Q_ℓ and \tilde{Q}_ℓ , we can put (18)–(21) in the form

$$\begin{aligned} x_\ell &= \phi(x_{\ell-1}, \tilde{x}_{\ell-1}) \\ \tilde{x}_\ell &= \tilde{\phi}(x_{\ell-1}, \tilde{x}_{\ell-1}). \end{aligned} \quad (22)$$

For all DE approximations considered in this work, the mappings Φ and Ψ and the functions ϕ and $\tilde{\phi}$ satisfy the following desirable properties.

- 1) $\Phi(\Delta_0) = 0$, $\Phi(\Delta_\infty) = 1$.
- 2) $\Psi(0) = \Delta_0$, $\Psi(1) = \Delta_\infty$.
- 3) ϕ and $\tilde{\phi}$ are defined on $[0, 1] \times [0, 1]$ and have range in $[0, 1]$.
- 4) $\phi(0, 0) > 0$ and $\tilde{\phi}(0, 0) > 0$.
- 5) $\phi(1, 1) = \tilde{\phi}(1, 1) = 1$, i.e., $(1, 1)$ is a fixed point of the recursion (22). Moreover, this fixed point corresponds to the zero-BER fixed point $(\Delta_\infty, \Delta_\infty)$ of the exact DE.
- 6) If $F_u \neq \Delta_0$, the function $\tilde{\phi}(x, \tilde{x}) - \tilde{x}$ is strictly decreasing in \tilde{x} for all $x \in [0, 1]$. Therefore, the equation

$$\tilde{x} = \tilde{\phi}(x, \tilde{x})$$

has a unique solution in $[0, 1]$ for all $x \in [0, 1]$. This solution will be denoted by $\tilde{x}(x)$.

It follows that all fixed points of (22) must satisfy

$$x = \phi(x, \tilde{x}(x)) \quad (23)$$

and that in order to avoid fixed points other than $(1, 1)$, (23) must not have solutions in the interval $[0, 1)$, i.e., it must satisfy

$$x < \phi(x, \tilde{x}(x)), \quad \forall x \in [0, 1). \quad (24)$$

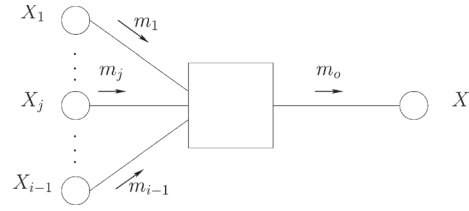


Fig. 3. EXIT model.

Notice that, in general, (24) is neither a necessary nor a sufficient condition for the uniqueness of the zero-BER fixed point of the exact DE. However, if the quality of the DE approximation is good, this provides a heuristic for the code ensemble optimization.

By replacing the constraint $\text{BER}(\nu) = 0$ by (24) in (17), we obtain the *approximated* IRA ensemble optimization method as

$$\begin{cases} \text{maximize} & a \sum_{i=2}^d \lambda_i / i \\ \text{subject to} & \sum_{i=2}^d \lambda_i = 1, \lambda_i \geq 0, \quad \forall i \\ \text{and to} & x < \phi(x, \tilde{x}(x)), \quad \forall x \in [0, 1). \end{cases} \quad (25)$$

Approximations of the DE recursion differ essentially in the choice of Φ and Ψ , and in the way the *intermediate* distributions Q_ℓ and \tilde{Q}_ℓ and the channel message distribution F_u are approximated. Next, we illustrate the approximation methods considered in this work.

A. EXIT Functions

Several recent works show that DE can be accurately described in terms of the evolution of the mutual information between the variables associated with the bitnodes and their messages (see [12], [33]–[35], [13], [23], [18]).

The key idea in order to approximate DE by mutual information evolution is to describe each computation node in BP decoding by a mutual information transfer function. For historical reasons, this function is usually referred to as the EXtrinsic mutual Information Transfer (EXIT) function.

EXIT functions are generally defined as follows. Consider the model of Fig. 3, where the box represents a generalized computation node of the BP algorithm (i.e., it might contain a subgraph formed by several nodes and edges, and might depend on some other random variables such as channel observations, not shown in Fig. 3). Let m_1, \dots, m_{i-1} denote the input messages, assumed independent and identically distributed (i.i.d.) $\sim F_{\text{in}}$, and let $m_o \sim F_{\text{out}}$ denote the output message. Let X_j denote the binary code symbol associated with message m_j , for $j = 1, \dots, i-1$, and let X denote the binary code symbol associated with message m_o . Since $F_{\text{in}}, F_{\text{out}} \in \mathcal{F}_{\text{sym}}$, we can think of m_j and m_o as the outputs of binary-input symmetric-output channels with inputs X_j and X and transition probabilities

$$P(m_j \leq z | X_j = 0) = F_{\text{in}}(z) \quad (26)$$

$$P(m_o \leq z | X = 0) = F_{\text{out}}(z) \quad (27)$$

respectively.

The channel (26) models the *a priori* information that the node receives about the symbols X_j 's, and the channel (27) models the *extrinsic information* [1] that the node generates about the symbol X .

We define the binary-input symmetric-output capacity functional $\mathcal{I}: \mathcal{F}_{\text{sym}} \rightarrow [0, 1]$, such that

$$\mathcal{I}(F) = 1 - \int_{-\infty}^{\infty} \log_2(1 + e^{-z}) dF(z). \quad (28)$$

Namely, \mathcal{I} maps any symmetric distribution F into the capacity² of the binary-input symmetric-output channel with transition probability $p_{Y|X}(y|0) = F(y)$.

Then, we let

$$\begin{aligned} I_A &= I(X_j; m_j) = \mathcal{I}(F_{\text{in}}) \\ I_E &= I(X; m_o) = \mathcal{I}(F_{\text{out}}) \end{aligned}$$

denote the capacities of the channels (26) and (27), respectively. The EXIT function of the node of Fig. 3 is the set of pairs (I_A, I_E) , for all $I_A \in [0, 1]$ and for some (arbitrary) choice of the input distribution F_{in} such that $\mathcal{I}(F_{\text{in}}) = I_A$. Notice that the EXIT function of a node is not uniquely defined, since it depends on the choice of F_{in} . In general, different choices yield different transfer functions.

The approximations of the DE considered in this work are based on EXIT functions, and track the evolution of the mutual information between the messages output by the bitnodes and the associated code symbols.

Remark. Two properties of binary-input symmetric-output channels: Before concluding this section, we take a brief detour in order to point out two properties of binary-input symmetric-output channels. Consider a binary-input symmetric-output channel with $p_{Y|X}(y|0) = G(y)$, where G is not necessarily symmetric (in the sense of (8)). Its capacity can be written as

$$C = 1 - \int_{-\infty}^{\infty} \log_2 \left(1 - \frac{dG(-z)}{dG(z)} \right) dG(z). \quad (29)$$

By concatenating the transformation $y \mapsto u = \log \frac{p_{Y|X}(y|0)}{p_{Y|X}(y|1)}$ to the channel output, we obtain a new binary-input symmetric-output channel with $p'_{U|X}(u|0) = F(u)$ such that $F \in \mathcal{F}_{\text{sym}}$. Moreover, since U is a sufficient statistic for Y , the original channel has the same capacity as the new channel, given by $C = \mathcal{I}(F)$. Therefore, by defining appropriately the channel output, the capacity of any binary-input symmetric-output channel can always be put in the form (28).

Another interesting property is the following.

Proposition 2: The mutual information functional is not strictly convex on the set of binary-input symmetric-output channels with transition probability $p_{Y|X}(y|0) \in \mathcal{F}_{\text{sym}}$.

Proof: See Appendix II. \square

B. Method 1

The first approximation of the DE considered in this work assumes that the distributions at any iteration are Gaussian. A Gaussian distribution satisfies the symmetry condition (9) if and only if its variance is equal to twice the absolute value of its mean. We introduce the shorthand notation $\mathcal{N}_{\text{sym}}(\mu)$ to denote the symmetric Gaussian distribution (or density, depending on the context) with mean μ , i.e., $\mathcal{N}_{\text{sym}}(\mu) \triangleq \mathcal{N}(\mu, 2|\mu|)$.

²Recall that the capacity of a binary-input symmetric-output memoryless channel is achieved by uniform i.i.d. inputs.

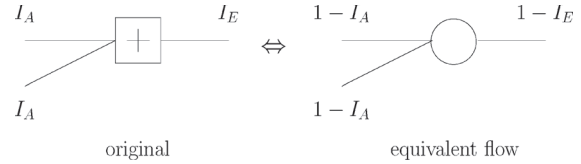


Fig. 4. Reciprocal channel approximation.

For a distribution $F \in \mathcal{F}_{\text{sym}}$, we let the mapping Φ be equal to \mathcal{I} defined in (28), and for all $x \in [0, 1]$ we define the mapping

$$\Psi: x \mapsto \mathcal{N}_{\text{sym}}(J^{-1}(x)) \quad (30)$$

where

$$\begin{aligned} J(\mu) &\triangleq \mathcal{I}(\mathcal{N}_{\text{sym}}(\mu)) \\ &= 1 - \frac{1}{\sqrt{\pi}} \int_{-\infty}^{+\infty} e^{-z^2} \log_2(1 + e^{-2\sqrt{\mu}z - \mu}) dz \end{aligned} \quad (31)$$

Namely, Ψ maps $x \in [0, 1]$ into the symmetric Gaussian distribution $\mathcal{N}_{\text{sym}}(\mu)$ such that the BIAWGNC with transition probability $p_{Y|X}(y|0) = \mathcal{N}_{\text{sym}}(\mu)$ has capacity x .

The first key approximation in Method 1 is

$$\begin{aligned} Q_\ell &\approx \mathcal{N}_{\text{sym}}(\mu_\ell) \\ \tilde{Q}_\ell &\approx \mathcal{N}_{\text{sym}}(\tilde{\mu}_\ell) \end{aligned} \quad (32)$$

for some $\mu_\ell, \tilde{\mu}_\ell \geq 0$.

In order to compute μ_ℓ and $\tilde{\mu}_\ell$, we make use of the reciprocal channel approximation [24] also called *approximate duality* property of EXIT functions in [22]. This states that the EXIT function of a checknode is accurately approximated by the EXIT function of a bitnode with the same degree after the change of variables $I_A \mapsto 1 - I_A$ and $I_E \mapsto 1 - I_E$ (see Fig. 4). Using approximate duality, we replace the checknode by a bitnode and change $(x_{\ell-1}, \tilde{x}_{\ell-1})$ into $(1 - x_{\ell-1}, 1 - \tilde{x}_{\ell-1})$. Since for a bitnode the output message is the sum of the input messages (see (5)), and since the input distributions $\Psi(1 - x_{\ell-1})$ and $\Psi(1 - \tilde{x}_{\ell-1})$ are Gaussian, also the output distribution is Gaussian, with mean

$$(a - 1)J^{-1}(1 - x_{\ell-1}) + 2J^{-1}(1 - \tilde{x}_{\ell-1})$$

for messages sent to information bitnodes and

$$aJ^{-1}(1 - x_{\ell-1}) + J^{-1}(1 - \tilde{x}_{\ell-1})$$

for messages sent to parity bitnodes. Finally, μ_ℓ and $\tilde{\mu}_\ell$ are given by

$$\begin{aligned} \mu_\ell &= J^{-1} \left(1 - J \left((a - 1)J^{-1}(1 - x_{\ell-1}) + 2J^{-1}(1 - \tilde{x}_{\ell-1}) \right) \right) \\ \tilde{\mu}_\ell &= J^{-1} \left(1 - J \left(aJ^{-1}(1 - x_{\ell-1}) + J^{-1}(1 - \tilde{x}_{\ell-1}) \right) \right). \end{aligned} \quad (33)$$

The second key approximation in Method 1 is to replace F_u with a discrete (symmetric) distribution such that

$$F_u \approx \sum_{j=1}^D p_j \Delta_{v_j} \quad (34)$$

for some integer $D \geq 2$, $v_j \in \mathbb{R}$, and $p_j \in \mathbb{R}_+$ such that $\sum_{j=1}^D p_j = 1$.

With this assumption, from the definition (28) of the operator \mathcal{I} and since [11]: a) the convolution of symmetric distributions is symmetric, and b) the convex combination of symmetric dis-

tributions is symmetric; it is immediate to write (18) and (19) as (35) at the bottom of the page. The desired DE approximation in the form (22) is obtained (implicitly) by combining (33) and (35). Notice that (35) is linear in the repetition profile and the optimization problem (25) can be solved as linear programming.

Example 1. Discrete-output channels: In general, when the channel output is discrete then the approximation (34) holds exactly. For example, for the BSC with transition probability p we have

$$F_u = p\Delta_{-\log \frac{1-p}{p}} + (1-p)\Delta_{\log \frac{1-p}{p}}. \quad \diamond$$

Example 2: The BIAWGNC defined by $y = (-1)^x + z$, where $z \sim \mathcal{N}(0, \sigma^2)$, is a channel such that

$$F_u = \mathcal{N}_{\text{sym}}(2/\sigma^2). \quad (36)$$

In this case, since convolving symmetric Gaussian distributions yields a symmetric Gaussian distribution whose mean is the sum of the means, the discretization approximation (34) is not necessary and we have

$$\begin{aligned} F_u \otimes \lambda(Q_\ell) &= \sum_{i=2}^d \lambda_i \mathcal{N}_{\text{sym}}(2/\sigma^2 + (i-1)\mu_\ell) \\ F_u \otimes \tilde{Q}_\ell &= \mathcal{N}_{\text{sym}}(2/\sigma^2 + \tilde{\mu}_\ell). \end{aligned} \quad (37)$$

By applying the operator \mathcal{I} and using (31) we obtain the DE approximation for the BIAWGNC as (38) at the bottom of the page. \diamond

C. Method 2

The second approximation of the DE considered in this work assumes that the distributions of messages at any iteration consist of two mass points, one at zero and the other at $+\infty$. For such distributions, we introduce the shorthand notation $\mathcal{E}_{\text{sym}}(\epsilon) \triangleq \epsilon\Delta_0 + (1-\epsilon)\Delta_\infty$.

We let the mapping Φ be equal to \mathcal{I} defined in (28) and the mapping Ψ be

$$\Psi: x \mapsto \mathcal{E}_{\text{sym}}(1-x) \quad (39)$$

for all $x \in [0, 1]$.

With these mappings, (20) and (21) can be put in the form

$$\begin{aligned} Q_\ell &= \mathcal{E}_{\text{sym}}(1 - x_{\ell-1}^{a-1} \tilde{x}_{\ell-1}^2) \\ \tilde{Q}_\ell &= \mathcal{E}_{\text{sym}}(1 - x_{\ell-1}^a \tilde{x}_{\ell-1}) \end{aligned} \quad (40)$$

where we used the fact that, as it can be easily seen from the definitions of Γ and Γ^{-1} in (46)–(48)

$$\begin{aligned} \Gamma^{-1}(\Gamma(\mathcal{E}_{\text{sym}}(\epsilon_1)) \otimes \Gamma(\mathcal{E}_{\text{sym}}(\epsilon_2))) \\ = \mathcal{E}_{\text{sym}}(1 - (1 - \epsilon_1)(1 - \epsilon_2)). \end{aligned}$$

Notice that, while in Method 1 we assumed Q_ℓ and \tilde{Q}_ℓ to be symmetric Gaussian (see (32)), here (40) holds exactly.

As a consequence of these mappings, the communication channel of the parity bits, with distribution F_u , is replaced by a BEC with erasure probability $\epsilon = 1 - \mathcal{I}(F_u)$.

Furthermore, for any $F \in \mathcal{F}_{\text{sym}}$ we have

$$\mathcal{I}(F \otimes \mathcal{E}_{\text{sym}}(\epsilon)) = 1 - (1 - \mathcal{I}(F))\epsilon.$$

From this result, it is immediate to obtain the approximated DE recursion as

$$\begin{aligned} x_\ell &= 1 - (1 - \mathcal{I}(F_u)) \sum_{i=2}^d \lambda_i (1 - x_{\ell-1}^{a-1} \tilde{x}_{\ell-1}^2)^{i-1} \\ \tilde{x}_\ell &= 1 - (1 - \mathcal{I}(F_u)) (1 - x_{\ell-1}^a \tilde{x}_{\ell-1}). \end{aligned} \quad (41)$$

Notice that (41) is the standard (exact) DE for the IRA ensemble $(\{\lambda_i\}, a)$ over a BEC (see [19]) with the same capacity of the actual binary-input symmetric-output channel, given by $\mathcal{I}(F_u)$. We point out here that this method, consisting of replacing the actual channel with a BEC with equal capacity and optimizing the code ensemble for the BEC, was proposed in [24] for the optimization of LDPC ensembles. Interestingly, this method follows as a special case of our general approach for DE approximation, for a particular choice of the mappings Φ and Ψ .

In this case, the fixed-point equation corresponding to (23) is obtained in closed form as

$$\begin{aligned} x &= 1 - (1 - \mathcal{I}(F_u)) \\ &\times \sum_{i=2}^d \lambda_i \left(1 - \frac{x^{a-1} \mathcal{I}(F_u)^2}{(1 - (1 - \mathcal{I}(F_u))x^a)^2} \right)^{i-1} \end{aligned} \quad (42)$$

(for details, see [19]).

$$\begin{cases} x_\ell = 1 - \sum_{i=2}^d \sum_{j=1}^D \lambda_i p_j \frac{1}{\sqrt{\pi}} \int_{-\infty}^{\infty} e^{-z^2} \log_2 \left(1 + e^{-2\sqrt{(i-1)\mu_\ell}z - (i-1)\mu_\ell - v_j} \right) dz \\ \tilde{x}_\ell = 1 - \sum_{j=1}^D p_j \frac{1}{\sqrt{\pi}} \int_{-\infty}^{\infty} e^{-z^2} \log_2 \left(1 + e^{-2\sqrt{\mu_\ell}z - \tilde{\mu}_\ell - v_j} \right) dz. \end{cases} \quad (35)$$

$$\begin{aligned} x_\ell &= \sum_{i=2}^d \lambda_i J \left(\frac{2}{\sigma^2} + (i-1)J^{-1} \left(1 - J \left((a-1)J^{-1}(1-x_{\ell-1}) + 2J^{-1}(1-\tilde{x}_{\ell-1}) \right) \right) \right) \\ \tilde{x}_\ell &= J \left(\frac{2}{\sigma^2} + J^{-1} \left(1 - J \left(aJ^{-1}(1-x_{\ell-1}) + J^{-1}(1-\tilde{x}_{\ell-1}) \right) \right) \right). \end{aligned} \quad (38)$$

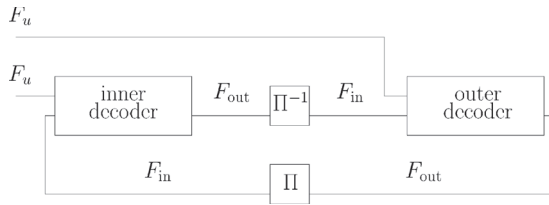


Fig. 5. Turbo-like IRA decoder.

D. Methods 3 and 4

Methods 1 and 2 yield (almost) closed-form DE approximations at the price of some approximations of the message distributions and, above all, of the checknodes output distributions Q_ℓ and \tilde{Q}_ℓ .

In much of the current literature on random-like code ensemble optimization, the EXIT function of a decoding block is obtained by Monte Carlo simulation, by generating i.i.d. input messages, estimating the distribution of the output messages, and computing a one-dimensional quantity [12]–[18]. Following this approach, we shall consider the IRA decoder with turbo-like scheduling (see Fig. 5) and obtain the EXIT functions of the inner and outer decoders.

The inner (accumulator) and outer (repetition) decoders are characterized by an EXIT function as defined in Section III-A, for some guess of the (symmetric) distribution F_{in} . In general, the EXIT function of the decoders can be obtained as follows.

- 1) Let the channel observation messages be i.i.d., $\sim F_u$.
- 2) Assume the decoder input messages are i.i.d., $\sim F_{in}$.
- 3) Obtain either in closed form or by Monte Carlo simulation the corresponding marginal distribution F_{out} of the decoder output messages.
- 4) Let $I_A = \mathcal{I}(F_{in})$, $I_E = \mathcal{I}(F_{out})$ be a point on the EXIT function curve.

Our Methods 3 and 4 consist of applying the above approach under the assumptions $F_{in} = \mathcal{N}_{\text{sym}}(J^{-1}(I_A))$ and $F_{in} = \mathcal{E}_{\text{sym}}(1 - I_A)$, respectively.

Let the resulting EXIT functions of the inner and outer decoders be denoted by $I_E = g(I_A)$ and by $I_E = h(I_A)$, respectively, and let x denote the mutual information between the messages at the output of the outer decoder (repetition code) and the corresponding symbols (information bitnodes).

The resulting approximated DE is given by

$$x_\ell = h(g(x_{\ell-1})). \quad (43)$$

The corresponding fixed-point equation is given by $x = h(g(x))$, and the condition for the uniqueness of the fixed point at $x = 1$, corresponding to (24), is $x < h(g(x))$ for all $x \in [0, 1)$. The resulting IRA optimization methods are obtained by using this condition in (25).

While for the inner decoder (accumulator) we are forced to resort to Monte Carlo simulation, it is interesting to notice that, due to the simplicity of the repetition code, for both Methods 3 and 4 the EXIT function of the outer decoder ($I_E = h(I_A)$) can be obtained in closed form.

For Method 3, by discretizing the channel observation distribution as in (34), we have³

$$h(I_A) = 1 - \sum_{i=2}^d \sum_{j=1}^D \lambda_i p_j \frac{1}{\sqrt{\pi}} \int_{-\infty}^{\infty} e^{-z^2} \times \log_2 \left(1 + e^{-2\sqrt{(i-1)J^{-1}(I_A)z - (i-1)J^{-1}(I_A) - v_j}} \right) dz. \quad (44)$$

For Method 4 we have

$$h(I_A) = 1 - (1 - \mathcal{I}(F_u)) \sum_{i=2}^d \lambda_i (1 - I_A)^{i-1}. \quad (45)$$

IV. PROPERTIES OF THE APPROXIMATED DE

In this section, we show some properties of the approximated DE derived in Section III.

A. Stability Condition

Consider the DE approximation of Method 1. As indicated in Section III-B, $(x, \tilde{x}) = (1, 1)$ is a fixed-point of the system (33)–(35). We have the following result.

Theorem 2: The fixed point at $(1, 1)$ of the system (33)–(35) is stable if and only if the fixed point $(\Delta_\infty, \Delta_\infty)$ of the exact DE (10)–(13) is stable.

Proof: See Appendix III. \square

For other DE approximations, stability does not generally imply stability of the corresponding exact DE. Consider the DE approximation of Method 2. $(1, 1)$ is a fixed point of the system (41). We have the following result.

Proposition 3: The local stability condition of the approximated DE with Method 2 is less stringent than that of the exact DE.

Proof: See Appendix IV. \square

If an approximated DE has a less stringent stability condition, then the exact stability condition must be added to the ensemble optimization as an explicit additional constraint. It should be noticed that the DE approximations used in [24], [14], [19] require the additional stability constraint. For example, the codes presented in [19] for the BIAWGNC and for which $\lambda_2 > 0$ are not stable. Therefore, the BER for an arbitrary large number of iterations is not vanishing.

B. Fixed-Points, Coding Rate, and Channel Capacity

An interesting property of optimization Methods 2 and 4 is that the optimized ensemble for a given channel with channel observation distribution F_u and capacity $C = \mathcal{I}(F_u)$ has coding rate not larger than C . In fact, as a corollary of a general result of [23] (see Appendix V), we have the following.

Theorem 3: The DE approximations of Methods 2 and 4 have unique fixed point $(1, 1)$ only if the IRA ensemble coding rate R satisfies $R < C = \mathcal{I}(F_u)$.

Proof: See Appendix V. \square

³Just prior to the submission of the final revised version of this work we became aware of [36] which proposes essentially the same method as Method 3.

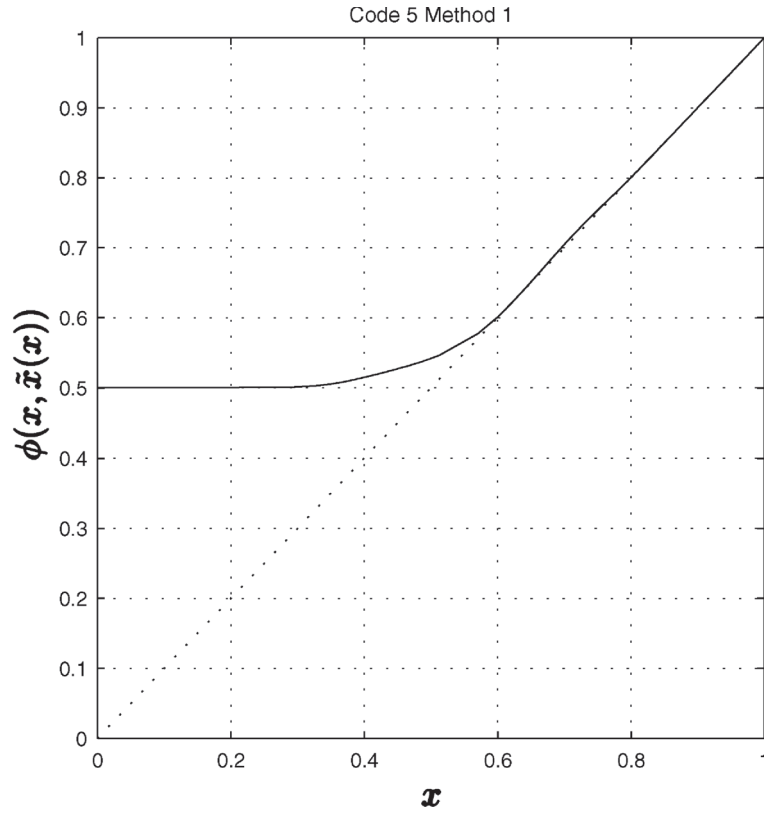


Fig. 6. Mutual information EXIT functions for BIAWGNC and Method 1.

TABLE I
OPTIMIZATION FOR THE BIAWGNC

	Method 1		Method 2		Method 3		Method 4	
	i	λ_i	i	λ_i	i	λ_i	i	λ_i
	2	0.04227	2	0.05554	2	0.05266	2	0.05554
	3	0.16242	3	0.16330	3	0.11786	3	0.14480
	7	0.06529	8	0.06133	5	0.05906	7	0.18991
	8	0.06489	9	0.19357	6	0.06517	8	0.00996
	9	0.06207	25	0.14460	8	0.03615	19	0.03721
	10	0.01273	26	0.08842	9	0.11288	20	0.25894
	11	0.13072	100	0.29323	13	0.06068	100	0.30366
	14	0.04027			14	0.04650		
	25	0.00013			22	0.08606		
	26	0.05410			23	0.01610		
	36	0.13031			34	0.11019		
	37	0.13071			35	0.11919		
	100	0.10402			100	0.11751		
Rate	0.50183		0.49697		0.50154		0.49465	
a	8		8		8		8	
d	7.94153		8.09755		7.95087		8.17305	
SNR(DE)	-2.739		-2.457		-2.727		-2.588	
SNR _{gap} (DE)	0.059		0.406		0.075		0.306	
SNR _{gap} (approx.)	-0.025		0.040		-0.021		0.071	

We show in Section V-A through some examples that this property does not hold in general for other code ensemble optimization methods, for which the ensemble rate R might result to be larger than the (nominal) capacity $\mathcal{I}(F_u)$. This means that the threshold ν^* , evaluated by exact DE, is worse than the channel parameter ν used for the ensemble design.

V. NUMERICAL RESULTS

A. Design Example for Rate-1/2 Codes

In this subsection we present the result of optimization for codes of rate 1/2 and give examples for the BSC with crossover

probability p and the BIAWGNC with signal-to-noise ratio (SNR)

$$\text{SNR} \triangleq \frac{E_s}{N_0} = \frac{1}{2\sigma^2}.$$

In Fig. 6, the curve is the fixed-point equation used for the optimization in Method 1, i.e., the function $\phi(x, \tilde{x}(x))$. The fixed-point equation curves for the other three methods are very similar.

In Fig. 6, the curve (solid line) shows $\phi(x, \tilde{x}(x))$ as a function of $x \in [0, 1]$ for Method 1. The solutions of the fixed point (23)

TABLE II
OPTIMIZATION FOR THE BSC

	Method 1		Method 2		Method 3		Method 4	
	i	λ_i	i	λ_i	i	λ_i	i	λ_i
	2	0.03545	2	0.04732	2	0.03115	2	0.04657
	3	0.14375	3	0.17984	3	0.14991	3	0.14932
	6	0.03057	9	0.19715	6	0.04630	7	0.07693
	7	0.10963	10	0.06259	7	0.06217	8	0.16249
	9	0.10654	26	0.16429	8	0.08666	20	0.07001
	10	0.02388	27	0.05676	10	0.12644	21	0.20550
	11	0.04856	100	0.29205	17	0.03430	100	0.28919
	12	0.00461			18	0.01506		
	21	0.03035			26	0.00228		
	28	0.22576			27	0.02258		
	29	0.09453			28	0.21774		
	100	0.14635			29	0.08021		
					100	0.12521		
Rate	0.48908		0.49620		0.49226		0.49091	
a	8		8		8		8	
d	8.35724		8.12253		8.25157		8.29627	
$p(\text{DE})$	0.1091		0.0938		0.1091		0.1009	
$p_{\text{gap}}(\text{DE})$	0.0046		0.0175		0.0035		0.0122	
$p_{\text{gap}}(\text{approx.})$	0.0037		0.0013		0.0026		0.0018	

correspond to the intersection of this curve with the main diagonal (dotted line). Tables I and II give the degree sequences, the grouping factors, and the information bitnode average degrees for the four methods, for codes of rate $1/2$ over the BIAWGNC and the BSC, respectively. We compute the true iterative decoding thresholds (by using the exact DE) for all the ensembles (denoted by the SNR (DE) or p (DE) in the tables) and report also the gap of these thresholds with respect to the Shannon limit (denoted by SNR_{gap} (DE) or p_{gap} (DE) in the tables). Then, we compare it to the threshold of the approximated DE (SNR_{gap} (approximately) and p_{gap} (approximately)). We observe that the codes designed by using Methods 2 or 4 have rate below capacity, which is consistent with Theorem 3. On the other hand, the codes designed by using Methods 1 or 3 have rate possibly larger than the capacity corresponding to the channel parameter used for design. It can easily be checked that all the designed codes are stable.

B. Thresholds of IRA Ensembles

In this subsection, we present results for codes designed according to the four methods, for rates from 0.1 to 0.9, and we compare the methods on the basis of the true thresholds obtained by DE. We present the code rate, the grouping factor, the average repetition factor, and the gap to Shannon limit, for both BSC and BIAWGNC.

Tables III and IV show the performance of IRA codes on the BIAWGNC. Tables V and VI show the performance of IRA codes on the BSC.

For all rates, and for both channels, IRA codes designed assuming GA (Methods 1 and 3) perform much better than those designed assuming BEC *a priori* (Methods 2 and 4). Nevertheless, Method 4 yields better codes than Method 2, especially at low rates. This is due to the fact that, in Method 2, the communication channel is replaced with a BEC with the same capacity, while this is not the case in Method 4. This difference in performance decreases as the rate increases.

Fig. 7 compares the performance of IRA ensembles with the best known LDPC ensembles [6] on the BIAWGNC. As ex-

TABLE III
IRA CODES, DESIGNED WITH METHODS 1 AND 3, EVALUATED WITH DE, FOR BIAWGNC

Method 1				Method 3			
Rate	a	\bar{d}	SNR_{gap}	Rate	a	\bar{d}	SNR_{gap}
0.10109	2	17.78	0.151	0.10133	2	17.74	0.163
0.20191	3	11.86	0.096	0.20199	3	11.85	0.126
0.30153	4	9.27	0.081	0.30175	4	9.26	0.111
0.40196	6	8.93	0.057	0.40201	6	8.93	0.067
0.50184	8	7.94	0.059	0.50154	8	7.95	0.075
0.60188	11	7.28	0.065	0.60147	11	7.29	0.065
0.70154	16	6.81	0.067	0.70093	16	6.83	0.068
0.79904	29	7.29	0.066	0.79912	29	7.29	0.062
0.89677	61	7.02	0.088	0.89712	61	7.00	0.083

TABLE IV
IRA CODES, DESIGNED WITH METHODS 2 AND 4, EVALUATED WITH DE, FOR BIAWGNC

Method 2				Method 4			
Rate	a	\bar{d}	SNR_{gap}	Rate	a	\bar{d}	SNR_{gap}
0.09407	2	19.26	0.906	0.09752	2	18.51	0.316
0.19842	3	12.12	0.573	0.19725	3	12.21	0.293
0.29767	4	9.44	0.529	0.29671	4	9.48	0.336
0.39703	6	9.11	0.466	0.39445	6	9.21	0.343
0.49697	8	8.10	0.406	0.49465	8	8.17	0.306
0.59689	11	7.43	0.362	0.59577	11	7.46	0.338
0.69580	16	7.00	0.323	0.69584	16	6.99	0.296
0.79737	26	6.61	0.272	0.79678	26	6.63	0.271
0.89827	56	6.34	0.212	0.89826	56	6.34	0.214

pected, the performance of IRA ensembles is inferior to that of LDPC ensembles. However, in view of the simplicity of their encoding and decoding, IRA codes, optimized using Methods 1 or 3, emerge as a very attractive design alternative.

Fig. 8 compares the performance of IRA ensembles obtained via the proposed methods for the BSC. The best codes are those designed with Method 3.

VI. CONCLUSION

This paper has tackled the optimization of IRA codes in the limit for large code block length. This assumption allows to consider a cycle-free graph and enables to evaluate the threshold of the code by iteratively calculating message densities (DE).

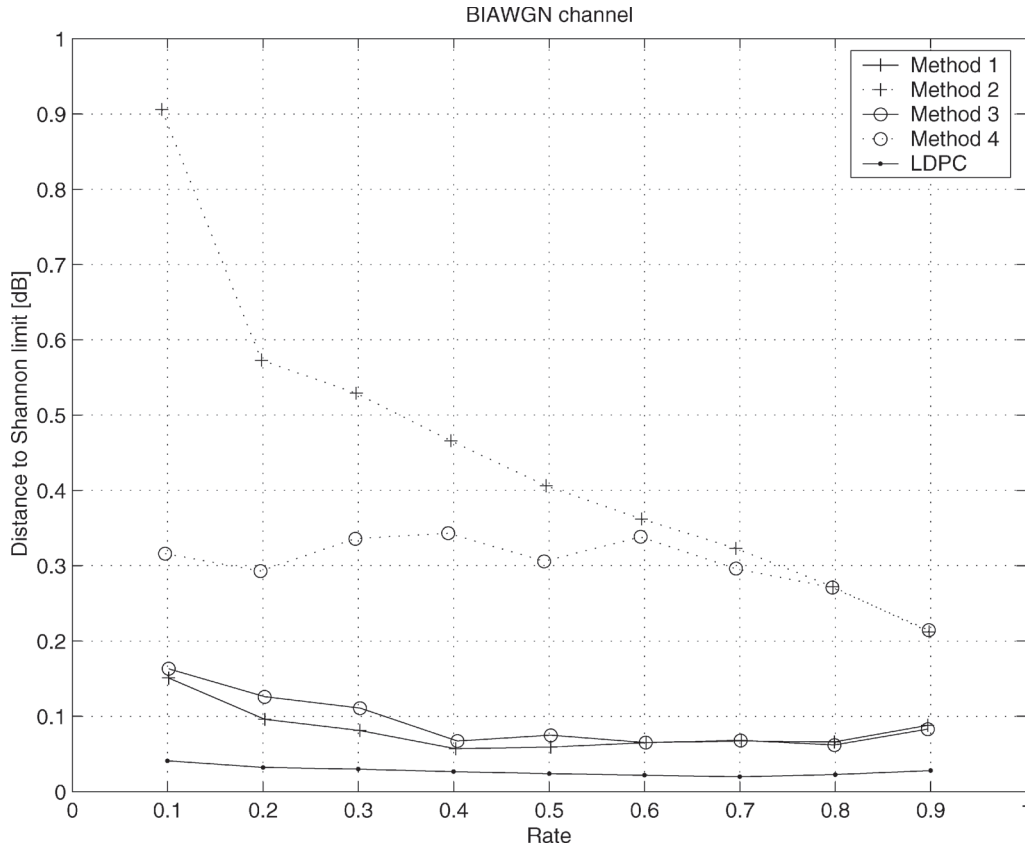


Fig. 7. Gap to Shannon limit (obtained by DE) versus rate for BIAWGN.

TABLE V
IRA CODES, DESIGNED WITH METHODS 1 AND 3, EVALUATED WITH DE, FOR BSC

Method 1				Method 3			
Rate	a	\bar{d}	p_{gap}	Rate	a	\bar{d}	p_{gap}
0.10042	2	17.92	0.0032	0.10137	2	17.73	0.0036
0.19910	3	12.07	0.0037	0.20086	3	11.94	0.0041
0.29573	4	9.53	0.0044	0.29897	4	9.38	0.0031
0.39298	6	9.27	0.0044	0.39621	6	9.14	0.0032
0.48908	8	8.36	0.0046	0.49226	8	8.25	0.0035
0.58590	12	8.48	0.0044	0.58815	11	7.70	0.0040
0.68271	17	7.90	0.0044	0.68409	16	7.39	0.0039
0.78155	28	7.83	0.0038	0.78235	28	7.79	0.0035
0.88437	59	7.71	0.0026	0.88457	63	8.22	0.0025

TABLE VI
IRA CODES, DESIGNED WITH METHODS 2 AND 4, EVALUATED WITH DE, FOR BSC

Method 2				Method 4			
Rate	a	\bar{d}	p_{gap}	Rate	a	\bar{d}	p_{gap}
0.09406	2	19.26	0.0194	0.09952	2	18.10	0.0121
0.19833	3	12.13	0.0175	0.19842	3	12.12	0.0101
0.29743	4	9.45	0.0190	0.28836	4	9.87	0.0114
0.39650	6	9.13	0.0187	0.38865	6	9.44	0.0149
0.49620	8	8.12	0.0175	0.49091	8	8.30	0.0122
0.59580	11	7.46	0.0155	0.59349	11	7.53	0.0124
0.69559	16	7.00	0.0126	0.69107	16	7.15	0.0116
0.79583	26	6.67	0.0091	0.79283	26	6.79	0.0090
0.89692	56	6.44	0.0049	0.89337	57	6.80	0.0051

For the sake of tractable analysis, we proposed four methods to approximate those densities as a one-dimensional parameter. These approximations were motivated by recent results in the field of code design (EXIT functions, reciprocal channel approximation, and the nonstrict convexity of mutual information), and have led to four optimization methods that can all be solved as a linear program.

We found a general stability condition for IRA codes under exact DE. We showed formally that one of the proposed methods (GA, with reciprocal channel approximation) yields a one-dimensional DE approximation with the same stability condition, whereas the exact stability condition must be added to the ensemble optimization as an explicit additional constraint for another method (BEC *a priori*, with reciprocal channel approximation). We derived also results related to the rates of the codes:

in general, the Gaussian *a priori* methods are optimistic, in the sense that there is no guarantee that the optimized rate is below capacity. On the contrary, the BEC *a priori* methods have always rates below capacity.

Our numerical results show that, for the BIAWGN and BSC, the Gaussian *a priori* approximation is more attractive since the codes designed under this assumption have the smallest gap to Shannon limit. Depending on the desired rate, the EXIT function of the inner decoder has to be computed either with Monte Carlo simulation (Method 3) or with the reciprocal channel approximation (Method 1). At least in the BIAWGN there is some evidence that the best LDPC codes [6] designed with DE slightly outperform our designed codes. In view of this and the very simple encoding structure of IRA codes, they emerge as attractive design choices.

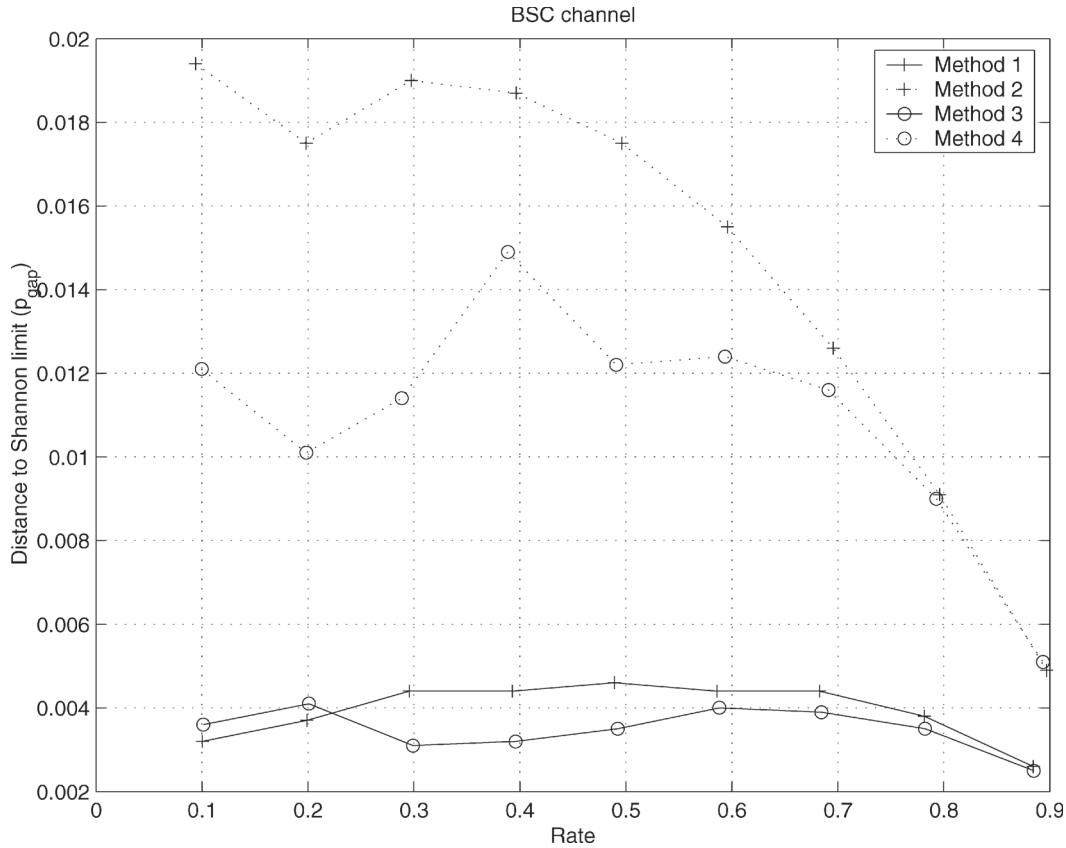


Fig. 8. Gap to Shannon limit (obtained by DE) versus rate for BSC.

APPENDIX I PROOF OF THEOREM 1

We follow in the footsteps of [11] and analyze the local stability of the zero-BER fixed point by using a small perturbation approach. In order to do this, we need more details on the mapping Γ and its inverse.

Given a random variable x with distribution $F_x(z)$, the distribution of $\gamma(x)$ is given by

$$\Gamma(F_x)(s, z) = \chi_{\{s=0\}}\Gamma_0(F_x)(z) + \chi_{\{s=1\}}\Gamma_1(F_x)(z) \quad (46)$$

where

$$\Gamma_0(F_x)(z) = 1 - F_x^- \left(-\log \tanh \frac{z}{2} \right)$$

$$\Gamma_1(F_x)(z) = F_x \left(\log \tanh \frac{z}{2} \right)$$

and where $\chi_{\mathcal{A}}$ denotes the indicator function of the event \mathcal{A} .

In particular, the mapping Γ applied to Δ_0 and Δ_∞ yields

$$\begin{aligned} \Gamma(\Delta_0)(s, z) &= \frac{1}{2}\chi_{\{s=0\}}\Delta_\infty(z) + \frac{1}{2}\chi_{\{s=1\}}\Delta_\infty(z) \\ \Gamma(\Delta_\infty)(s, z) &= \chi_{\{s=0\}}\Delta_0(z). \end{aligned} \quad (47)$$

Given

$$G(s, z) = \chi_{\{s=0\}}G_0(z) + \chi_{\{s=1\}}G_1(z)$$

applying Γ^{-1} yields

$$\begin{aligned} \Gamma^{-1}(G)(z) &= \chi_{\{z>0\}} \left(1 - G_0 \left(-\log \tanh \frac{z}{2} \right) \right) \\ &\quad + \chi_{\{z<0\}} G_1 \left(-\log \tanh \frac{-z}{2} \right). \end{aligned} \quad (48)$$

For the sake of brevity, we introduce the shorthand notation

$$G(s, z) = \chi_{\{s=0\}}G_0(z) + \chi_{\{s=1\}}G_1(z) = \chi_0 G_0 + \chi_1 G_1.$$

The m -fold convolution of $G(s, z)$ by itself is given by

$$\begin{aligned} &(\chi_0 G_0(z) + \chi_1 G_1(z))^{\otimes m} \\ &= \chi_0 \left(\sum_{j=0}^{\lfloor \frac{m}{2} \rfloor} \binom{m}{2j} G_0^{\otimes(m-2j)} \otimes G_1^{\otimes 2j} \right) \\ &\quad + \chi_1 \left(\sum_{j=0}^{\lfloor \frac{m-1}{2} \rfloor} \binom{m}{2j+1} G_0^{\otimes(m-2j-1)} \otimes G_1^{\otimes 2j+1} \right) \end{aligned} \quad (49)$$

where $\lfloor \cdot \rfloor$ stands for the integer part.

In order to study the local stability of the fixed point $(\Delta_\infty, \Delta_\infty)$, we initialize the DE recursion at the point

$$\begin{cases} P_0 = (1 - 2\epsilon)\Delta_\infty + 2\epsilon\Delta_0 \\ \tilde{P}_0 = (1 - 2\delta)\Delta_\infty + 2\delta\Delta_0 \end{cases}$$

for some small $\epsilon, \delta > 0$, and we apply one iteration of the DE recursion (10)–(13). The step-by-step derivation is as follows.

From (47) we have

$$\begin{cases} \Gamma(P_0) = \chi_0 ((1 - 2\epsilon)\Delta_0 + \epsilon\Delta_\infty) + \chi_1 (\epsilon\Delta_\infty) \\ \Gamma(\tilde{P}_0) = \chi_0 ((1 - 2\delta)\Delta_0 + \delta\Delta_\infty) + \chi_1 (\delta\Delta_\infty). \end{cases}$$

By applying (49) we obtain

$$\begin{aligned} \Gamma(P_0)^{\otimes n} &= \chi_0 ((1 - 2n\epsilon)\Delta_0 + n\epsilon\Delta_\infty) \\ &\quad + \chi_1 (n\epsilon\Delta_\infty) + O(\epsilon^2) \\ \Gamma(\tilde{P}_0)^{\otimes 2} &= \chi_0 ((1 - 4\delta)\Delta_0 + 2\delta\Delta_\infty) \\ &\quad + \chi_1 (2\delta\Delta_\infty) + O(\delta^2). \end{aligned}$$

By applying Γ^{-1} we get

$$\begin{cases} Q_1 = \Gamma^{-1} \left(\Gamma(P_0)^{\otimes(a-1)} \otimes \Gamma(\tilde{P}_0)^{\otimes 2} \right) \\ \tilde{Q}_1 = \Gamma^{-1} \left(\Gamma(P_0)^{\otimes a} \otimes \Gamma(\tilde{P}_0) \right) \end{cases}$$

and

$$\begin{aligned} Q_1 &= (1 - 2(a-1)\epsilon - 4\delta)\Delta_\infty \\ &\quad + (2(a-1)\epsilon + 4\delta)\Delta_0 + O(\epsilon^2, \delta^2) \\ \tilde{Q}_1 &= (1 - 2a\epsilon - 2\delta)\Delta_\infty + (2a\epsilon + 2\delta)\Delta_0 + O(\epsilon^2, \delta^2). \end{aligned}$$

Hence, by noticing (50) at the bottom of the page we have

$$\begin{aligned} \lambda(Q_1) &= (1 - 2(a-1)\lambda_2\epsilon - 4\lambda_2\delta)\Delta_\infty \\ &\quad + (2(a-1)\lambda_2\epsilon + 4\lambda_2\delta)\Delta_0 + O(\epsilon^2, \delta^2). \end{aligned}$$

Finally, by using the fact that $P_1 = F_u \otimes \lambda(Q_1)$ and that $\tilde{P}_1 = F_u \otimes \tilde{Q}_1$, the message distributions after one DE iteration are given by

$$\begin{bmatrix} P_1 \\ \tilde{P}_1 \end{bmatrix} = \mathbf{A} \begin{bmatrix} 2\epsilon \\ 2\delta \end{bmatrix} F_u + \left(\begin{bmatrix} 1 \\ 1 \end{bmatrix} - \mathbf{A} \begin{bmatrix} 2\epsilon \\ 2\delta \end{bmatrix} \right) \Delta_\infty + \begin{bmatrix} O(\epsilon^2) \\ O(\delta^2) \end{bmatrix}$$

where

$$\mathbf{A} = \begin{bmatrix} (a-1)\lambda_2 & 2\lambda_2 \\ a & 1 \end{bmatrix}. \quad (51)$$

After ℓ iterations we obtain

$$\begin{aligned} \begin{bmatrix} P_\ell \\ \tilde{P}_\ell \end{bmatrix} &= \mathbf{A}^\ell \begin{bmatrix} 2\epsilon \\ 2\delta \end{bmatrix} F_u^{\otimes \ell} \\ &\quad + \left(\begin{bmatrix} 1 \\ 1 \end{bmatrix} - \mathbf{A}^\ell \begin{bmatrix} 2\epsilon \\ 2\delta \end{bmatrix} \right) \Delta_\infty + \begin{bmatrix} O(\epsilon^2) \\ O(\delta^2) \end{bmatrix}. \end{aligned} \quad (52)$$

From the large deviation theory we get that [11]

$$\begin{aligned} r &= - \lim_{\ell \rightarrow \infty} \frac{1}{\ell} \log \text{Pe}(F_u^{\otimes \ell}) \\ &= - \log \left(\inf_{s>0} \int e^{-sz} dF_u(z) \right) \\ &= - \log \left(\int e^{-z/2} dF_u(z) \right) \end{aligned} \quad (53)$$

where the last equality follows from the fact that $F_u(z) \in \mathcal{F}_{\text{sym}}$.

Then, by applying $\text{Pe}(\cdot)$ to P_ℓ in (52) we obtain that $\lim_{\ell \rightarrow \infty} \text{Pe}(P_\ell) = 0$ (implying that $\lim_{\ell \rightarrow \infty} P_\ell = \Delta_\infty$) if the eigenvalues of the matrix $\mathbf{A}e^{-r}$ are inside the unit circle.

The stability condition is obtained by computing explicitly the largest (in magnitude) eigenvalue. We obtain

$$\frac{1}{2} \left(1 + \lambda_2(a-1) + \sqrt{1 + (2+6a)\lambda_2 + (a-1)^2\lambda_2^2} \right) < e^r. \quad (54)$$

Since the left-hand side (LHS) of (54) is increasing, condition (54) is indeed an upperbound on λ_2 , given explicitly by (15).

APPENDIX II PROOF OF PROPOSITION 2

Proposition 2 is a particular case of a more general result that we state in the following.

Proposition 4: Let X be binary with $P[X=0] = p$ and $P[X=1] = 1-p$. Let S be independent of X and take M (finite) values with $P[S=i] = q_i$. Conditioned on $S=j$, Y is a continuous random variable with conditional density function

$$f_{Y|X=1}^{(j)}(y) = e^{-y} f_{Y|X=0}^{(j)}(y).$$

Then

$$I(X; Y|S) = I(X; Y).$$

Proof of Proposition 4: First, notice that

$$\begin{aligned} f_{Y|X=0}(y) &= \sum_i q_i f_{Y|X=0}^{(i)}(y) = \sum_i q_i e^y f_{Y|X=1}^{(i)}(y) \\ &= e^y f_{Y|X=1}(y). \end{aligned}$$

Hence, we have (55) at the top of the following page. \square

Proof of Proposition 2: The assertion of Proposition 2 follows from Proposition 4 since for a collection of binary-input symmetric-output channels with symmetric transition probability we have that $\forall i, \forall y$

$$\begin{aligned} p_{Y|X,S}(y|X=1, S=i) &= p_{Y|X,S}(-y|X=0, S=i) \\ &= e^{-y} p_{Y|X,S}(y|X=0, S=i). \end{aligned} \quad \square$$

APPENDIX III PROOF OF THEOREM 2

The local stability condition for the system ((33) and (35)) is given by the eigenvalues of the Jacobian matrix for the functions $(\phi, \tilde{\phi})$ in the fixed point $(x, \tilde{x}) = (1, 1)$. The partial derivatives of ϕ and $\tilde{\phi}$ are

$$\frac{\partial \phi}{\partial x}(1, 1) = \sum_{i=2}^d \sum_{j=1}^D \lambda_i p_j (i-1)(a-1) \lim_{\mu \rightarrow +\infty} \frac{J'_{v_j}((i-1)\mu)}{J'(\mu)}$$

$$\frac{\partial \phi}{\partial \tilde{x}}(1, 1) = \sum_{i=2}^d \sum_{j=1}^D \lambda_i p_j (i-1)2 \lim_{\mu \rightarrow +\infty} \frac{J'_{v_j}((i-1)\mu)}{J'(\mu)}$$

$$\frac{\partial \tilde{\phi}}{\partial x}(1, 1) = \sum_{j=1}^D p_j a \lim_{\mu \rightarrow +\infty} \frac{J'_{v_j}(\mu)}{J'(\mu)}$$

$$\frac{\partial \tilde{\phi}}{\partial \tilde{x}}(1, 1) = \sum_{j=1}^D p_j \lim_{\mu \rightarrow +\infty} \frac{J'_{v_j}(\mu)}{J'(\mu)}$$

where

$$J_{v_j}(\mu) \triangleq 1 - \frac{1}{\sqrt{\pi}} \int_{-\infty}^{+\infty} e^{-z^2} \log_2 \left(1 + e^{-2\sqrt{\mu}z - \mu - v_j} \right) dz. \quad (56)$$

$$\begin{aligned} Q_1^{\otimes n} &= \sum_{j=0}^n \binom{n}{j} (1 - 2(a-1)\epsilon - 4\delta)^{n-j} (2(a-1)\epsilon + 4\delta)^j \Delta_\infty^{\otimes n-j} \otimes \Delta_0^{\otimes j} + O(\epsilon^2, \delta^2) \\ &= \begin{cases} \Delta_\infty + O(\epsilon^2, \delta^2), & \text{for } n \geq 2 \\ (1 - 2(a-1)\epsilon - 4\delta)\Delta_\infty + (2(a-1)\epsilon + 4\delta)\Delta_0 + O(\epsilon^2, \delta^2), & \text{for } n = 1 \end{cases} \end{aligned} \quad (50)$$

$$\begin{aligned}
I(X; Y) &= p \int f_{Y|X=0}(y) \log_2 \frac{f_{Y|X=0}(y)}{p f_{Y|X=0}(y) + (1-p) f_{Y|X=1}(y)} dy \\
&\quad + (1-p) \int f_{Y|X=1}(y) \log_2 \frac{f_{Y|X=1}(y)}{p f_{Y|X=0}(y) + (1-p) f_{Y|X=1}(y)} dy \\
&= p \int f_{Y|X=0}(y) \log_2 \frac{1}{p + (1-p)e^{-y}} dy + (1-p) \int f_{Y|X=1}(y) \log_2 \frac{1}{pe^y + (1-p)} dy \\
&= p \int \sum_i^M q_i f_{Y|X=0}^{(i)}(y) \log_2 \frac{1}{p + (1-p)e^{-y}} dy + (1-p) \int \sum_i^M q_i f_{Y|X=1}^{(i)}(y) \log_2 \frac{1}{pe^y + (1-p)} dy \\
&= \sum_i^M q_i \left(p \int f_{Y|X=0}^{(i)}(y) \log_2 \frac{1}{p + (1-p)e^{-y}} dy + (1-p) \int f_{Y|X=1}^{(i)}(y) \log_2 \frac{1}{pe^y + (1-p)} dy \right) \\
&= \sum_i^M q_i \left(p \int f_{Y|X=0}^{(i)}(y) \log_2 \frac{f_{Y|X=0}^{(i)}(y)}{p f_{Y|X=0}^{(i)}(y) + (1-p) f_{Y|X=1}^{(i)}(y)} dy \right. \\
&\quad \left. + (1-p) \int f_{Y|X=1}^{(i)}(y) \log_2 \frac{f_{Y|X=1}^{(i)}(y)}{p f_{Y|X=0}^{(i)}(y) + (1-p) f_{Y|X=1}^{(i)}(y)} dy \right) \\
&= I(X; Y|S). \tag{55}
\end{aligned}$$

Note that $J_0(\mu) = J(\mu)$. Since both limits tend to 0, we derive an asymptotic expansion for $J'_{v_j}(\mu)$ and $J'(\mu)$.

The derivative of J_{v_j} is given by

$$\begin{aligned}
J'_{v_j}(\mu) &= \frac{\log_2(e)}{\sqrt{\mu}} \frac{1}{\sqrt{\pi}} \int_{-\infty}^{+\infty} (z + \sqrt{\mu}) \\
&\quad \times e^{-v_j} \frac{e^{-(z+\sqrt{\mu})^2}}{1 + e^{-2\sqrt{\mu}z - \mu - v_j}} dz.
\end{aligned}$$

Since F_u is symmetric, the sum over j can be rewritten as

$$\sum_{j=1}^D p_j J'_{v_j}(\mu) = p_0 J'_0(\mu) + \sum_{j=1}^{D'} p_j \left(J'_{v_j}(\mu) + e^{-v_j} J'_{-v_j}(\mu) \right).$$

Let us define

$$\begin{aligned}
f_0(\mu) &= \frac{1}{\log_2(e)} J'_0(\mu) \\
f_{v_j}(\mu) &= \frac{1}{\log_2(e)} \left(J'_{v_j}(\mu) + e^{-v_j} J'_{-v_j}(\mu) \right). \tag{57}
\end{aligned}$$

Following [38], (57) can be rewritten as (58) at the bottom of the page. The second equality in (58) is obtained by the change

of variable $z' = z + \sqrt{\mu}/2$. The fourth equality is due to the fact that the first and second integrands in the third line of (58) are odd and even functions of z , respectively. Then we use the changes of variable $z' = \sqrt{\mu}z + \frac{v_j}{2}$ and $z' = \sqrt{\mu}z - \frac{v_j}{2}$.

Lebesgue's dominated convergence theorem completes the proof. Since the sequence of measurable functions verifies

$$\forall z \in \mathbb{R}, \quad \frac{e^{-\frac{z^2}{\mu}}}{\cosh(z)} \xrightarrow{\mu \rightarrow +\infty} \frac{1}{\cosh(z)}$$

and since these functions are bounded by an integrable function independent of μ

$$\forall \mu > 0, \quad \forall z \in \mathbb{R}, \quad \left| \frac{e^{-\frac{z^2}{\mu}}}{\cosh(z)} \right| \leq \frac{1}{\cosh(z)} \in L^1(\mathbb{R}).$$

Thus, Lebesgue's dominated convergence theorem [37] applies and

$$\int_{-\infty}^{+\infty} \frac{e^{-\frac{z^2}{\mu}}}{\cosh(z)} dz \xrightarrow{\mu \rightarrow +\infty}$$

$$\begin{aligned}
f_{v_j}(\mu) &= \frac{1}{\sqrt{\pi}} \int_{-\infty}^{+\infty} \left(1 + \frac{z}{\sqrt{\mu}} \right) e^{-(z+\sqrt{\mu})^2} \left(\frac{e^{-v_j}}{1 + e^{-2\sqrt{\mu}z - \mu - v_j}} + \frac{1}{1 + e^{-2\sqrt{\mu}z - \mu + v_j}} \right) dz. \\
&= \frac{1}{\sqrt{\pi}} \int_{-\infty}^{+\infty} \frac{1}{\sqrt{\mu}} \left(z + \frac{\sqrt{\mu}}{2} \right) e^{-(z+\frac{\sqrt{\mu}}{2})^2} \left(\frac{e^{-v_j}}{1 + e^{-2\sqrt{\mu}z - v_j}} + \frac{1}{1 + e^{-2\sqrt{\mu}z + v_j}} \right) dz \\
&= \frac{1}{\sqrt{\pi}} \int_{-\infty}^{+\infty} \frac{z}{\sqrt{\mu}} e^{-z^2 - \frac{\mu}{4} - \frac{v_j}{2}} \left(\frac{1}{e^{\sqrt{\mu}z + \frac{v_j}{2}} + e^{-\sqrt{\mu}z - \frac{v_j}{2}}} + \frac{1}{e^{\sqrt{\mu}z - \frac{v_j}{2}} + e^{-\sqrt{\mu}z + \frac{v_j}{2}}} \right) dz \\
&\quad + \frac{1}{\sqrt{\pi}} \int_{-\infty}^{+\infty} \frac{1}{2} e^{-z^2 - \frac{\mu}{4} - \frac{v_j}{2}} \left(\frac{1}{e^{\sqrt{\mu}z + \frac{v_j}{2}} + e^{-\sqrt{\mu}z - \frac{v_j}{2}}} + \frac{1}{e^{\sqrt{\mu}z - \frac{v_j}{2}} + e^{-\sqrt{\mu}z + \frac{v_j}{2}}} \right) dz \\
&= \frac{e^{-\frac{\mu}{4} - \frac{v_j}{2}}}{4\sqrt{\pi}} \int_{-\infty}^{+\infty} e^{-z^2} \left(\frac{1}{\cosh(\sqrt{\mu}z + \frac{v_j}{2})} + \frac{1}{\cosh(\sqrt{\mu}z - \frac{v_j}{2})} \right) dz \\
&= \frac{e^{-\frac{\mu}{4} - \frac{v_j}{2}}}{4\sqrt{\pi\mu}} \int_{-\infty}^{+\infty} \frac{e^{-\frac{(z-\frac{v_j}{2})^2}{\mu}} + e^{-\frac{(z+\frac{v_j}{2})^2}{\mu}}}{\cosh(z)} dz. \tag{58}
\end{aligned}$$

$$\int_{-\infty}^{+\infty} \frac{1}{\cosh(z)} dz = [2 \arctan(e^z)]_{-\infty}^{+\infty} = \pi.$$

Therefore, for large μ

$$f_{v_j}(\mu) \sim \frac{\sqrt{\pi}}{2} \frac{e^{-\frac{\mu}{4}} e^{-\frac{v_j}{2}}}{\sqrt{\mu}}.$$

Similarly, we get

$$f_0(\mu) \sim \frac{\sqrt{\pi}}{4} \frac{e^{-\frac{\mu}{4}}}{\sqrt{\mu}}.$$

And thus, for $n \geq 1$

$$\lim_{\mu \rightarrow +\infty} \frac{f_{v_j}(n\mu)}{f_0(\mu)} = \begin{cases} 2e^{-\frac{v_j}{2}}, & \text{if } n = 1 \\ 0, & \text{if } n > 1 \end{cases}$$

and

$$\lim_{\mu \rightarrow +\infty} \frac{f_0(n\mu)}{f_0(\mu)} = \begin{cases} 1, & \text{if } n = 1 \\ 0, & \text{if } n > 1. \end{cases}$$

The partial derivatives of ϕ and $\tilde{\phi}$ are

$$\begin{aligned} \frac{\partial \phi}{\partial x}(1, 1) &= \lambda_2(a-1) \left(p_0 + \sum_{j=1}^{D'} 2p_j e^{-\frac{v_j}{2}} \right) \\ &= \lambda_2(a-1) \sum_{j=1}^D p_j e^{-\frac{v_j}{2}} \\ &= \lambda_2(a-1)e^{-r} \end{aligned} \quad (59)$$

where r is defined in (53). Similarly

$$\frac{\partial \phi}{\partial \tilde{x}}(1, 1) = \lambda_2 2e^{-r} \quad (60)$$

$$\frac{\partial \tilde{\phi}}{\partial x}(1, 1) = ae^{-r} \quad (61)$$

$$\frac{\partial \tilde{\phi}}{\partial \tilde{x}}(1, 1) = e^{-r}. \quad (62)$$

We get the Jacobian matrix as

$$\mathbf{J} = \begin{bmatrix} (a-1)\lambda_2 & 2\lambda_2 \\ a & 1 \end{bmatrix} e^{-r}.$$

In order to be stable, the eigenvalues of \mathbf{J} should be inside the unit circle. Therefore, the stability condition reduces to

$$\frac{1}{2} \left(1 + \lambda_2(a-1) + \sqrt{1 + (2+6a)\lambda_2 + (a-1)^2\lambda_2^2} \right) < e^r. \quad (63)$$

Notice from (54) and (63) that the stability conditions under DE and approximated DE are the same.

APPENDIX IV PROOF OF PROPOSITION 3

The Jacobian matrix of the approximated DE (41) about the fixed point $(x, \tilde{x}) = (1, 1)$, for a given input channel distribution F_u , is

$$\mathbf{J} = \begin{bmatrix} (a-1)\lambda'(0) & 2\lambda'(0) \\ a & 1 \end{bmatrix} (1 - \mathcal{I}(F_u)) = \mathbf{A}(1 - \mathcal{I}(F_u))$$

where \mathbf{A} was already defined in (51). The stability of the exact DE is given by the eigenvalues of $\mathbf{A}e^{-r}$ (where r is defined in (53)) while it is given by those of $\mathbf{A}(1 - \mathcal{I}(F_u))$ for the approximated DE (where $\mathcal{I}(F)$ is given in (28)).

Under the assumption that F_u is symmetric, we get

$$\begin{aligned} \int_{-\infty}^0 e^{-z/2} dF_u(z) &= \int_0^{+\infty} e^{-z/2} dF_u(z) \\ \int_{-\infty}^0 \log_2(1+e^{-z}) dF_u(z) &= \int_0^{+\infty} e^{-z} \log_2(1+e^z) dF_u(z). \end{aligned}$$

It follows that

$$e^{-r} = \int_0^{+\infty} 2e^{-z/2} dF_u(z)$$

and that

$$1 - \mathcal{I}(F_u) = \int_0^{+\infty} \left((1+e^{-z}) \log_2(1+e^{-z}) + \frac{z}{\log 2} e^{-z} \right) dF_u(z).$$

From the inequality

$$\forall z \geq 0, \quad (1+e^{-z}) \log(1+e^{-z}) + ze^{-z} \leq 2(\log 2)e^{-z/2} \quad (64)$$

we get

$$\forall F_u \in \mathcal{F}_{\text{sym}}, \quad 1 - \mathcal{I}(F_u) \leq e^{-r}$$

and the conclusion follows. \square

In the following, we show inequality (64). Letting $x = e^{-z}$, (64) becomes equivalent to

$$\forall x \in [0, 1], \quad f(x) \leq 0$$

where

$$f(x) \triangleq (1+x) \log(1+x) - x \log x - 2 \log 2 \sqrt{x}. \quad (65)$$

It can be shown that $f(x)$ has a single minimum in the open interval $(0, 1)$. Hence, by noticing that

$$\lim_{x \rightarrow 0} f(x) = 0 \quad \text{and} \quad f(1) = 0$$

we get inequality (64).

APPENDIX V PROOF OF THEOREM 3

Theorem 3 follows as a corollary of a result of [23] that we state here for the sake of completeness as Lemma 1. In order to introduce this result, we consider the model of Fig. 9, where \mathbf{b} , \mathbf{x}_1 , and \mathbf{x} are binary sequences and where Channel 1 is the communication channel with output \mathbf{y} and Channel 2 is a BEC channel with output \mathbf{z} . Let the decoder be a maximum *a posteriori* (MAP) symbol-by-symbol decoder, producing for all $i = 1, \dots, n$, output messages of the form

$$m_{o,i} = \log \frac{P(x_{1,i} = 0 | \mathbf{y}, \mathbf{z}_{[i]})}{P(x_{1,i} = 1 | \mathbf{y}, \mathbf{z}_{[i]})} \quad (66)$$

where $\mathbf{z}_{[i]} \triangleq (z_1, \dots, z_{i-1}, z_{i+1}, \dots, z_n)$. Following [23], we generalize the definition of I_A and I_E given in Section III-A to the case of sequences as

$$\begin{aligned} I_A &= \frac{1}{n} \sum_{i=1}^n I(x_{1,i}; z_i) \\ I_E &= \frac{1}{n} \sum_{i=1}^n I(x_{1,i}; m_{o,i}) \end{aligned}$$

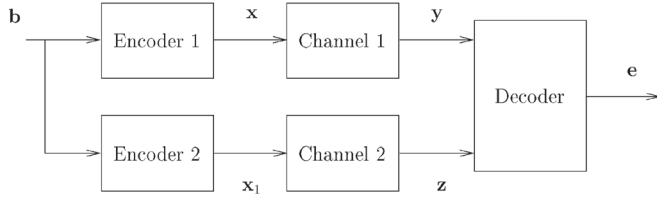


Fig. 9. General decoding model.

$$\stackrel{(a)}{=} \frac{1}{n} \sum_{i=1}^n I(x_{1,i}; \mathbf{y}, \mathbf{z}_{[i]}) \quad (67)$$

where (a) follows from the fact that the decoder is MAP. Again, the decoder EXIT function is the set of points (I_A, I_E) for all $I_A \in [0, 1]$.

For the setup of Fig. 9 with the above assumptions, the following result applies.

Lemma 1: [23] Let \mathbf{b} be uniformly distributed and i.i.d. If Encoder 2 is linear with generator matrix having no all-zero columns, then the area under the EXIT characteristic satisfies

$$\mathcal{A} \triangleq \int_0^1 I_E(z) dz = 1 - \frac{1}{n} H(\mathbf{x}_1 | \mathbf{y}) \quad (68)$$

□

We start by proving Theorem 3 for the approximated DE of Method 4. Consider the IRA encoder of Fig. 1 and the turbo-like decoder of Fig. 5.

The inner MAP decoder receives channel observations \mathbf{u}_p for the parity bits and input messages for the symbols of \mathbf{x}_1 , and produces output messages for the symbols of \mathbf{x}_1 . The general decoding model of Fig. 9, applied to the inner decoder, yields the model of Fig. 10(a).

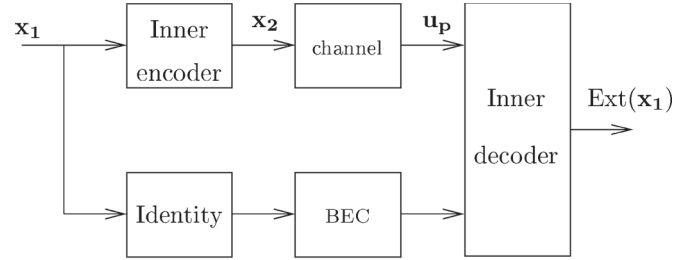
The outer MAP decoder receives channel observations \mathbf{u}_s for the information bits and input messages for the symbols of \mathbf{x}_1 , and produces output messages for the symbols of \mathbf{x}_1 . The general decoding model of Fig. 9, applied to the outer decoder, yields the model of Fig. 10(b).

The upper channel is the communication channel with capacity $\mathcal{I}(F_u)$. Since we consider approximation Method 4, we let lower channel to be a BEC in both Fig. 10(a) and (b). Let k , n , and m denote the number of information bits (length of \mathbf{b} and of \mathbf{u}_s), the number of repeated information bits (length of \mathbf{x}_1), and the number of parity bits (length of \mathbf{x}_2 and of \mathbf{u}_p), respectively. The inner and outer coding rates are $R_{\text{in}} = n/m$ and $R_{\text{out}} = k/n$, and the overall IRA coding rate (3) is given by

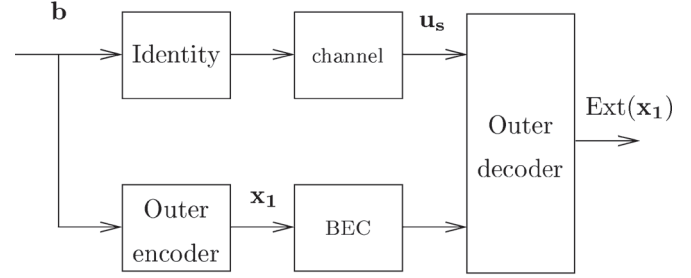
$$R = \frac{k}{k+m} = \frac{R_{\text{in}} R_{\text{out}}}{1 + R_{\text{in}} R_{\text{out}}}.$$

By applying Lemma 1 to the inner code model (Fig. 10(a)), we obtain

$$\begin{aligned} \mathcal{A}_{\text{in}} &= 1 - \frac{1}{n} H(\mathbf{x}_1 | \mathbf{u}_p) \\ &= 1 - \frac{1}{n} (H(\mathbf{x}_1) - I(\mathbf{x}_1; \mathbf{u}_p)) \\ &\stackrel{(a)}{=} \frac{1}{n} I(\mathbf{x}_1; \mathbf{u}_p) \\ &\stackrel{(b)}{=} \frac{m}{n} I(x_{2,i}; u_{p,i}) = \mathcal{I}(F_u) / R_{\text{in}} \end{aligned} \quad (69)$$



(a)



(b)

Fig. 10. Model of inner (a) and outer (b) decoders Method 4.

where (a) follows from the fact that, by the model assumption, \mathbf{x}_1 is an i.i.d. uniformly distributed binary sequence, and (b) follows from the fact that the accumulator (inner code) generates \mathbf{x}_2 with uniform probability (and uniform marginals) if driven by the i.i.d. uniform input sequence \mathbf{x}_1 .

By applying Lemma 1 to the outer code model (Fig. 10(b)), we obtain

$$\begin{aligned} \mathcal{A}_{\text{out}} &= 1 - \frac{1}{n} H(\mathbf{x}_1 | \mathbf{u}_s) \\ &= 1 - \frac{1}{n} (H(\mathbf{x}_1) - I(\mathbf{x}_1; \mathbf{u}_s)) \\ &\stackrel{(a)}{=} 1 - \frac{k}{n} + \frac{1}{n} I(\mathbf{x}_1; \mathbf{u}_s) \\ &\stackrel{(b)}{=} 1 - \frac{k}{n} + \frac{k}{n} I(b_i; u_{s,i}) \\ &= 1 - R_{\text{out}} + R_{\text{out}} \mathcal{I}(F_u) \end{aligned} \quad (70)$$

where both (a) and (b) follow from the fact that the repetition code is an invertible mapping, so the entropy $H(\mathbf{x}_1)$ is equal to the entropy of the information sequence \mathbf{b} (equal to k bits) and $I(\mathbf{x}_1; \mathbf{u}_s) = I(\mathbf{b}; \mathbf{u}_s) = kI(b_i; u_{s,i}) = k\mathcal{I}(F_u)$.

As seen in Section III-D, the approximated DE has no fixed points other than $(1, 1)$ if and only if $g(x) > h^{-1}(x)$ for all $x \in [0, 1]$, where $g(x)$ and $h(x)$ denote the inner and outer decoder EXIT functions. This implies that

$$\mathcal{A}_{\text{in}} = \int_0^1 g(x) dx > \int_0^1 h^{-1}(x) dx = 1 - \mathcal{A}_{\text{out}}.$$

By using (69) and (70), we obtain

$$\begin{aligned} \mathcal{I}(F_u) / R_{\text{in}} &> R_{\text{out}} - R_{\text{out}} \mathcal{I}(F_u) \\ &\Downarrow \\ \mathcal{I}(F_u) &> \frac{R_{\text{in}} R_{\text{out}}}{1 + R_{\text{in}} R_{\text{out}}} = R. \end{aligned} \quad (71)$$

For Method 2, the above derivation still holds, since the communication channel in Fig. 9 is replaced by a BEC with erasure

probability $\epsilon = 1 - \mathcal{I}(F_u)$. In fact, the inner and outer decoder EXIT functions can be rewritten as

$$h(x) = 1 - (1 - \mathcal{I}(F_u)) \sum_{i=2}^d \lambda_i (1-x)^{i-1}$$

$$g(x) = \frac{x^{a-1} \mathcal{I}(F_u)^2}{(1 - (1 - \mathcal{I}(F_u))x^a)^2}$$

and the area under these functions are again

$$A_{\text{out}} = \int_0^1 h(x) dx = 1 - (1 - \mathcal{I}(F_u)) \sum_{i=2}^d \lambda_i / i$$

$$= 1 - R_{\text{out}} + R_{\text{out}} \mathcal{I}(F_u)$$

$$A_{\text{in}} = \int_0^1 g(x) dx = \mathcal{I}(F_u) / a = \mathcal{I}(F_u) / R_{\text{in}}.$$

Therefore, the final result (71) holds also for Method 2.

ACKNOWLEDGMENT

The authors wish to thank Dr. Alex Ashikhmin for the helpful discussion concerning the results in [23].

REFERENCES

- [1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting and decoding: Turbo codes," in *Proc. IEEE Int. Conf. Communications*, Geneva, Switzerland, May 1993, pp. 1064–1070.
- [2] R. G. Gallager, *Low-Density Parity-Check Codes*. Cambridge, MA: MIT Press, 1963.
- [3] M. Luby, M. Mitzenmacher, A. Shokrollahi, D. Spielman, and V. Stemann, "Practical loss-resilient codes," in *Proc. 29th ACM Symp. Theory of Computing (STOC)*, 1997, pp. 150–159.
- [4] M. Luby, M. Mitzenmacher, A. Shokrollahi, and D. Spielman, "Efficient erasure correcting codes," *IEEE Trans. Inform. Theory*, vol. 47, pp. 569–584, Feb. 2001.
- [5] D. Divsalar, H. Jin, and R. McEliece, "Coding theorems for 'Turbo-like' codes," in *Proc. 36th Annu. Allerton Conf. Communication, Control, and Computing*, Sept. 1998, pp. 201–210.
- [6] R. Urbanke *et al.* (2002) Web page. [Online]. Available: <http://lthcwww.epfl.ch/research/ldpcopt/>
- [7] N. Varnica and A. Kavčić, "Optimized LDPC codes for partial response channels," in *Proc. IEEE Int. Symp. Information Theory (ISIT 2002)*, Lausanne, Switzerland, June/July 2002, p. 197.
- [8] X. Ma, N. Varnica, and A. Kavčić, "Matched information rate codes for binary ISI channels," in *Proc. IEEE Int. Symp. Information Theory (ISIT 2002)*, Lausanne, Switzerland, June/July 2002, p. 269.
- [9] B. M. Kurkoski, P. H. Siegel, and J. K. Wolf, "Joint message-passing decoding of LDPC codes and partial-response channels," *IEEE Trans. Inform. Theory*, vol. 48, pp. 1410–1422, June 2002.
- [10] M. Luby, M. Mitzenmacher, A. Shokrollahi, and D. Spielman, "Analysis of low-density codes and improved designs using irregular graphs," in *Proc. 30th ACM Symp. Theory of Computing*, 1998, pp. 249–258.
- [11] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 47, pp. 619–637, Feb. 2001.
- [12] S. ten Brink, "Designing iterative decoding schemes with the extrinsic information transfer chart," *AEÜ Int. J. Electron. Commun.*, vol. 54, no. 6, pp. 389–398, Dec. 2000.
- [13] —, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Trans. Commun.*, vol. 49, pp. 1727–1737, Oct. 2001.
- [14] S.-Y. Chung, T. J. Richardson, and R. Urbanke, "Analysis of sum-product decoding of low-density parity-check codes using a Gaussian approximation," *IEEE Trans. Inform. Theory*, vol. 47, pp. 657–670, Feb. 2001.
- [15] H. El Gamal and A. R. Hammons, Jr., "Analyzing the turbo decoder using the Gaussian approximation," *IEEE Trans. Inform. Theory*, vol. 47, pp. 671–686, Feb. 2001.
- [16] J. Boutros and G. Caire, "Iterative multiuser joint decoding: Unified framework and asymptotic analysis," *IEEE Trans. Inform. Theory*, vol. 48, pp. 1772–1793, July 2002.
- [17] F. Lehmann and G. M. Maggio, "An approximate analytical model of the message passing decoder of LDPC codes," in *Proc. IEEE Int. Symp. Information Theory (ISIT 2002)*, Lausanne, Switzerland, June/July 2002, p. 31.
- [18] M. Ardakani and F. R. Kschischang, "Designing irregular LDPC codes using exit charts based on message error rate," in *Proc. IEEE Int. Symp. Information Theory (ISIT 2002)*, Lausanne, Switzerland, June/July 2002, p. 454.
- [19] H. Jin, A. Khandekar, and R. McEliece, "Irregular repeat-accumulate codes," in *Proc. 2nd Int. Symp. Turbo Codes and Related Topics*, Brest, France, Sept. 2000, pp. 1–8.
- [20] J. Boutros, G. Caire, E. Viterbo, H. Sawaya, and S. Vialle, "Turbo code at 0.03 dB from capacity limit," in *Proc. IEEE Int. Symp. Information Theory (ISIT 2002)*, Lausanne, Switzerland, June/July 2002, p. 56.
- [21] H. Jin, "Analysis and design of turbo-like codes," Ph.D. dissertation, Calif. Inst. Technol., Pasadena, 2001.
- [22] A. Ashikhmin, G. Kramer, and S. ten Brink, "Extrinsic information transfer functions: A model and two properties," in *Proc. 36th Annu. Conf. Information Sciences and Systems (CISS 2002)*, Princeton, NJ, Mar. 2002.
- [23] —, "Code rate and the area under extrinsic information transfer curves," in *Proc. IEEE Int. Symp. Information Theory (ISIT 2002)*, Lausanne, Switzerland, July 2002, p. 115.
- [24] S. Y. Chung, "On the construction of some capacity-approaching coding schemes," Ph.D. dissertation, MIT, Cambridge, MA, 2000.
- [25] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inform. Theory*, vol. IT-27, pp. 533–547, Sept. 1981.
- [26] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, CA: Morgan Kaufmann, 1988.
- [27] R. J. McEliece, D. J. C. MacKay, and J.-F. Cheng, "Turbo decoding as an instance of Pearl's belief propagation algorithm," *IEEE J. Select. Areas Communications*, vol. 16, pp. 140–152, Feb. 1998.
- [28] F. R. Kschischang and B. J. Frey, "Iterative decoding of compound codes by probability propagation in graphical models," *IEEE J. Selected Areas Commun.*, vol. 16, pp. 219–230, Feb. 1998.
- [29] D. Forney, "Codes on graphs: Normal realizations," *IEEE Trans. Inform. Theory*, vol. 47, pp. 520–548, Feb. 2001.
- [30] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, vol. IT-20, pp. 284–287, Mar. 1974.
- [31] T. J. Richardson and R. L. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. Inform. Theory*, vol. 47, pp. 599–618, Feb. 2001.
- [32] S. Y. Chung, G. D. Forney, Jr., T. J. Richardson, and R. Urbanke, "On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit," *IEEE Commun. Lett.*, vol. 5, pp. 58–60, Feb. 2001.
- [33] S. ten Brink, "Exploiting the chain rule of mutual information for the design of iterative decoding schemes," in *Proc. 39th Annu. Allerton Conf. Communication, Control, and Computing*, Oct. 2001, pp. 293–300.
- [34] M. Tüchler, S. ten Brink, and J. Hagenauer, "Measures for tracing convergence of iterative decoding algorithms," in *Proc. 4th Int. ITG Conf. Source and Channel Coding*, Berlin, Germany, Jan. 2002, pp. 53–60.
- [35] S. Huettinger and J. Huber, "Extrinsic and intrinsic information in systematic coding," in *Proc. IEEE Int. Symp. Information Theory (ISIT 2002)*, Lausanne, Switzerland, June/July 2002, p. 116.
- [36] S. ten Brink and G. Kramer, "Turbo processing for scalar and vector channels," in *Proc. 3rd Int. Symp. Turbo Codes and Related Topics*, Brest, France, Sept. 2003, pp. 23–30.
- [37] A. Browder, *Mathematical Analysis: An Introduction*. New York: Springer-Verlag, 1996.
- [38] T. F. Wong, "Numerical calculation of symmetric capacity of Rayleigh fading channel with BPSK/QPSK," *IEEE Commun. Lett.*, vol. 5, pp. 328–330, Aug. 2001.