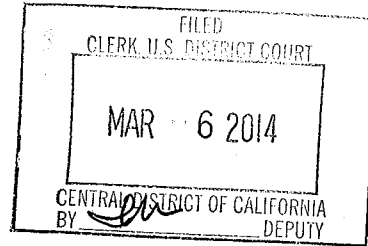


ORIGINAL

1 QUINN EMANUEL URQUHART  
& SULLIVAN, LLP  
2 James R. Asperger (Bar No. 083188)  
jimasperger@quinnemanuel.com  
3 865 S. Figueroa St., 10th Floor  
Los Angeles, California 90017  
4 Telephone: (213) 443-3000  
Facsimile: (213) 443-3100

5 Kevin P.B. Johnson (Bar No. 177129)  
6 kevinjohnson@quinnemanuel.com  
555 Twin Dolphin Drive, 5th Floor  
7 Redwood Shores, California 94065  
Telephone: (650) 801-5000  
8 Facsimile: (650) 801-5100

9 Attorneys for Plaintiff the California  
Institute of Technology



10  
11  
12 UNITED STATES DISTRICT COURT  
13 CENTRAL DISTRICT OF CALIFORNIA

14  
15 The CALIFORNIA INSTITUTE OF  
TECHNOLOGY, a California  
16 corporation,

17 Plaintiff,

18 vs.

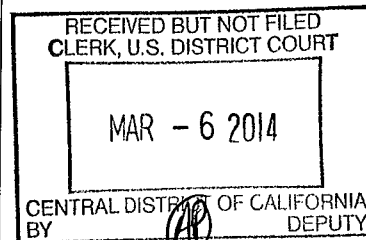
19 HUGHES COMMUNICATIONS,  
INC., a Delaware corporation,  
20 HUGHES NETWORK SYSTEMS,  
LLC, a Delaware limited liability  
21 company, DISH NETWORK  
CORPORATION, a Nevada  
22 corporation, DISH NETWORK L.L.C.,  
a Colorado limited liability company,  
23 and DISHNET SATELLITE  
BROADBAND L.L.C., a Colorado  
24 limited liability company,

25 Defendants.  
26  
27  
28

CASE NO. 2:13-cv-07245-MRP-JEM

**AMENDED COMPLAINT FOR  
PATENT INFRINGEMENT**

**JURY TRIAL DEMANDED**



1 Plaintiff the California Institute of Technology (“Caltech” or “Plaintiff”), by  
2 and through its undersigned counsel, complains and alleges as follows against  
3 Hughes Communications, Inc., Hughes Network Systems, LLC, DISH Network  
4 Corporation, DISH Network L.L.C., and dishNET Satellite Broadband L.L.C.  
5 (collectively, “Defendants”):

6 **NATURE OF THE ACTION**

7 1. This is a civil action for patent infringement arising under the patent  
8 laws of the United States, 35 U.S.C. §§ 1 *et seq.*

9 2. Defendants have infringed and continue to infringe, contributed to and  
10 continue to contribute to the infringement of, and/or actively induced and continue  
11 to induce others to infringe Caltech’s U.S. Patent No. 7,116,710, U.S. Patent No.  
12 7,421,032, U.S. Patent No. 7,916,781, and U.S. Patent No. 8,284,833 (collectively,  
13 “the Asserted Patents”). Caltech is the legal owner by assignment of the Asserted  
14 Patents, which were duly and legally issued by the United States Patent and  
15 Trademark Office. Caltech seeks injunctive relief and monetary damages.

16 **THE PARTIES**

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

20 4. On information and belief, Hughes Communications, Inc. (“Hughes  
21 Communications”) is a corporation organized under the laws of the State of  
22 Delaware, with its principal place of business located at 11717 Exploration Lane,  
23 Germantown, Maryland 20876. On information and belief, Hughes  
24 Communications is a wholly-owned subsidiary of Hughes Satellite Systems  
25 Corporation, which is a wholly-owned subsidiary of EchoStar Corporation  
26 (“EchoStar”).

27 5. On information and belief, Hughes Network Systems, LLC (“Hughes  
28 Network”) is a limited liability company organized under the laws of the State of

1 Delaware, with its principal place of business located at 11717 Exploration Lane,  
2 Germantown, Maryland 20876. On information and belief, Hughes Network is a  
3 wholly owned subsidiary of Hughes Communications. Hughes Communications  
4 and Hughes Network, collectively, are referred to as “Hughes Defendants.”

5 6. On information and belief, DISH Network Corporation (“DISH Corp.”)  
6 is a corporation organized under the laws of the State of Nevada with its principal  
7 place of business located at 9601 South Meridian Boulevard, Englewood, Colorado  
8 80112.

9 7. On information and belief, DISH Network L.L.C. (“DISH L.L.C.”) is a  
10 limited liability company organized under the laws of the State of Colorado with its  
11 principal place of business located at 9601 South Meridian Boulevard, Englewood,  
12 Colorado 80112. On information and belief, DISH L.L.C. is a wholly owned  
13 subsidiary of DISH Corp.

14 8. On information and belief, dishNET Satellite Broadband L.L.C.  
15 (“dishNET”) is a limited liability company organized under the laws of the State of  
16 Colorado with its principal place of business located at 9601 South Meridian  
17 Boulevard, Englewood, Colorado 80112. On information and belief, dishNET is a  
18 wholly owned subsidiary of DISH Corp. On information and belief, dishNET and  
19 DISH L.L.C. are related entities. DISH Corp., DISH L.L.C., and dishNET,  
20 collectively, are referred to as “Dish Defendants.”

21 9. On information and belief, Hughes Defendants’ parent company,  
22 EchoStar, and Dish Defendants were previously one company. On information and  
23 belief, around January 2008, EchoStar and Dish Defendants became two separate  
24 companies (the “spin-off”).

25 10. On information and belief, the business relationship among Dish  
26 Defendants, EchoStar and Hughes Defendants remains extremely integrated. The  
27 same individual serves as the Chairman of both Dish Defendants and EchoStar.  
28 Further, since the spin-off, a substantial majority of the voting power of the shares

1 of both Dish Defendants and EchoStar is owned beneficially by the Chairman, or by  
2 certain trusts established by the Chairman. Additionally, on information and belief,  
3 in addition to the Chairman, an individual responsible for the development and  
4 implementation of advanced technologies that are of potential utility and importance  
5 to both Dish Defendants and EchoStar serves on the board of both companies. On  
6 information and belief, in 2010, Dish Defendants accounted for 82.5% of EchoStar's  
7 total revenue and in 2012, Dish Defendants accounted for 49.5% of EchoStar's total  
8 revenue. Additionally, on information and belief, in October 2012, Dish Defendants  
9 and Hughes Defendants entered into a distribution agreement relating to Hughes  
10 Defendants' satellite internet service.

11 **JURISDICTION AND VENUE**

12 11. This Court has jurisdiction over the subject matter of this action under  
13 28 U.S.C. §§ 1331 and 1338(a).

14 12. Hughes Defendants are subject to this Court's personal jurisdiction. On  
15 information and belief, Hughes Defendants regularly conduct business in the State  
16 of California, including in the Central District of California, and have committed  
17 acts of patent infringement and/or contributed to or induced acts of patent  
18 infringement by others in this District and elsewhere in California and the United  
19 States. As such, Hughes Defendants have purposefully availed themselves of the  
20 privilege of conducting business within this District; have established sufficient  
21 minimum contacts with this District such that they should reasonably and fairly  
22 anticipate being haled into court in this District; have purposefully directed activities  
23 at residents of this State; and at least a portion of the patent infringement claims  
24 alleged herein arise out of or are related to one or more of the foregoing activities.

25 13. Dish Defendants are subject to this Court's personal jurisdiction. On  
26 information and belief, Dish Defendants regularly conduct business in the State of  
27 California, including in the Central District of California, maintain employees in this  
28 District and elsewhere in California, and have committed acts of patent infringement



1 and/or contributed to or induced acts of patent infringement by others in this District  
2 and elsewhere in California and the United States. As such, Dish Defendants have  
3 purposefully availed themselves of the privilege of conducting business within this  
4 District; have established sufficient minimum contacts with this District such that  
5 they should reasonably and fairly anticipate being haled into court in this District;  
6 have purposefully directed activities at residents of this State; and at least a portion  
7 of the patent infringement claims alleged herein arise out of or are related to one or  
8 more of the foregoing activities.

9 14. Venue is proper in this judicial district pursuant to 28 U.S.C. §§ 1391  
10 and 1400 because Defendants regularly conduct business in this District, and certain  
11 of the acts complained of herein occurred in this District.

12 **CALTECH'S ASSERTED PATENTS**

13 15. On October 3, 2006, the United States Patent Office issued U.S. Patent  
14 No. 7,116,710, titled "Serial Concatenation of Interleaved Convolutional Codes  
15 Forming Turbo-Like Codes" (the "'710 patent"). A true and correct copy of the  
16 '710 patent is attached hereto as Exhibit A.

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

22 17. On March 29, 2011, the United States Patent Office issued U.S. Patent  
23 No. 7,916,781, titled "Serial Concatenation of Interleaved Convolutional Codes  
24 Forming Turbo-Like Codes" (the "'781 patent"). A true and correct copy of the  
25 '781 patent is attached hereto as Exhibit C. The '781 patent is a continuation of the  
26 application that led to the '032 patent, which is a continuation of the application that  
27 led to the '710 patent.

28 18. On October 9, 2012, the United States Patent Office issued U.S. Patent

1 No. 8,284,833, titled “Serial Concatenation of Interleaved Convolutional Codes  
2 Forming Turbo-Like Codes” (the “’833 patent”). A true and correct copy of the  
3 ’833 patent is attached hereto as Exhibit D. The ’833 patent is a continuation of the  
4 application that led to the ’781 patent, which is a continuation of the application that  
5 led to the ’032 patent, which is a continuation of the application that led to the ’710  
6 patent.

7 19. The Asserted Patents identify Hui Jin, Aamod Khandekar, and Robert  
8 J. McEliece as the inventors (the “Named Inventors”).

9 20. Caltech is the owner of all right, title, and interest in and to each of the  
10 Asserted Patents with full and exclusive right to bring suit to enforce the Asserted  
11 Patents, including the right to recover for past damages and/or royalties.

12 21. The Asserted Patents are valid and enforceable.

13 **BACKGROUND TO THIS ACTION**

14 22. The Asserted Patents disclose a seminal improvement to coding  
15 systems and methods used for digital satellite transmission. The Asserted Patents  
16 disclose an ensemble of codes called irregular repeat-accumulate (IRA) codes,  
17 which are specific types of low-density parity check (LDPC) codes. The IRA codes  
18 disclosed in the Asserted Patents enable a transmission rate close to the theoretical  
19 limit, while also providing the advantage of a low encoding complexity.

20 23. In September 2000, the Named Inventors of the Asserted Patents  
21 published a paper regarding their invention, titled “Irregular Repeat-Accumulate  
22 Codes” for the Second International Conference on Turbo Codes. (Exhibit E.) This  
23 paper has been widely cited by experts in the industry.

24 24. Experts recognize the importance and usefulness of the IRA codes  
25 disclosed in the September 2000 paper by the Named Inventors of the Asserted  
26 Patents. For example, a paper praising these IRA codes was published in August  
27 2004 by Aline Roumy, Souad Guemghar, Giuseppe Caire, and Sergio Verdú in the  
28 IEEE Transactions on Information Theory. This paper, titled “Design Methods for

1 Irregular Repeat-Accumulate Codes,” states:

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

8 (Exhibit F, at 1.) This paper also notes that, four years after the September 2000  
9 paper, the Named Inventors were the only ones to propose a method to design IRA  
10 codes. (*Id.*)

11 25. The current standard for digital satellite transmissions embodies the  
12 invention of the Asserted Patents by using channel codes that are IRA codes. This  
13 digital satellite transmission standard is titled “Digital Video Broadcasting (DVB);  
14 Second generation framing structure, channel coding and modulation systems for  
15 Broadcasting, Interactive Services, News Gathering and other broadband satellite  
16 applications” (the “DVB-S2 standard”).

17 26. Experts in the industry recognize that the DVB-S2 standard uses the  
18 IRA codes initially disclosed by the Named Inventors of the Asserted Patents. For  
19 example, a 2005 paper published by the highly regarded Institute of Electrical and  
20 Electronics Engineers (IEEE), titled “A Synthesizable IP Core for DVB-S2 LDPC  
21 Code Decoding,” and authored by Frank Kienle, Torben Brack, and Norbert Wehn  
22 recognizes:

23 The LDPC codes as defined in the DVB-S2 standard are IRA  
24 codes, thus the encoder realization is straight forward.  
25 Furthermore, the DVB-S2 code shows regularities which can be  
26 exploited for an efficient hardware realization.

27 (Exhibit G, at 1.)

28 27. Moreover, this paper provides credit to the September 2000 paper  
authored by the Named Inventors of the Asserted Patents for the origination of the  
IRA codes that are defined in the DVB-S2 standard. (*Id.* at 1 & n.8.)

1           28. Similarly, on information and belief, a 2007 paper titled “Factorizable  
2 Modulo  $M$  Parallel Architecture for DVB-S2 LDPC Decoding,” and published in the  
3 Proceedings of the 6th Conference on Telecommunications, recognizes that the  
4 DVB-S2 standard uses the IRA codes initially disclosed by the Named Inventors of  
5 the Asserted Patents. This paper, authored by Marco Gomes, Gabriel Falcão, Vitor  
6 Silva, Vitor Ferreira, Alexandre Sengo, and Miguel Falcão, states:

7                   The new DVB-S2 [] standard adopted a special class of LDPC  
8 codes known by IRA codes [] as the main solution for the FEC  
9 system.

10 (Exhibit H, at 1.)

11           29. Moreover, this paper also credits the September 2000 paper authored  
12 by the Named Inventors of the Asserted Patents for the origination of the IRA codes  
13 that are defined in the DVB-S2 standard. (*Id.* at 1 & n.8.)

14           30. As even further support, on information and belief, a 2006 industry  
15 paper published in the Journal of Communications Software and Systems, titled  
16 “Design of LDPC Codes: A Survey and New Results” and authored by Gianluigi  
17 Liva, Shumei Song, Lan Lan, Yifei Zhang, Shu Lin, and William E. Ryan, confirms  
18 that the DVB-S2 standard uses the IRA codes, stating:

19                   The ETSI DVB S2 [] standard for digital video broadcast  
20 specifies two IRA code families with block lengths 64800 and  
21 16200.

22 (Exhibit I, at 10-11.)

23           31. As such, products, methods, equipment, and/or services that implement  
24 the DVB-S2 standard practice one or more claims of each of the Asserted Patents  
25 because the DVB-S2 standard embodies the invention of the Asserted Patents by  
26 using IRA codes.

27           32. On information and belief, Hughes Defendants manufacture, use,  
28 import, offer for sale, or sell products, methods, equipment, and/or services that  
implement the DVB-S2 standard. For example, Hughes Defendants provide satellite

1 broadband internet access to consumers and broadband network services to the  
2 enterprise markets, among other activities, including through their HN System and  
3 HX System product lines. Hughes Defendants have extensively publicized that their  
4 flagship HN System and HX System satellite broadband internet product lines  
5 implement the DVB-S2 standard. On information and belief, Hughes Defendants  
6 market and sell, among other activities, certain broadband equipment and services  
7 that implements the DVB-S2 standard through the HughesNet brand. On  
8 information and belief, Hughes Defendants further sell or provide certain broadband  
9 equipment and services that implements the DVB-S2 standard to Dish Defendants.  
10 On information and belief, Hughes Defendants use their broadband equipment that  
11 implements the DVB-S2 standard for testing, consulting, and/or support services,  
12 among other activities.

13       33. On information and belief, Dish Defendants manufacture, use, import,  
14 offer for sale, or sell products, methods, equipment, and/or services that implement  
15 the DVB-S2 standard. For example, on information and belief, Dish Defendants  
16 manufacture, market, offer for sale, sell, distribute, and/or use, among other  
17 activities, the Hopper set-top box that implements the DVB-S2 standard.  
18 Additionally, for example, on information and belief, Dish Defendants market, offer  
19 for sale, sell, and distribute, among other activities, Hughes Defendants' satellite  
20 internet service, among other products and services, under the dishNET brand  
21 pursuant to a distribution agreement entered into with Hughes Defendants in  
22 October 2012. On information and belief, Dish Defendants purchase certain  
23 broadband equipment and services that implements the DVB-S2 standard from  
24 Hughes Defendants and offer for sale, sell, provide, and/or distribute this equipment  
25 and service to its customers. On information and belief, Dish Defendants use this  
26 broadband equipment and service that implements the DVB-S2 standard for testing,  
27 consulting and/or support services, among other activities. On information and  
28 belief, the dishNET services are primarily bundled with other services offered by

1 Dish Defendants.

2 34. Hughes Defendants admit that their broadband satellite systems are  
3 compliant with “high-speed DVB-S2.” (Exhibit J.) Additionally, Hughes  
4 Defendants have touted that implementation of this DVB-S2 standard “provides for  
5 higher throughputs, better coding efficiency, and improved satellite resource  
6 utilization for the outbound channel.” (Exhibit K.)

7 35. Further, Hughes Defendants’ website advertises its HX System and  
8 provides a link to a brochure titled “High-Performance IP Satellite Broadband  
9 System.” (Exhibit L.) This brochure similarly highlights Hughes Defendants’  
10 implementation of the DVB-S2 standard, stating that the core component of the HX  
11 System, the HX Gateway, “uses a DVB-S2 carrier . . . for the outbound channel  
12 received by all HX System remote terminals.” (*Id.*)

13 36. Hughes Defendants’ website also advertises its HN System and states  
14 that it is compliant with DVB-S2. (Exhibit M.)

15 **COUNT I**

16 **Infringement of the ’710 Patent**

17 37. Plaintiff re-alleges and incorporates by reference the allegations of the  
18 preceding paragraphs of this Complaint as if fully set forth herein.

19 38. On information and belief, in violation of 35 U.S.C. § 271, Defendants  
20 have infringed and are currently infringing, directly and/or through intermediaries,  
21 the ’710 patent by making, using, selling, offering for sale, and/or importing into the  
22 United States, without authority, products, methods, equipment, and/or services that  
23 practice one or more claims of the ’710 patent. These products, methods,  
24 equipment, and/or services include products that implement the DVB-S2 standard,  
25 including without limitation products in the HN System and HX System product  
26 lines, satellite internet product lines distributed under the dishNET brand, the  
27 Hopper set-top box, network and network services that employ these products,  
28 and/or marketing, consulting, and/or support services provided for these products



1 and services (collectively, the “Accused Services and Products”). For example, at  
2 least Paragraphs 32 and 33 illustrate a limited number of examples of Defendants’  
3 direct infringement of the ’710 patent. Defendants have infringed and are currently  
4 infringing literally and/or under the doctrine of equivalents.

5 39. On information and belief, in violation of 35 U.S.C. § 271, Defendants  
6 have infringed and are continuing to infringe the ’710 patent by contributing to  
7 and/or actively inducing the infringement by others of the ’710 patent by making,  
8 using, selling, offering for sale, and/or importing into the United States, without  
9 authority, products, methods, equipment, and/or services, including the Accused  
10 Services and Products, that practice one or more claims of the ’710 patent.

11 40. Hughes Defendants have had actual knowledge of their infringement of  
12 the ’710 patent before the filing date of this Complaint through letters alleging such  
13 infringement, or at least have had actual knowledge of their infringement of the ’710  
14 patent since no later than the filing date of this Complaint.

15 41. On information and belief, Dish Defendants have had actual  
16 knowledge of their infringement of the ’710 patent before the filing date of this  
17 Complaint based on their marketing, sale, and distribution, among other activities,  
18 of Hughes Defendants’ satellite internet service and their relationship with Hughes  
19 Defendants (*see* Paragraphs 9, 10, 33). Dish Defendants at least have had actual  
20 knowledge of their infringement of the ’710 patent since no later than the filing  
21 date of this Complaint.

22 42. Notwithstanding Defendants’ actual notice of infringement,  
23 Defendants have continued, directly and/or through intermediaries, to manufacture,  
24 use, import, offer for sale, or sell the Accused Services and Products with  
25 knowledge of or willful blindness to the fact that their actions will induce others,  
26 including but not limited to their customers, partners, and/or end users, to infringe  
27 the ’710 patent. Defendants have induced and continue to induce others to infringe  
28 the ’710 patent in violation of 35 U.S.C. § 271 by encouraging and facilitating

1 others to perform actions that Defendants know to be acts of infringement of the  
2 '710 patent with intent that those performing the acts infringe the '710 patent.  
3 Upon information and belief, Defendants, directly and/or through intermediaries,  
4 advertise and distribute the Accused Services and Products, publish instruction  
5 materials, specifications and/or promotional literature describing the operation of  
6 the Accused Services and Products, and/or offer training and/or consulting services  
7 regarding the Accused Services and Products to their customers, partners, and/or  
8 end users. At least consumers, partners, and/or end users of these Accused Services  
9 and Products then directly or jointly infringe the '710 patent by making, using,  
10 selling, offering for sale, and/or importing into the United States, without authority,  
11 the Accused Services and Products.

12 43. Upon information and belief, Defendants know that the Accused  
13 Services and Products are especially made or especially adapted for use in the  
14 infringement of the '710 patent. The infringing components of these products are  
15 not staple articles or commodities of commerce suitable for substantial non-  
16 infringing use, and the infringing components of these products are a material part  
17 of the invention of the '710 patent. Accordingly, in violation of 35 U.S.C. § 271,  
18 Defendants are also contributing, directly and/or through intermediaries, to the  
19 direct infringement of the '710 patent by at least the customers, partners, and/or end  
20 users of these Accused Services and Products. The customers, partners, and/or end  
21 users of these Accused Services and Products directly infringe the '710 patent by  
22 making, using, selling, offering for sale, and/or importing into the United States,  
23 without authority, the Accused Services and Products.

24 44. As but one example of Hughes Defendants' contributory and/or  
25 induced infringement, Hughes Defendants explicitly encourage their customers to  
26 practice the methods disclosed and claimed in the '710 patent by using the Accused  
27 Services and Products. As detailed in Paragraphs 34 through 36, Hughes  
28 Defendants' website advertises its HN System and HX System, and provides

1 information and brochures regarding these systems. (*See* Exhibits J, K, L, M.)  
2 These webpages and brochures highlight Hughes Defendants' implementation of the  
3 DVB-S2 standard. On information and belief, through materials such as these, the  
4 Hughes Defendants actively encourage their consumers, partners, and/or end users  
5 to infringe the '710 patent through at least use of the HN System and HX System  
6 product lines, knowing those acts to be infringement of the '710 patent with intent  
7 that those performing the acts infringe the '710 patent.

8         45. As but one example of Dish Defendants' contributory and/or induced  
9 infringement, Dish Defendants explicitly encourage their customers to practice the  
10 methods disclosed and claimed in the '710 patent by using the Accused Services and  
11 Products. According to Dish Defendants' 2012 Annual Report (10-K), Dish  
12 Defendants lease to dishNET satellite internet subscribers the customer premise  
13 equipment. Dish Defendants also advertise, market, offer for sale, and sell to  
14 customers the Hopper set-top box on their website. On information and belief, the  
15 dishNET customer premise equipment and the Hopper set-top box implement the  
16 DVB-S2 standard. On information and belief, through providing this equipment,  
17 Dish Defendants actively encourage their consumers and end users to infringe the  
18 '710 patent through at least use of the equipment, knowing those acts to be  
19 infringement of the '710 patent with intent that those performing the acts infringe  
20 the '710 patent.

21         46. Defendants are not licensed or otherwise authorized to practice,  
22 contributorily practice and/or induce third parties to practice the claims of the '710  
23 patent.

24         47. By reason of Defendants' infringing activities, Caltech has suffered,  
25 and will continue to suffer, substantial damages.

26         48. Caltech is entitled to recover from Defendants the damages sustained as  
27 a result of Defendants' wrongful acts in an amount subject to proof at trial.

28



1 the '032 patent by making, using, selling, offering for sale, and/or importing into the  
2 United States, without authority, products, methods, equipment, and/or services that  
3 practice one or more claims of the '032 patent. These products, methods,  
4 equipment, and/or services include products that implement the DVB-S2 standard,  
5 including without limitation products in the HN System and HX System product  
6 lines, satellite internet product lines distributed under the dishNET brand, the  
7 Hopper set-top box, network and network services that employ these products,  
8 and/or marketing, consulting, and/or support services provided for these products  
9 and services (collectively, the "Accused Services and Products"). For example, at  
10 least Paragraphs 32 and 33 illustrate a limited number of examples of Defendants'  
11 direct infringement of the '032 patent. Defendants have infringed and are currently  
12 infringing literally and/or under the doctrine of equivalents.

13 54. On information and belief, in violation of 35 U.S.C. § 271, Defendants  
14 have infringed and are continuing to infringe the '032 patent by contributing to  
15 and/or actively inducing the infringement by others of the '032 patent by making,  
16 using, selling, offering for sale, and/or importing into the United States, without  
17 authority, products, methods, equipment, and/or services, including the Accused  
18 Services and Products, that practice one or more claims of the '032 patent.

19 55. Hughes Defendants have had actual knowledge of their infringement of  
20 the '032 patent before the filing date of this Complaint through letters alleging such  
21 infringement, or at least have had actual knowledge of their infringement of the '032  
22 patent since no later than the filing date of this Complaint.

23 56. On information and belief, Dish Defendants have had actual knowledge  
24 of their infringement of the '032 patent before the filing date of this Complaint  
25 based on their marketing, sale, and distribution, among other activities, of Hughes  
26 Defendants' satellite internet service and their relationship with Hughes Defendants  
27 (*see* Paragraphs 9, 10, 33). Dish Defendants at least have had actual knowledge of  
28 their infringement of the '032 patent since no later than the filing date of this

1 Complaint.

2 57. Notwithstanding Defendants' actual notice of infringement, Defendants  
3 have continued, directly and/or through intermediaries, to manufacture, use, import,  
4 offer for sale, or sell the Accused Services and Products with knowledge of or  
5 willful blindness to the fact that their actions will induce others, including but not  
6 limited to their customers, partners, and/or end users, to infringe the '032 patent.  
7 Defendants have induced and continue to induce others to infringe the '032 patent in  
8 violation of 35 U.S.C. § 271 by encouraging and facilitating others to perform  
9 actions that Defendants know to be acts of infringement of the '032 patent with  
10 intent that those performing the acts infringe the '032 patent. Upon information and  
11 belief, Defendants, directly and/or through intermediaries, advertise and distribute  
12 the Accused Services and Products, publish instruction materials, specifications  
13 and/or promotional literature describing the operation of the Accused Services and  
14 Products, and/or offer training and/or consulting services regarding the Accused  
15 Services and Products to their customers, partners, and/or end users. At least  
16 consumers, partners, and/or end users of these Accused Services and Products then  
17 directly or jointly infringe the '032 patent by making, using, selling, offering for  
18 sale, and/or importing into the United States, without authority, the Accused  
19 Services and Products.

20 58. Upon information and belief, Defendants know that the Accused  
21 Services and Products are especially made or especially adapted for use in the  
22 infringement of the '032 patent. The infringing components of these products are  
23 not staple articles or commodities of commerce suitable for substantial non-  
24 infringing use, and the infringing components of these products are a material part  
25 of the invention of the '032 patent. Accordingly, in violation of 35 U.S.C. § 271,  
26 Defendants are also contributing, directly and/or through intermediaries, to the  
27 direct infringement of the '032 patent by at least the customers, partners, and/or end  
28 users of these Accused Services and Products. The customers, partners, and/or end



1 users of these Accused Services and Products directly infringe the '032 patent by  
2 making, using, selling, offering for sale, and/or importing into the United States,  
3 without authority, the Accused Services and Products.

4 59. As but one example of Hughes Defendants' contributory and/or  
5 induced infringement, Hughes Defendants explicitly encourage their customers to  
6 practice the methods disclosed and claimed in the '032 patent by using the Accused  
7 Services and Products. As detailed in Paragraphs 34 through 36, Hughes  
8 Defendants' website advertises its HN System and HX System, and provides  
9 information and brochures regarding these systems. (*See* Exhibits J, K, L, M.)  
10 These webpages and brochures highlight Hughes Defendants' implementation of the  
11 DVB-S2 standard. On information and belief, through materials such as these, the  
12 Hughes Defendants actively encourage their consumers, partners, and/or end users  
13 to infringe the '032 patent through at least use of the HN System and HX System  
14 product lines, knowing those acts to be infringement of the '032 patent with intent  
15 that those performing the acts infringe the '032 patent.

16 60. As but one example of Dish Defendants' contributory and/or induced  
17 infringement, Dish Defendants explicitly encourage their customers to practice the  
18 methods disclosed and claimed in the '032 patent by using the Accused Services and  
19 Products. According to Dish Defendants' 2012 Annual Report (10-K), Dish  
20 Defendants lease to dishNET satellite internet subscribers the customer premise  
21 equipment. Dish Defendants also advertise, market, offer for sale, and sell to  
22 customers the Hopper set-top box on their website. On information and belief, the  
23 dishNET customer premise equipment and the Hopper set-top box implement the  
24 DVB-S2 standard. On information and belief, through providing this equipment,  
25 Dish Defendants actively encourage their consumers and end users to infringe the  
26 '032 patent through at least use of the equipment, knowing those acts to be  
27 infringement of the '032 patent with intent that those performing the acts infringe  
28 the '032 patent.

1           61. Defendants are not licensed or otherwise authorized to practice,  
2 contributorily practice and/or induce third parties to practice the claims of the '032  
3 patent.

4           62. By reason of Defendants' infringing activities, Caltech has suffered,  
5 and will continue to suffer, substantial damages.

6           63. Caltech is entitled to recover from Defendants the damages sustained as  
7 a result of Defendants' wrongful acts in an amount subject to proof at trial.

8           64. Defendants' continuing acts of infringement are irreparably harming  
9 and causing damage to Caltech, for which Caltech has no adequate remedy at law,  
10 and will continue to suffer such irreparable injury unless Defendants' continuing  
11 acts of infringement are enjoined by the Court. The hardships that an injunction  
12 would impose are less than those faced by Caltech should an injunction not issue.  
13 The public interest would be served by issuance of an injunction. Thus, Caltech is  
14 entitled to a preliminary and a permanent injunction against further infringement.

15           65. Hughes Defendants' infringement of the '032 patent has been and  
16 continues to be willful and deliberate, justifying a trebling of damages under 35  
17 U.S.C. § 284. Among other facts, Hughes Defendants have had knowledge of their  
18 infringement of the '032 patent before the filing date of this Complaint through  
19 letters alleging such infringement. Upon information and belief, Hughes  
20 Defendants' accused actions continued despite an objectively high likelihood that  
21 they constituted infringement of the '032 patent. Hughes Defendants either knew or  
22 should have known about their risk of infringing the '032 patent. Hughes  
23 Defendants' conduct despite this knowledge was made with both objective and  
24 subjective reckless disregard for the infringing nature of their activities as  
25 demonstrated by Hughes Defendants' knowledge regarding the claims of the '032  
26 patent.

27           66. Defendants' infringement of the '032 patent is exceptional and entitles  
28 Caltech to attorneys' fees and costs incurred in prosecuting this action under 35

1 U.S.C. § 285.

2 **COUNT III**

3 **Infringement of the '781 Patent**

4 67. Plaintiff re-alleges and incorporates by reference the allegations of the  
5 preceding paragraphs of this Complaint as if fully set forth herein.

6 68. On information and belief, in violation of 35 U.S.C. § 271, Defendants  
7 have infringed and are currently infringing, directly and/or through intermediaries,  
8 the '781 patent by using, without authority, products, methods, equipment, and/or  
9 services that practice one or more claims of the '781 patent. These products,  
10 methods, equipment, and/or services include products that implement the DVB-S2  
11 standard, including without limitation products in the HN System and HX System  
12 product lines, satellite internet product lines distributed under the dishNET brand,  
13 the Hopper set-top box, network and network services that employ these products,  
14 and/or marketing, consulting, and/or support services provided for these products  
15 and services (collectively, the "Accused Services and Products"). For example, at  
16 least Paragraphs 32 and 33 illustrate a limited number of examples of Defendants'  
17 direct infringement of the '781 patent. Defendants have infringed and are currently  
18 infringing literally and/or under the doctrine of equivalents.

19 69. On information and belief, in violation of 35 U.S.C. § 271, Defendants  
20 have infringed and are continuing to infringe the '781 patent by contributing to  
21 and/or actively inducing the infringement by others of the '781 patent by making,  
22 using, selling, offering for sale, and/or importing into the United States, without  
23 authority, products, methods, equipment, and/or services, including the Accused  
24 Services and Products, that practice one or more claims of the '781 patent.

25 70. On information and belief, Hughes Defendants have had actual  
26 knowledge of their infringement of the '781 patent, the subject matter of the '781  
27 patent, and/or the invention of the '781 patent before the filing date of this  
28 Complaint. On information and belief, Hughes Defendants also had knowledge of

1 the application that led to the '781 patent before the filing date of this Complaint.  
2 Hughes Defendants at least have had actual knowledge of their infringement of the  
3 '781 patent since no later than the filing date of this Complaint.

4 71. On information and belief, Dish Defendants have had actual knowledge  
5 of their infringement of the '781 patent before the filing date of this Complaint  
6 based on their marketing, sale, and distribution, among other activities, of Hughes  
7 Defendants' satellite internet service and their relationship with Hughes Defendants  
8 (*see* Paragraphs 9, 10, 33). Dish Defendants at least have had actual knowledge of  
9 their infringement of the '781 patent since no later than the filing date of this  
10 Complaint.

11 72. Notwithstanding Defendants' actual notice of infringement, Defendants  
12 have continued, directly and/or through intermediaries, to manufacture, use, import,  
13 offer for sale, or sell the Accused Services and Products with knowledge of or  
14 willful blindness to the fact that their actions will induce others, including but not  
15 limited to their customers, partners, and/or end users, to infringe the '781 patent.  
16 Defendants have induced and continue to induce others to infringe the '781 patent in  
17 violation of 35 U.S.C. § 271 by encouraging and facilitating others to perform  
18 actions that Defendants know to be acts of infringement of the '781 patent with  
19 intent that those performing the acts infringe the '781 patent. Upon information and  
20 belief, Defendants, directly and/or through intermediaries, advertise and distribute  
21 the Accused Services and Products, publish instruction materials, specifications  
22 and/or promotional literature describing the operation of the Accused Services and  
23 Products, and/or offer training and/or consulting services regarding the Accused  
24 Services and Products to their customers, partners, and/or end users. At least  
25 consumers, partners, and/or end users of these Accused Services and Products then  
26 directly or jointly infringe the '781 patent by making, using, selling, offering for  
27 sale, and/or importing into the United States, without authority, the Accused  
28 Services and Products.

1           73. Upon information and belief, Defendants know that the Accused  
2 Services and Products are especially made or especially adapted for use in the  
3 infringement of the '781 patent. The infringing components of these products are  
4 not staple articles or commodities of commerce suitable for substantial non-  
5 infringing use, and the infringing components of these products are a material part  
6 of the invention of the '781 patent. Accordingly, in violation of 35 U.S.C. § 271,  
7 Defendants are also contributing, directly and/or through intermediaries, to the  
8 direct infringement of the '781 patent by at least the customers, partners, and/or end  
9 users of these Accused Services and Products. The customers, partners, and/or end  
10 users of these Accused Services and Products directly infringe the '781 patent by  
11 making, using, selling, offering for sale, and/or importing into the United States,  
12 without authority, the Accused Services and Products.

13           74. As but one example of Hughes Defendants' contributory and/or  
14 induced infringement, Hughes Defendants explicitly encourage their customers to  
15 practice the methods disclosed and claimed in the '781 patent by using the Accused  
16 Services and Products. As detailed in Paragraphs 34 through 36, Hughes  
17 Defendants' website advertises its HN System and HX System, and provides  
18 information and brochures regarding these systems. (*See Exhibits J, K, L, M.*)  
19 These webpages and brochures highlight Hughes Defendants' implementation of the  
20 DVB-S2 standard. On information and belief, through materials such as these, the  
21 Hughes Defendants actively encourage their consumers, partners, and/or end users  
22 to infringe the '781 patent through at least use of the HN System and HX System  
23 product lines, knowing those acts to be infringement of the '781 patent with intent  
24 that those performing the acts infringe the '781 patent.

25           75. As but one example of Dish Defendants' contributory and/or induced  
26 infringement, Dish Defendants explicitly encourage their customers to practice the  
27 methods disclosed and claimed in the '781 patent by using the Accused Services and  
28 Products. According to Dish Defendants' 2012 Annual Report (10-K), Dish

1 Defendants lease to dishNET satellite internet subscribers the customer premise  
2 equipment. Dish Defendants also advertise, market, offer for sale, and sell to  
3 customers the Hopper set-top box on their website. On information and belief, the  
4 dishNET customer premise equipment and the Hopper set-top box implement the  
5 DVB-S2 standard. On information and belief, through providing this equipment,  
6 Dish Defendants actively encourage their consumers and end users to infringe the  
7 '781 patent through at least use of the equipment, knowing those acts to be  
8 infringement of the '781 patent with intent that those performing the acts infringe  
9 the '781 patent.

10 76. Defendants are not licensed or otherwise authorized to practice,  
11 contributorily practice and/or induce third parties to practice the claims of the '781  
12 patent.

13 77. By reason of Defendants' infringing activities, Caltech has suffered,  
14 and will continue to suffer, substantial damages.

15 78. Caltech is entitled to recover from Defendants the damages sustained as  
16 a result of Defendants' wrongful acts in an amount subject to proof at trial.

17 79. Defendants' continuing acts of infringement are irreparably harming  
18 and causing damage to Caltech, for which Caltech has no adequate remedy at law,  
19 and will continue to suffer such irreparable injury unless Defendants' continuing  
20 acts of infringement are enjoined by the Court. The hardships that an injunction  
21 would impose are less than those faced by Caltech should an injunction not issue.  
22 The public interest would be served by issuance of an injunction. Thus, Caltech is  
23 entitled to a preliminary and a permanent injunction against further infringement.

24 80. Hughes Defendants' infringement of the '781 patent has been and  
25 continues to be willful and deliberate, justifying a trebling of damages under 35  
26 U.S.C. § 284. Among other facts, on information and belief, Hughes Defendants  
27 have had knowledge of their infringement of the '781 patent, the subject matter of  
28 the '781 patent, and/or the invention of the '781 patent before the filing date of this



1 Complaint. Upon information and belief, Hughes Defendants' accused actions  
2 continued despite an objectively high likelihood that they constituted infringement  
3 of the '781 patent. Hughes Defendants either knew or should have known about  
4 their risk of infringing the '781 patent. Hughes Defendants' conduct despite this  
5 knowledge was made with both objective and subjective reckless disregard for the  
6 infringing nature of their activities as demonstrated by Hughes Defendants'  
7 knowledge regarding the claims of the '781 patent.

8 81. Defendants' infringement of the '781 patent is exceptional and entitles  
9 Caltech to attorneys' fees and costs incurred in prosecuting this action under 35  
10 U.S.C. § 285.

11 **COUNT IV**

12 **Infringement of the '833 Patent**

13 82. Plaintiff re-alleges and incorporates by reference the allegations of the  
14 preceding paragraphs of this Complaint as if fully set forth herein.

15 83. On information and belief, in violation of 35 U.S.C. § 271, Defendants  
16 have infringed and are currently infringing, directly and/or through intermediaries,  
17 the '833 patent by making, using, selling, offering for sale, and/or importing into the  
18 United States, without authority, products, methods, equipment, and/or services that  
19 practice one or more claims of the '833 patent. These products, methods,  
20 equipment, and/or services include products that implement the DVB-S2 standard,  
21 including without limitation products in the HN System and HX System product  
22 lines, satellite internet product lines distributed under the dishNET brand, the  
23 Hopper set-top box, network and network services that employ these products,  
24 and/or marketing, consulting, and/or support services provided for these products  
25 and services (collectively, the "Accused Services and Products"). For example, at  
26 least Paragraphs 32 and 33 illustrate a limited number of examples of Defendants'  
27 direct infringement of the '833 patent. Defendants have infringed and are currently  
28 infringing literally and/or under the doctrine of equivalents.

1           84. On information and belief, in violation of 35 U.S.C. § 271, Defendants  
2 have infringed and are continuing to infringe the '833 patent by contributing to  
3 and/or actively inducing the infringement by others of the '833 patent by making,  
4 using, selling, offering for sale, and/or importing into the United States, without  
5 authority, products, methods, equipment, and/or services, including the Accused  
6 Services and Products, that practice one or more claims of the '833 patent.

7           85. On information and belief, Hughes Defendants have had actual  
8 knowledge of their infringement of the '833 patent, the subject matter of the '833  
9 patent, and/or the invention of the '833 patent before the filing date of this  
10 Complaint. On information and belief, Hughes Defendants also had knowledge of  
11 the application that led to the '833 patent before the filing date of this Complaint.  
12 Hughes Defendants at least have had actual knowledge of their infringement of the  
13 '833 patent since no later than the filing date of this Complaint.

14           86. On information and belief, Dish Defendants have had actual knowledge  
15 of their infringement of the '833 patent before the filing date of this Complaint  
16 based on their marketing, sale, and distribution, among other activities, of Hughes  
17 Defendants' satellite internet service and their relationship with Hughes Defendants  
18 (*see* Paragraphs 9, 10, 33). Dish Defendants at least have had actual knowledge of  
19 their infringement of the '833 patent since no later than the filing date of this  
20 Complaint.

21           87. Notwithstanding Defendants' actual notice of infringement, Defendants  
22 have continued, directly and/or through intermediaries, to manufacture, use, import,  
23 offer for sale, or sell the Accused Services and Products with knowledge of or  
24 willful blindness to the fact that their actions will induce others, including but not  
25 limited to their customers, partners, and/or end users, to infringe the '833 patent.  
26 Defendants have induced and continue to induce others to infringe the '833 patent in  
27 violation of 35 U.S.C. § 271 by encouraging and facilitating others to perform  
28 actions that Defendants know to be acts of infringement of the '833 patent with

1 intent that those performing the acts infringe the '833 patent. Upon information and  
2 belief, Defendants, directly and/or through intermediaries, advertise and distribute  
3 the Accused Services and Products, publish instruction materials, specifications  
4 and/or promotional literature describing the operation of the Accused Services and  
5 Products, and/or offer training and/or consulting services regarding the Accused  
6 Services and Products to their customers, partners, and/or end users. At least  
7 consumers, partners, and/or end users of these Accused Services and Products then  
8 directly or jointly infringe the '833 patent by making, using, selling, offering for  
9 sale, and/or importing into the United States, without authority, the Accused  
10 Services and Products.

11 88. Upon information and belief, Defendants know that the Accused  
12 Services and Products are especially made or especially adapted for use in the  
13 infringement of the '833 patent. The infringing components of these products are  
14 not staple articles or commodities of commerce suitable for substantial non-  
15 infringing use, and the infringing components of these products are a material part  
16 of the invention of the '833 patent. Accordingly, in violation of 35 U.S.C. § 271,  
17 Defendants are also contributing, directly and/or through intermediaries, to the  
18 direct infringement of the '833 patent by at least the customers, partners, and/or end  
19 users of these Accused Services and Products. The customers, partners, and/or end  
20 users of these Accused Services and Products directly infringe the '833 patent by  
21 making, using, selling, offering for sale, and/or importing into the United States,  
22 without authority, the Accused Services and Products.

23 89. As but one example of Hughes Defendants' contributory and/or  
24 induced infringement, Hughes Defendants explicitly encourage their customers to  
25 practice the methods disclosed and claimed in the '833 patent by using the Accused  
26 Services and Products. As detailed in Paragraphs 34 through 36, Hughes  
27 Defendants' website advertises its HN System and HX System, and provides  
28 information and brochures regarding these systems. (*See Exhibits J, K, L, M.*)

1 These webpages and brochures highlight Hughes Defendants' implementation of the  
2 DVB-S2 standard. On information and belief, through materials such as these, the  
3 Hughes Defendants actively encourage their consumers, partners, and/or end users  
4 to infringe the '833 patent through at least use of the HN System and HX System  
5 product lines, knowing those acts to be infringement of the '833 patent with intent  
6 that those performing the acts infringe the '833 patent.

7 90. As but one example of Dish Defendants' contributory and/or induced  
8 infringement, Dish Defendants explicitly encourage their customers to practice the  
9 methods disclosed and claimed in the '833 patent by using the Accused Services and  
10 Products. According to Dish Defendants' 2012 Annual Report (10-K), Dish  
11 Defendants lease to dishNET satellite internet subscribers the customer premise  
12 equipment. Dish Defendants also advertise, market, offer for sale, and sell to  
13 customers the Hopper set-top box on their website. On information and belief, the  
14 dishNET customer premise equipment and the Hopper set-top box implement the  
15 DVB-S2 standard. On information and belief, through providing this equipment,  
16 Dish Defendants actively encourage their consumers and end users to infringe the  
17 '833 patent through at least use of the equipment, knowing those acts to be  
18 infringement of the '833 patent with intent that those performing the acts infringe  
19 the '833 patent.

20 91. Defendants are not licensed or otherwise authorized to practice,  
21 contributorily practice and/or induce third parties to practice the claims of the '833  
22 patent.

23 92. By reason of Defendants' infringing activities, Caltech has suffered,  
24 and will continue to suffer, substantial damages.

25 93. Caltech is entitled to recover from Defendants the damages sustained as  
26 a result of Defendants' wrongful acts in an amount subject to proof at trial.

27 94. Defendants' continuing acts of infringement are irreparably harming  
28 and causing damage to Caltech, for which Caltech has no adequate remedy at law,

1 and will continue to suffer such irreparable injury unless Defendants' continuing  
2 acts of infringement are enjoined by the Court. The hardships that an injunction  
3 would impose are less than those faced by Caltech should an injunction not issue.  
4 The public interest would be served by issuance of an injunction. Thus, Caltech is  
5 entitled to a preliminary and a permanent injunction against further infringement.

6 95. Hughes Defendants' infringement of the '833 patent has been and  
7 continues to be willful and deliberate, justifying a trebling of damages under 35  
8 U.S.C. § 284. Among other facts, on information and belief, Hughes Defendants  
9 have had knowledge of their infringement of the '833 patent, the subject matter of  
10 the '833 patent, and/or the invention of the '833 patent before the filing date of this  
11 Complaint. Upon information and belief, Hughes Defendants' accused actions  
12 continued despite an objectively high likelihood that they constituted infringement  
13 of the '833 patent. Hughes Defendants either knew or should have known about  
14 their risk of infringing the '833 patent. Hughes Defendants' conduct despite this  
15 knowledge was made with both objective and subjective reckless disregard for the  
16 infringing nature of their activities as demonstrated by Hughes Defendants'  
17 knowledge regarding the claims of the '833 patent.

18 96. Defendants' infringement of the '833 patent is exceptional and entitles  
19 Caltech to attorneys' fees and costs incurred in prosecuting this action under 35  
20 U.S.C. § 285.

21 **PRAYER FOR RELIEF**

22 WHEREFORE, Plaintiff respectfully prays for the following relief:

23 (a) A judgment that Defendants have infringed each and every one of the  
24 Asserted Patents;

25 (b) A preliminary and permanent injunction against Defendants, its  
26 respective officers, agents, servants, employees, attorneys, parent and subsidiary  
27 corporations, assigns and successors in interest, and those persons in active concert  
28 or participation with them, enjoining them from infringement, inducement of

1 infringement, and contributory infringement of each and every one of the Asserted  
2 Patents, including but not limited to an injunction against making, using, selling,  
3 and/or offering for sale within the United States, and/or importing into the United  
4 States, any products, methods, equipment and/or services that infringe the Asserted  
5 Patents;

6 (c) Damages adequate to compensate Caltech for Defendants' infringement  
7 of the Asserted Patents pursuant to 35 U.S.C. § 284;

8 (d) Prejudgment interest;

9 (e) Post-judgment interest;

10 (f) A judgment holding Hughes Defendants' infringement of the Asserted  
11 Patents to be willful, and a trebling of damages pursuant to 35 U.S.C. § 284;

12 (g) A declaration that this Action is exceptional pursuant to 35 U.S.C.  
13 § 285, and an award to Caltech of its attorneys' fees, costs and expenses incurred in  
14 connection with this Action; and

15 (h) Such other relief as the Court deems just and equitable.

16

17

18 DATED: March 6, 2014

Respectfully submitted,

19

QUINN EMANUEL URQUHART &  
SULLIVAN, LLP

20

21

22

By /s/ James R. Asperger

23

James R. Asperger  
*Attorneys for Plaintiff California Institute  
of Technology*

24

25

26

27

28



1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28

**DEMAND FOR JURY TRIAL**

Pursuant to Rule 38 of the Federal Rules of Civil Procedure and Local Rule 38-1 of this Court, Plaintiff hereby demands a trial by jury as to all issues so triable.

DATED: March 6, 2014

Respectfully submitted,

QUINN EMANUEL URQUHART &  
SULLIVAN, LLP

By /s/ James R. Asperger  
James R. Asperger  
*Attorneys for Plaintiff California Institute  
of Technology*

**PROOF OF SERVICE**

I am employed in the County of Los Angeles, State of California. I am over the age of eighteen years and not a party to the within action; my business address is 865 South Figueroa Street, 10th Floor, Los Angeles, California 90017-2543.

On March 6, 2014, I served true copies of the following document(s) described as **AMENDED COMPLAINT FOR PATENT INFRINGEMENT** on the interested parties in this action as follows:

David C. Marcus  
david.marcus@wilmerhale.com  
Matthew J. Hawkinson  
Matthew.hawkinson@wilmerhale.com  
Aaron Thompson  
aaron.thompson@wilmerhale.com  
WILMER CUTLER PICKERING HALE AND DORR LLP  
350 South Grand Avenue, Suite 2100  
Los Angeles, CA 90071

*Attorneys for  
Defendants  
and  
Counterclaim  
-Plaintiffs*

William G. McElwain (*pro hac vice*)  
william.mcelwain@wilmerhale.com  
WILMER CUTLER PICKERING HALE AND DORR LLP  
1875 Pennsylvania Avenue NW  
Washington, DC 20006

William F. Lee (*pro hac vice*)  
william.lee@wilmerhale.com  
WILMER CUTLER PICKERING HALE AND DORR LLP  
60 State Street  
Boston, Massachusetts. 02109

**BY ELECTRONIC MAIL TRANSMISSION:** By electronic mail transmission from angeldelira@quinnemanuel.com on March 6, 2014, by transmitting a PDF format copy of such document(s) to each such person at the e mail address listed below their address(es). The document(s) was/were transmitted by electronic transmission and such transmission was reported as complete and without error.

**BY FEDEX:** I deposited such document(s) in a box or other facility regularly maintained by FedEx, or delivered such document(s) to a courier or driver authorized by FedEx to receive documents, in sealed envelope(s) or package(s) designated by FedEx with delivery fees paid or provided for, addressed to the person(s) being served.

I declare that I am employed in the office of a member of the bar of this Court at whose direction the service was made.

Executed on March 6, 2014, at Los Angeles, California.

  
\_\_\_\_\_  
Angel de Lira



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)

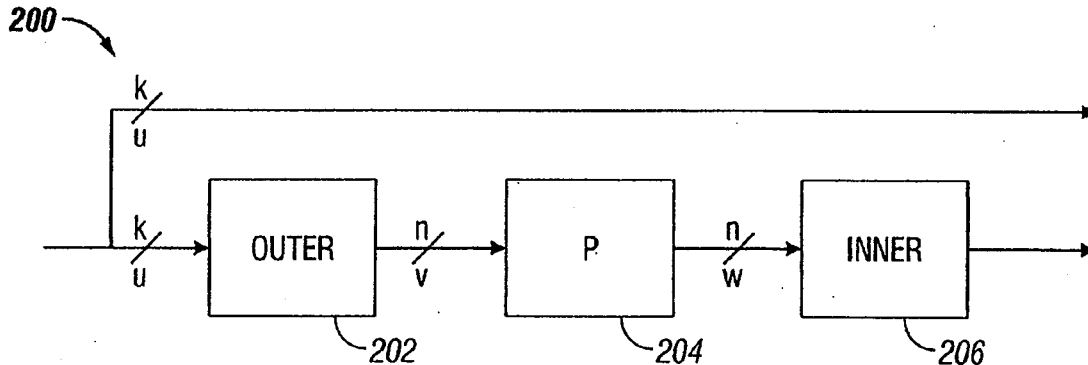
*Primary Examiner*—Dac V. Ha

(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 36<sup>th</sup> 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," 2<sup>nd</sup> 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

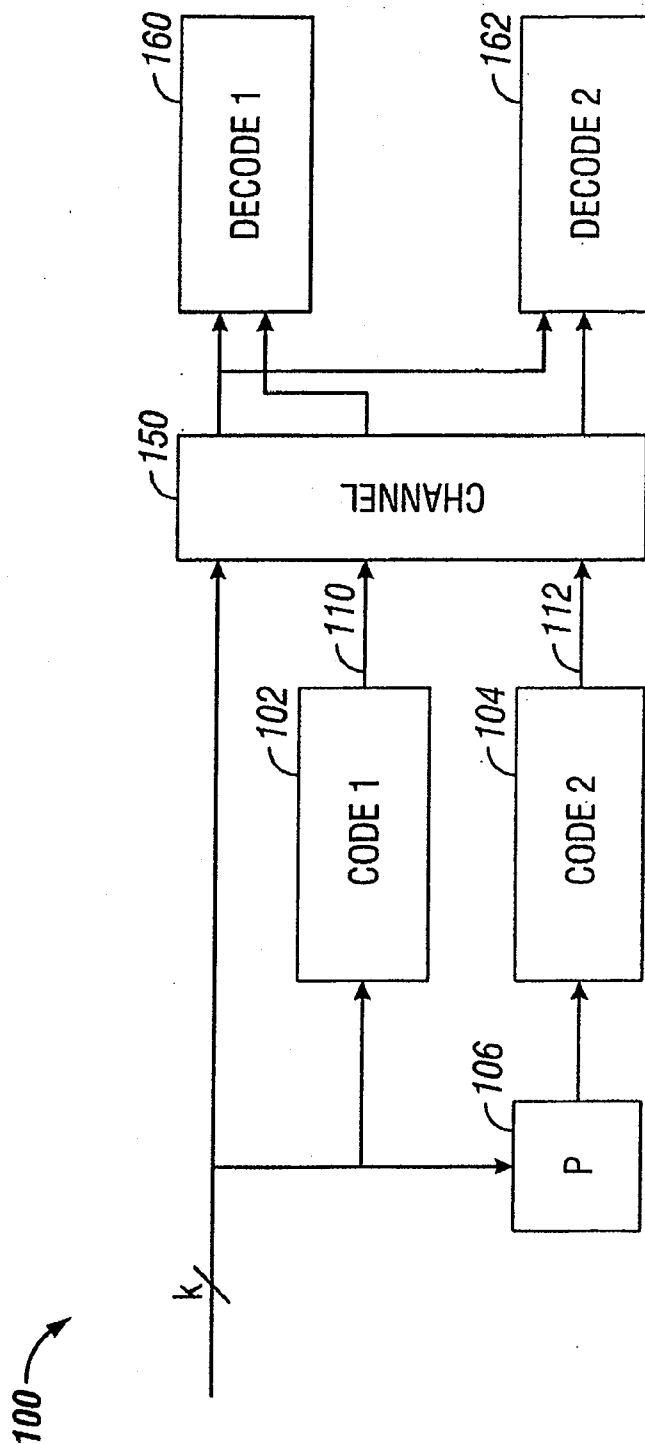


FIG. 1  
(Prior Art)

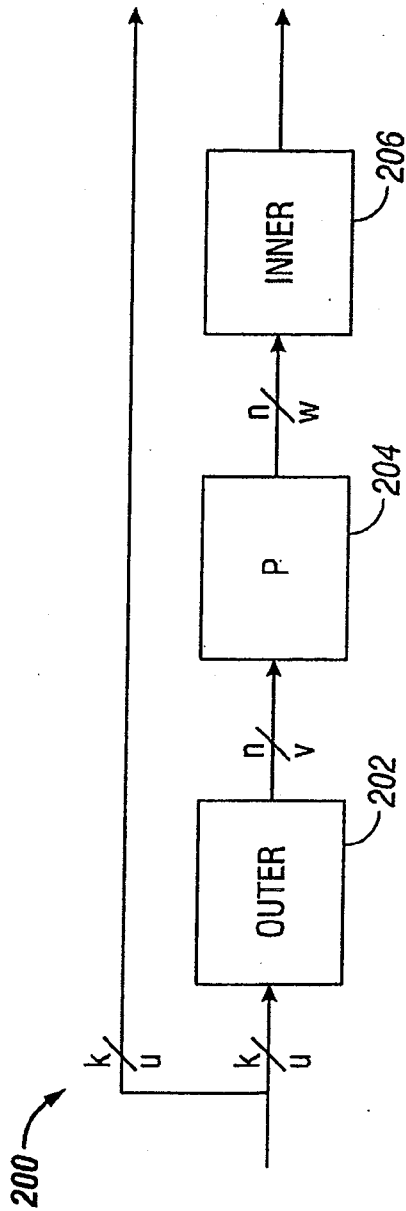


FIG. 2

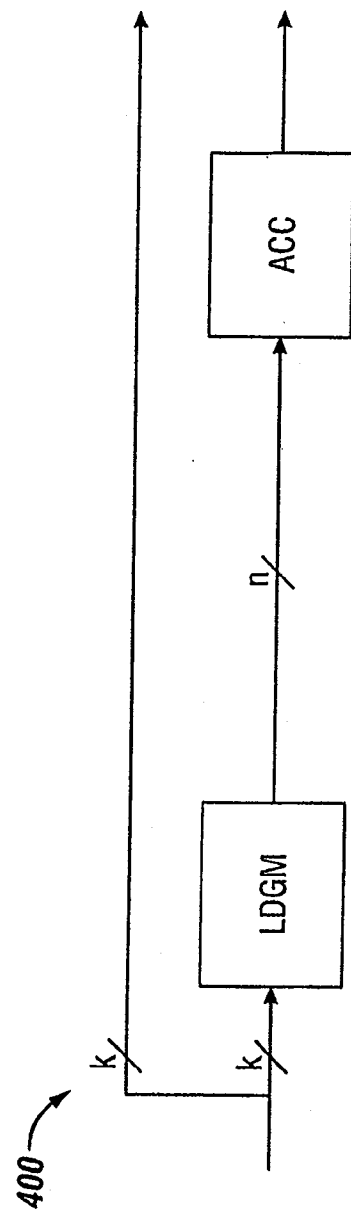


FIG. 4



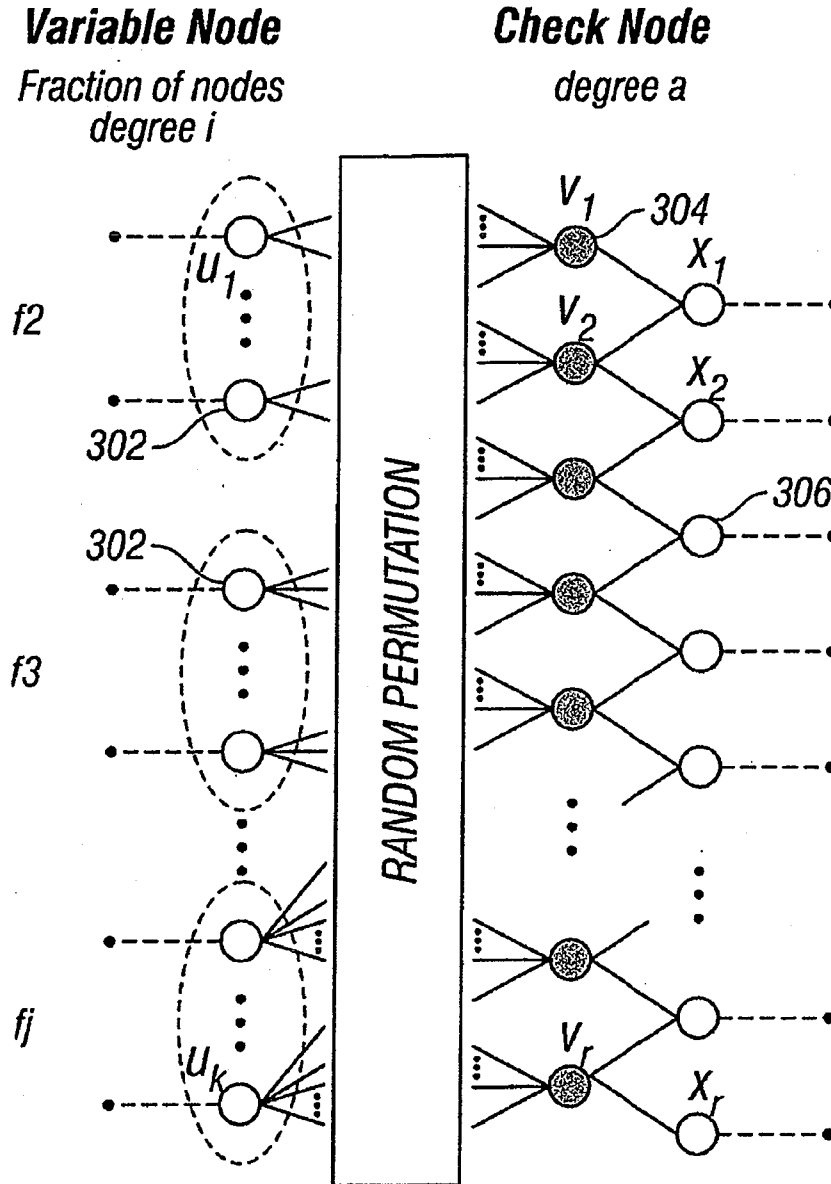


FIG. 3

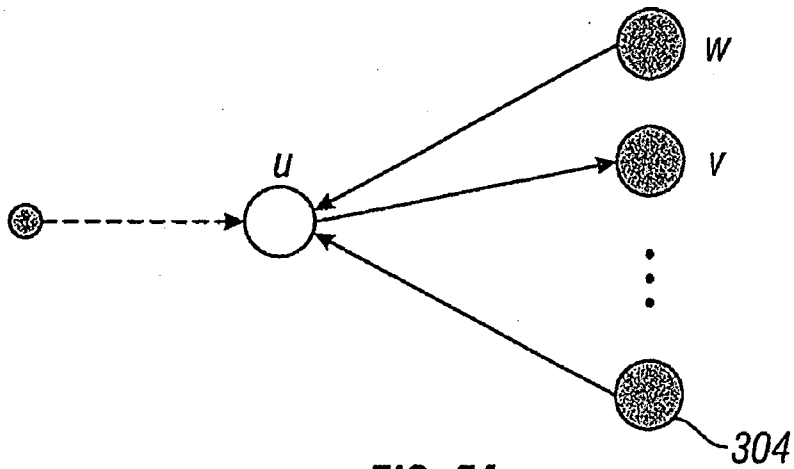


FIG. 5A

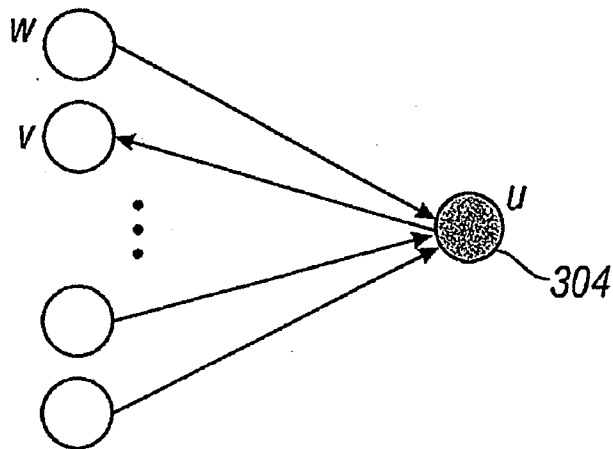


FIG. 5B

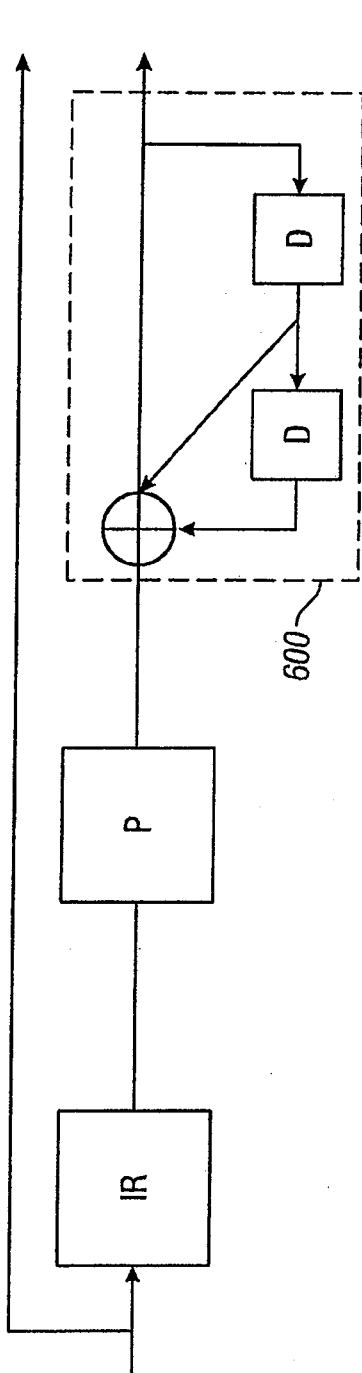


FIG. 6

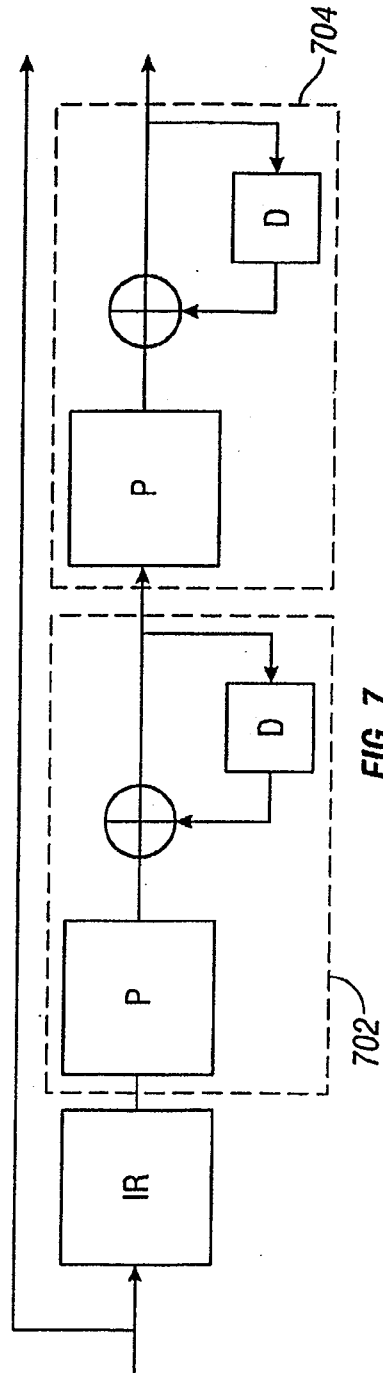


FIG. 7

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  $v = T_o u$ , 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 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) code 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 the permutation box are  $(v_1, \dots, v_{ra})$ , then we have the recursive formula

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

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  $\lambda_i$  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  $\rho_i$  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  $(\lambda, \rho)$  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
$\lambda_2$	0.139025	0.078194	0.054485
$\lambda_3$	0.2221555	0.128085	0.104315
$\lambda_5$		0.160813	
$\lambda_6$	0.638820	0.036178	0.126755
$\lambda_{10}$			0.229816
$\lambda_{11}$			0.016484
$\lambda_{12}$		0.108828	
$\lambda_{13}$		0.487902	
$\lambda_{14}$			
$\lambda_{16}$			
$\lambda_{27}$			0.450302
$\lambda_{28}$			0.017842
Rate	0.333364	0.333223	0.333218
$\sigma_{GA}$	1.1840	1.2415	1.2615
$\sigma^*$	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 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

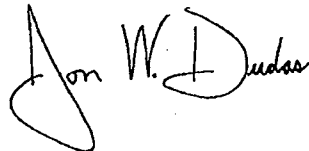
It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

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



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



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: **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 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	Cheunnakeshu 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)

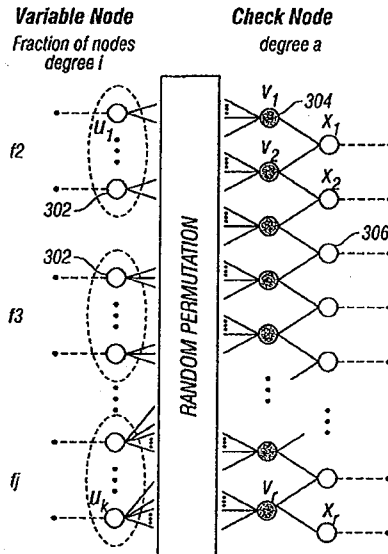
*Primary Examiner*—Dac V. Ha

(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 36<sup>th</sup> 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," MILCOM 95, 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," 2<sup>nd</sup> 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

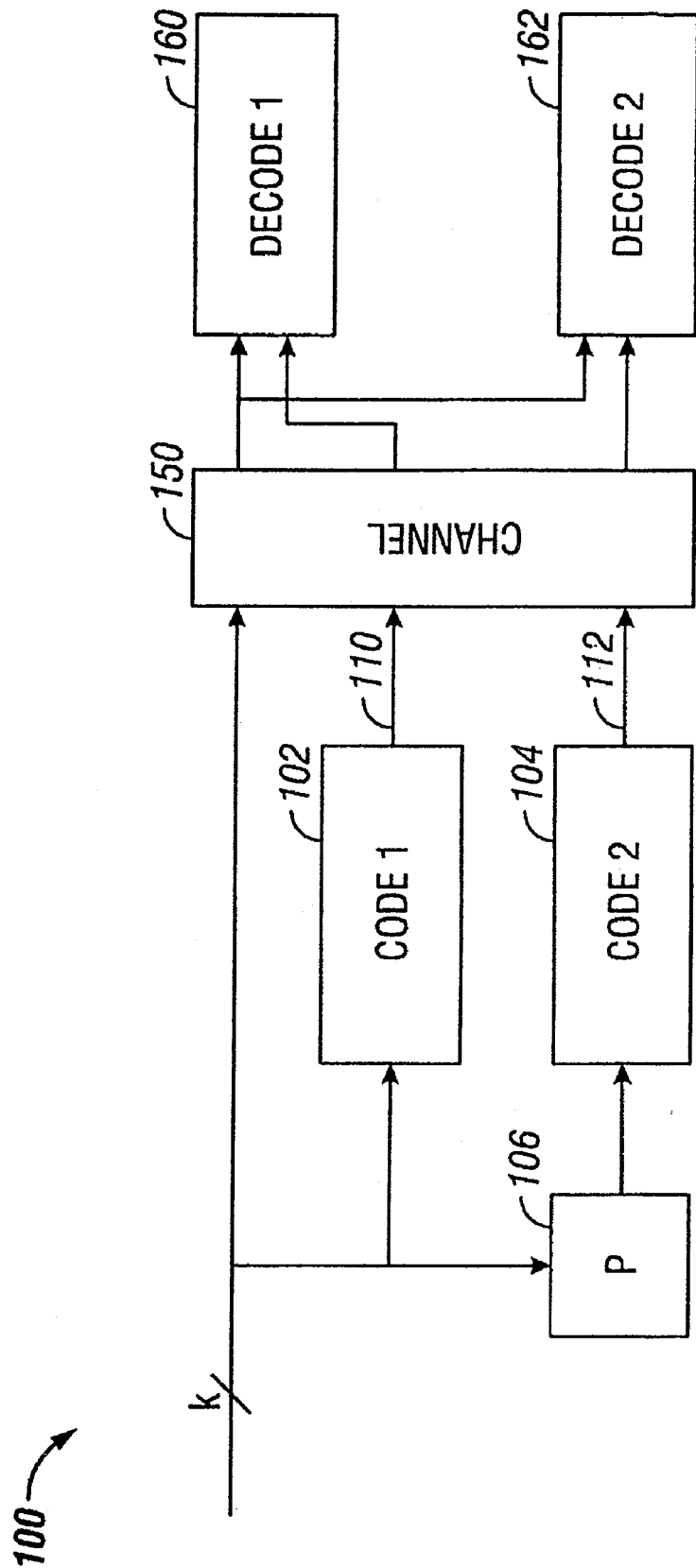


FIG. 1  
(Prior Art)

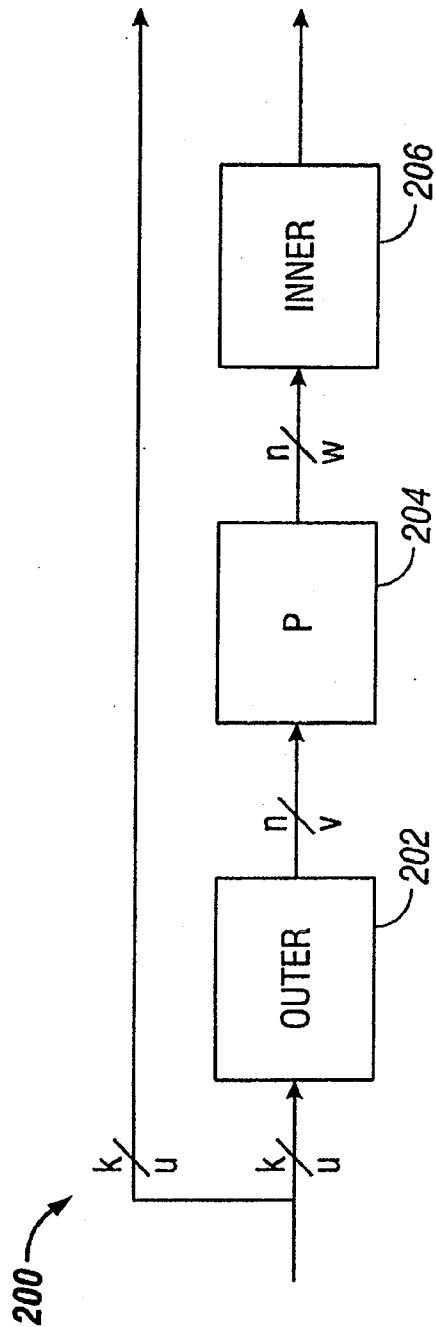


FIG. 2

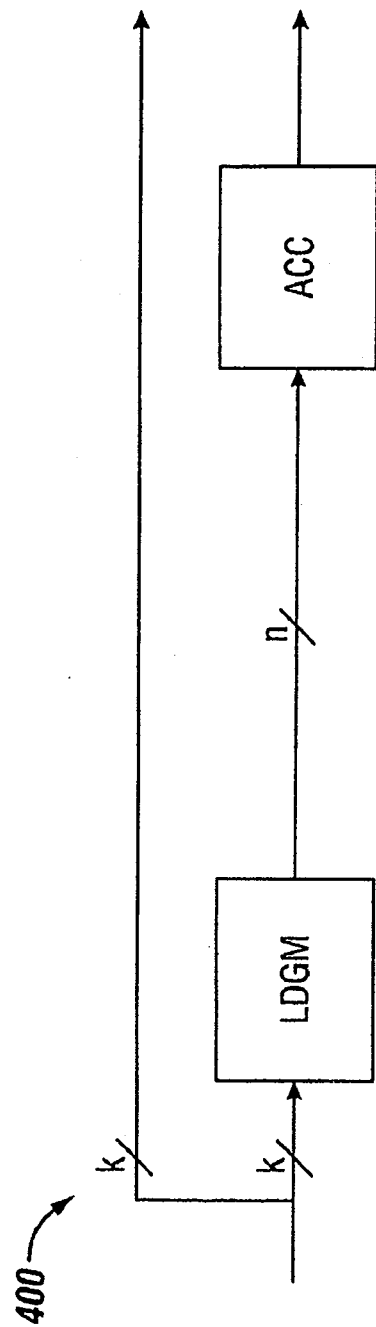
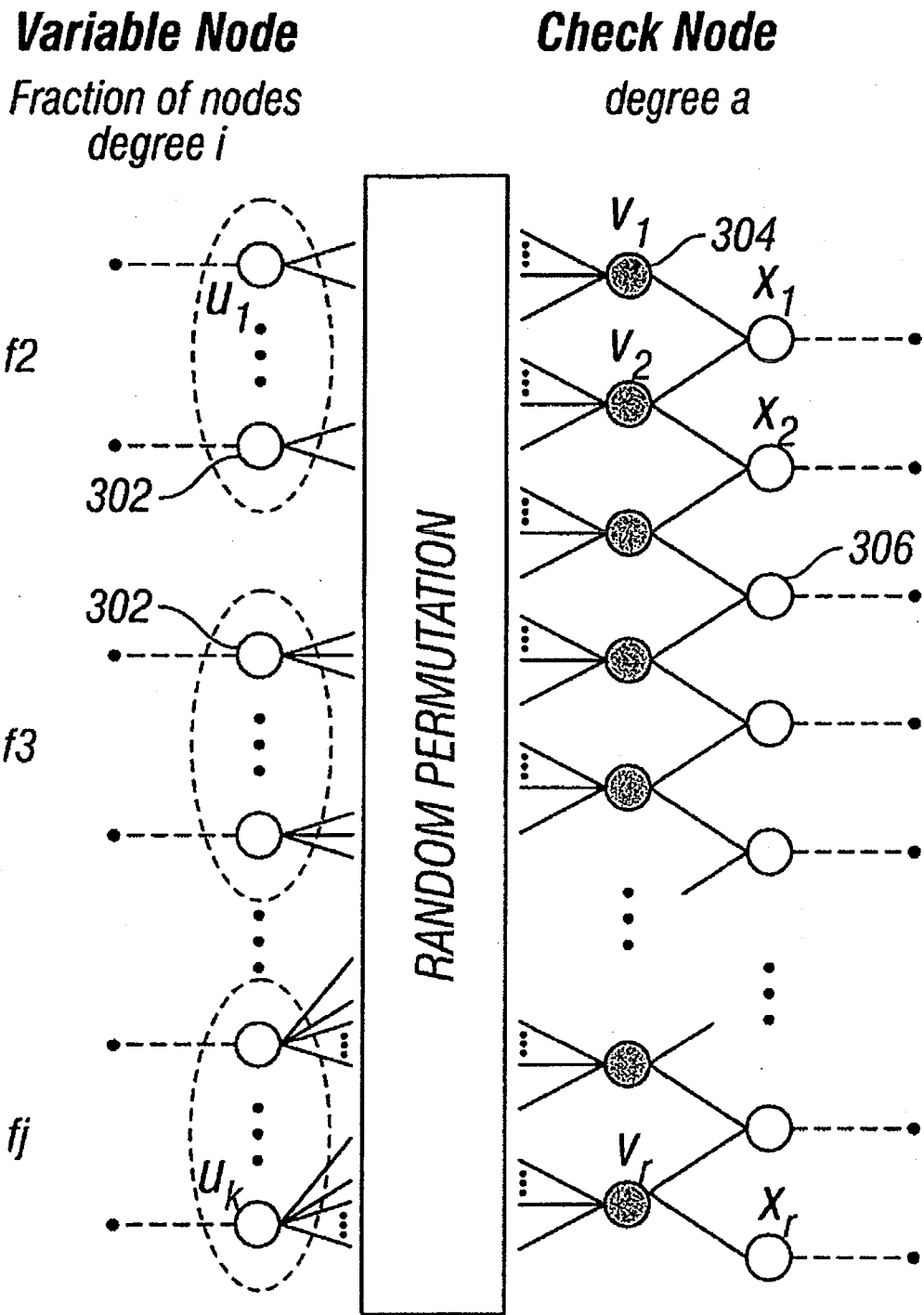


FIG. 4





**FIG. 3**

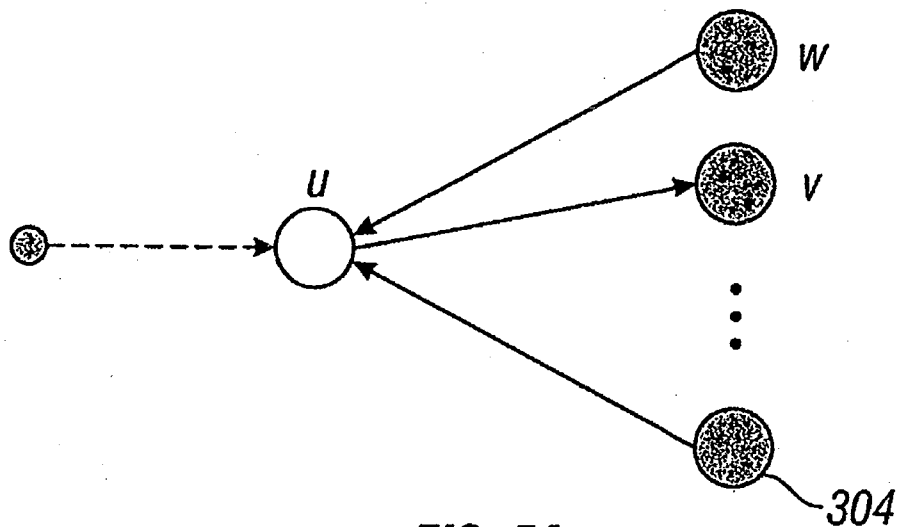


FIG. 5A

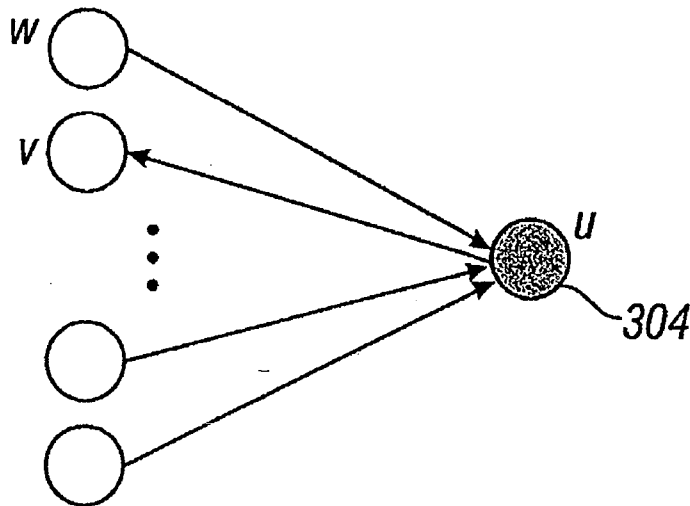


FIG. 5B

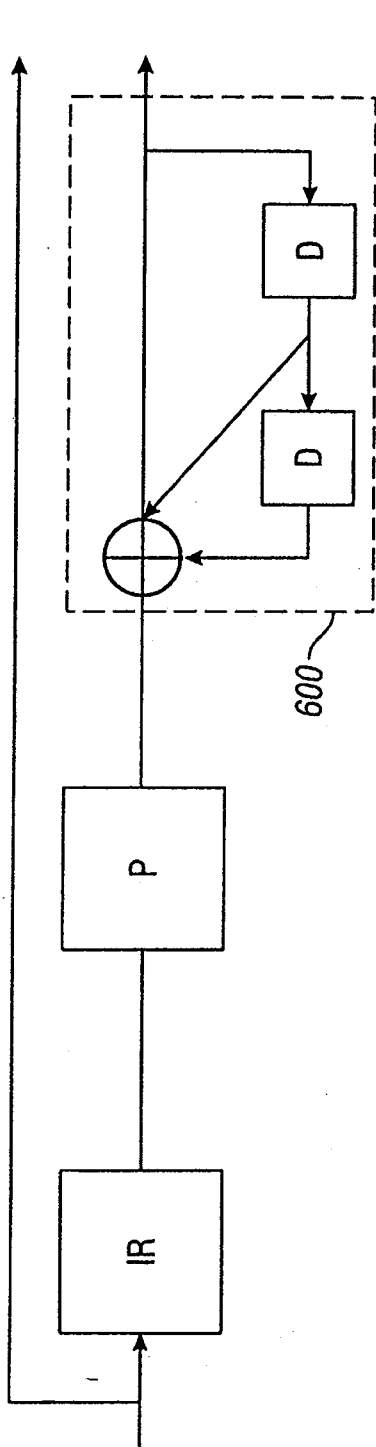


FIG. 6

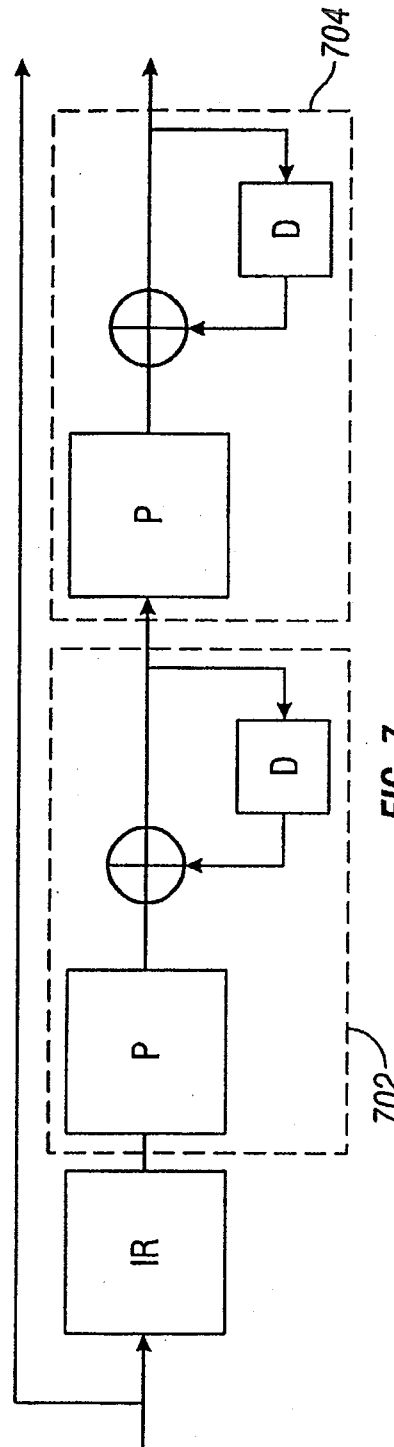


FIG. 7

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

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

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

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

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

Two types of IRA codes are represented in FIG. 3, a 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

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

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  $\lambda_i$  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  $\rho_i$  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  $(\lambda, \rho)$  is called a degree distribution. For  $L(x) = \sum_i f_i x_i$ ,

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

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

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

$$m(u \rightarrow v) = \sum_{w \in \mathcal{N}(u) \setminus v} m(w \rightarrow u) + m_0(u)$$

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

$$\tanh \frac{m(u \rightarrow v)}{2} = \prod_{w \in \mathcal{N}(u) \setminus v} \tanh \frac{m(w \rightarrow u)}{2}$$

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
$\lambda_2$	0.139025	0.078194	0.054485
$\lambda_3$	0.2221555	0.128085	0.104315
$\lambda_5$		0.160813	
$\lambda_6$	0.638820	0.036178	0.126755
$\lambda_{10}$			0.229816
$\lambda_{11}$			0.016484
$\lambda_{12}$		0.108828	
$\lambda_{13}$		0.487902	
$\lambda_{14}$			
$\lambda_{16}$			
$\lambda_{27}$			0.450302
$\lambda_{28}$			0.017842
Rate	0.333364	0.333223	0.333218
$\sigma_{GA}$	1.1840	1.2415	1.2615
$\sigma^*$	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 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 the 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

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

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.

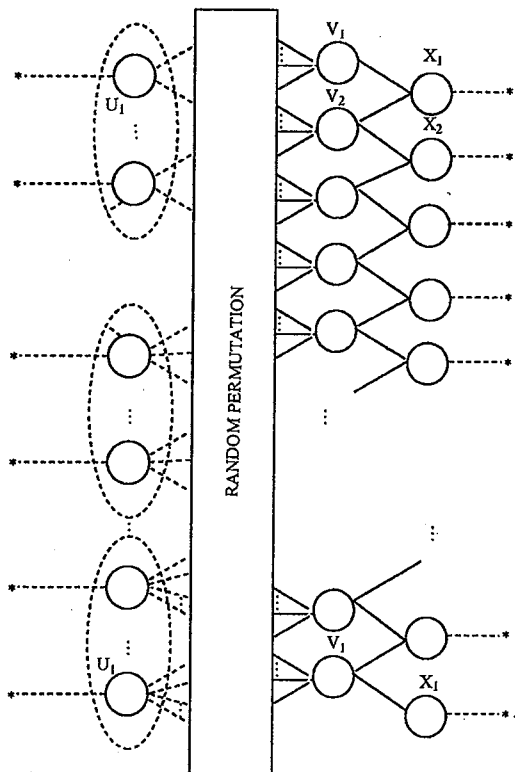
8. The method of claim 1, further comprising transmitting the sequence of parity bits.

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.

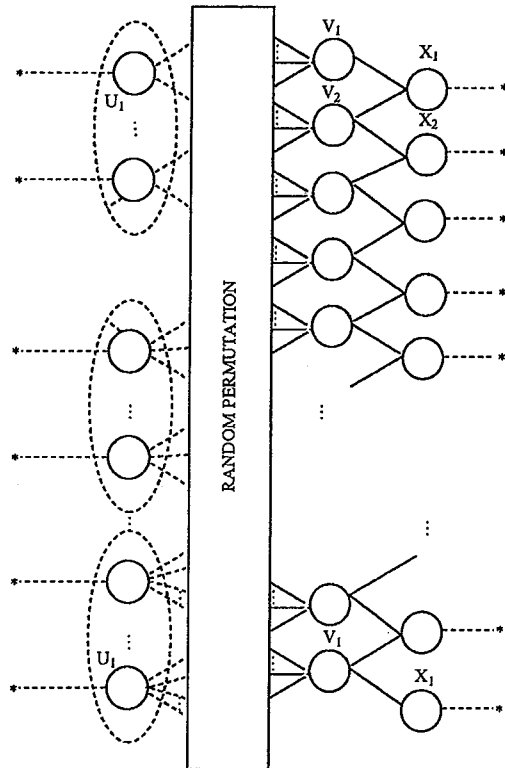
16. The device of claim 14, wherein the recursive convolutional coder has a transfer function of  $1/(1+D)$ .

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 "California" and insert --California--, therefor.

Claim 11, Column 9, line 28, delete "V<sub>1</sub>" and insert --V<sub>r</sub>--, therefor.

Claim 11, Column 9, line 29, delete "U<sub>1</sub>" and insert --U<sub>k</sub>--, therefor.

Claim 11, Column 9, line 29, delete "X<sub>1</sub>" and insert --X<sub>r</sub>--, therefor.

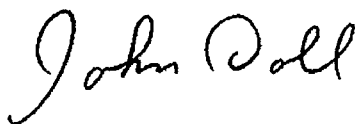
Claim 18, Column 10, line 35, delete "V<sub>1</sub>" and insert --V<sub>r</sub>--, therefor.

Claim 18, Column 10, line 36, delete "U<sub>1</sub>" and insert --U<sub>k</sub>--, therefor.

Claim 18, Column 10, line 37, delete "X<sub>1</sub>" and insert --X<sub>r</sub>--, 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

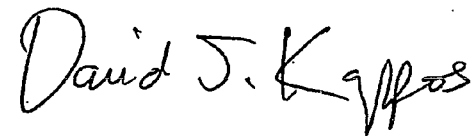
$$x_j = x_{j-1} + \sum_{i=1}^a v_{(j-1)a+i},$$

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



(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
- (65) **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)

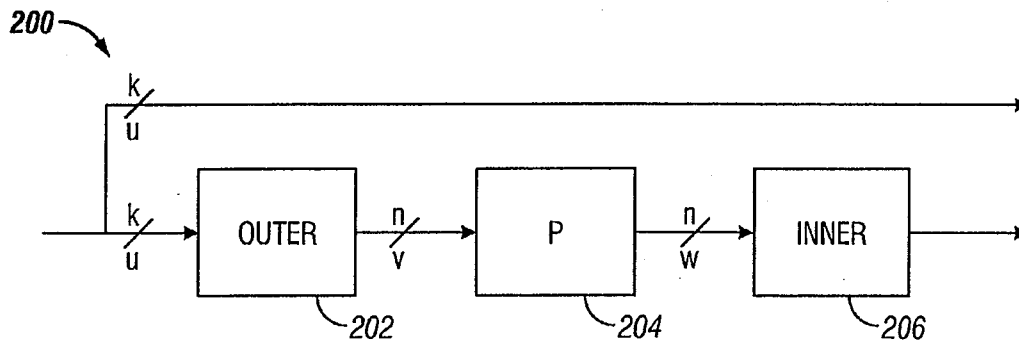
*Primary Examiner* — Dac V Ha

(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 (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 36<sup>th</sup> 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," *2<sup>nd</sup> International Symposium on Turbo Codes*, Brest, France, 25 pages, Sep. 2000.
- Jin, H., et al., "Irregular Repeat—Accumulate Codes," *2<sup>nd</sup> 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



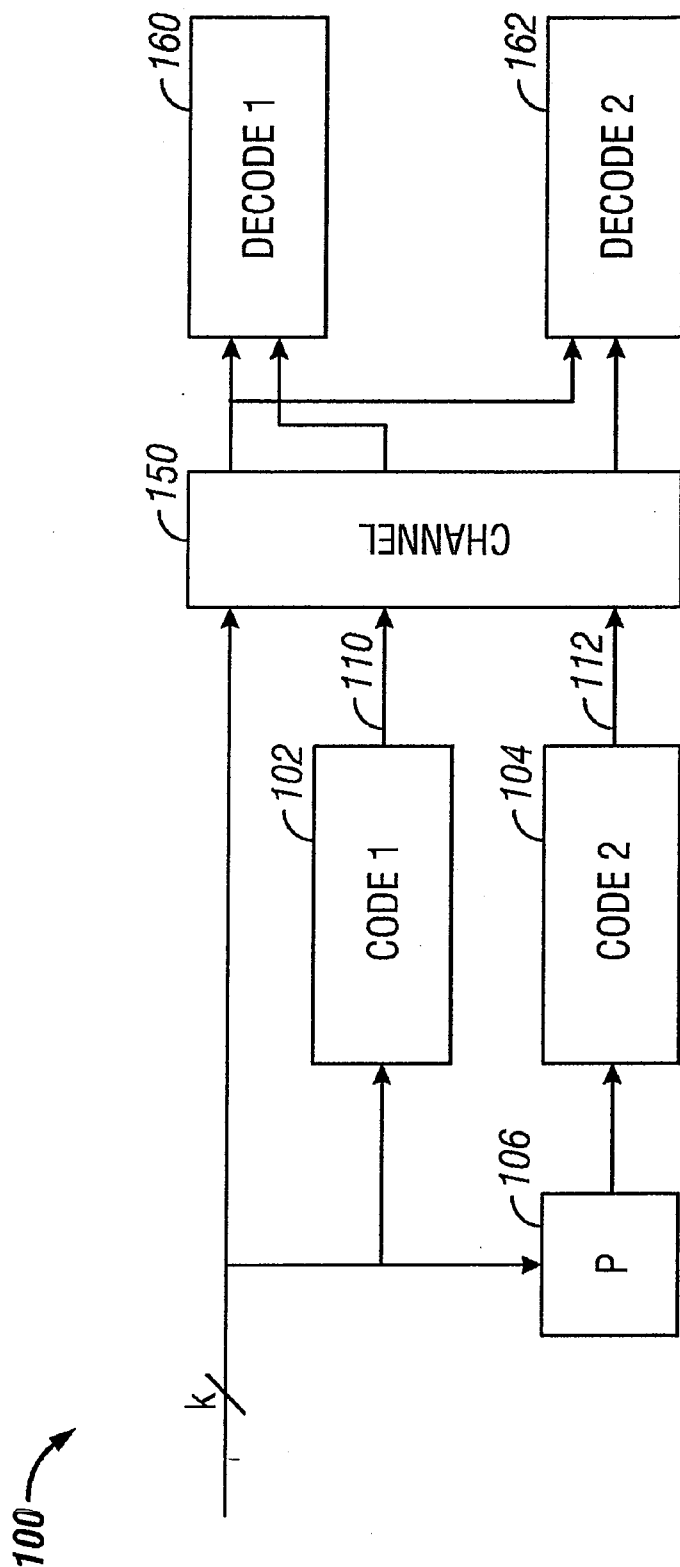


FIG. 1  
(Prior Art)

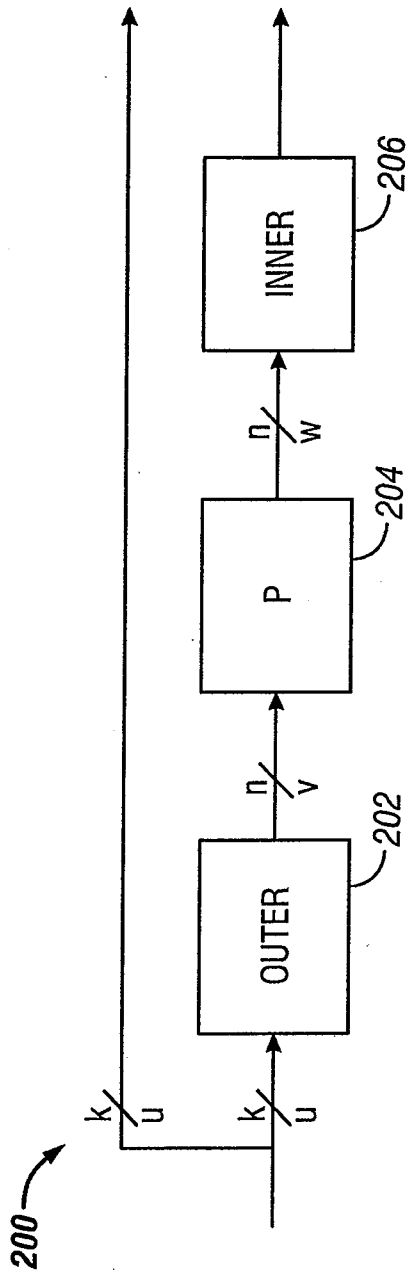


FIG. 2

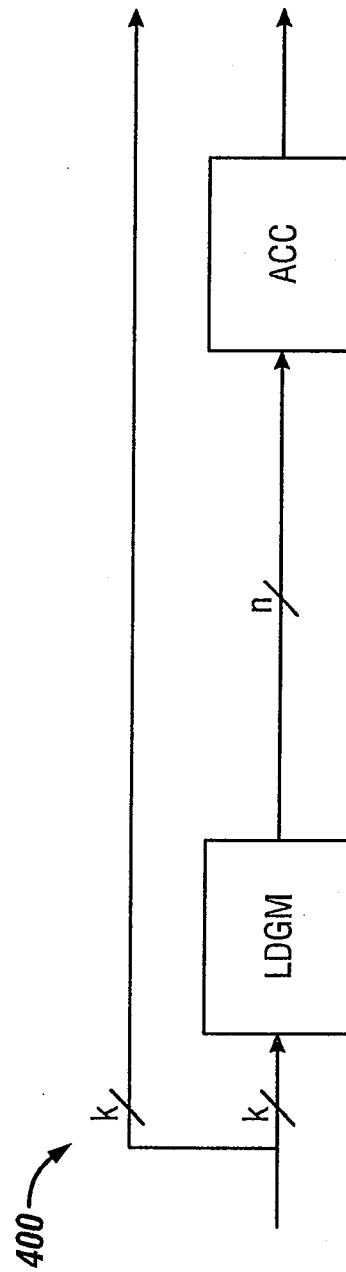
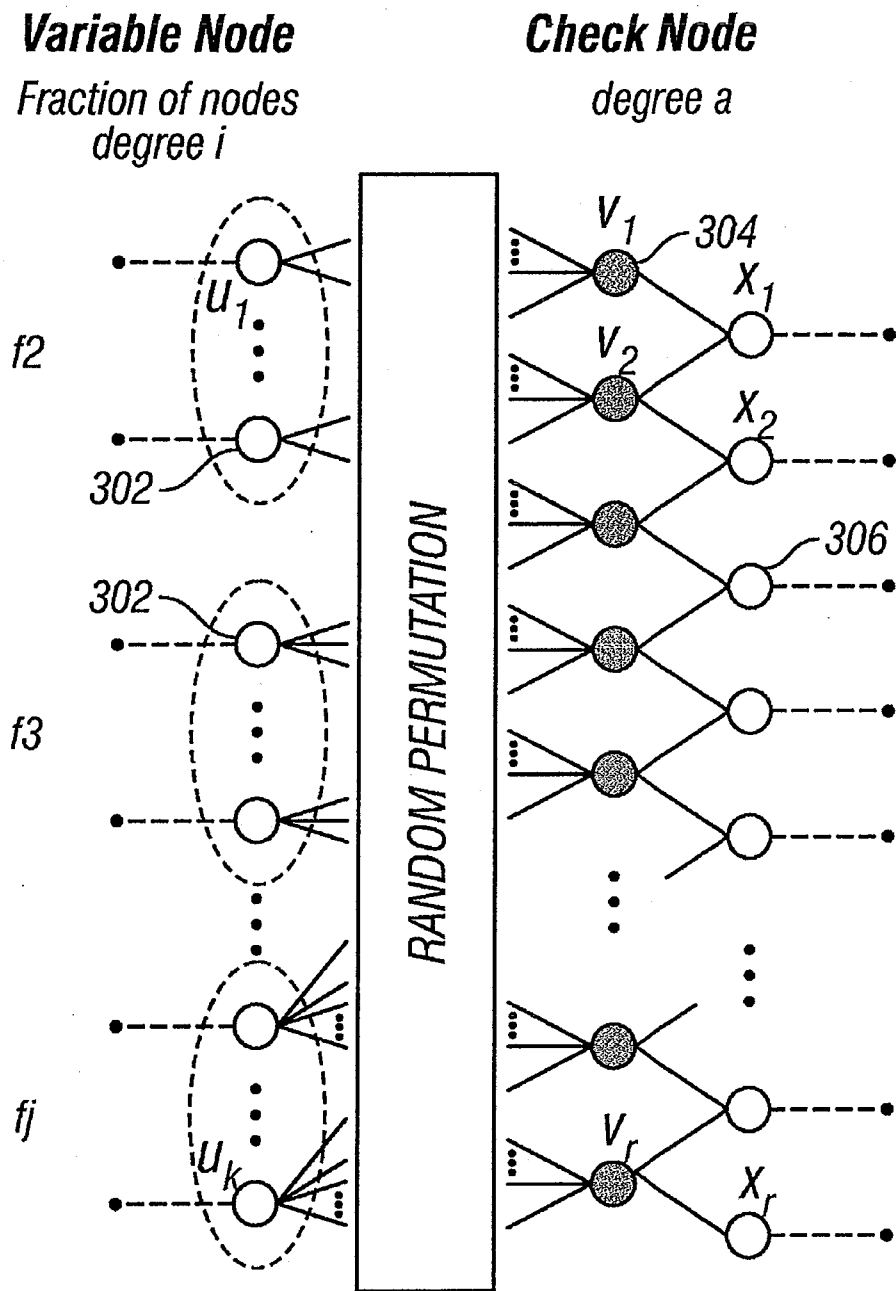


FIG. 4



**FIG. 3**

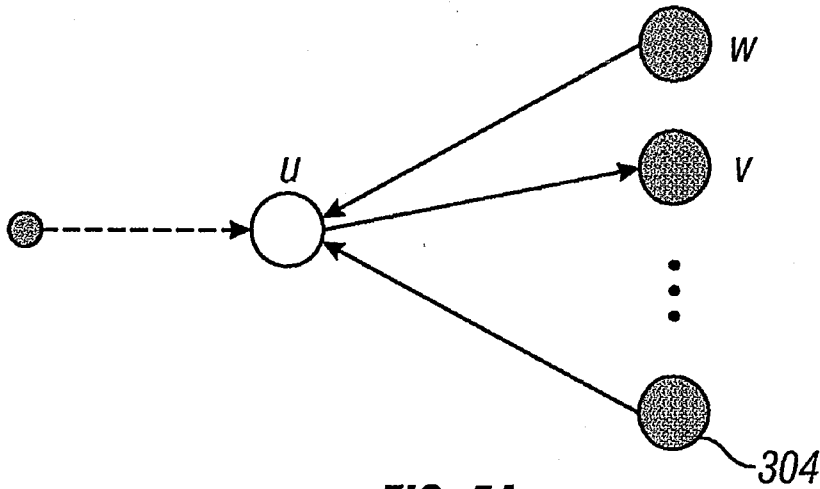


FIG. 5A

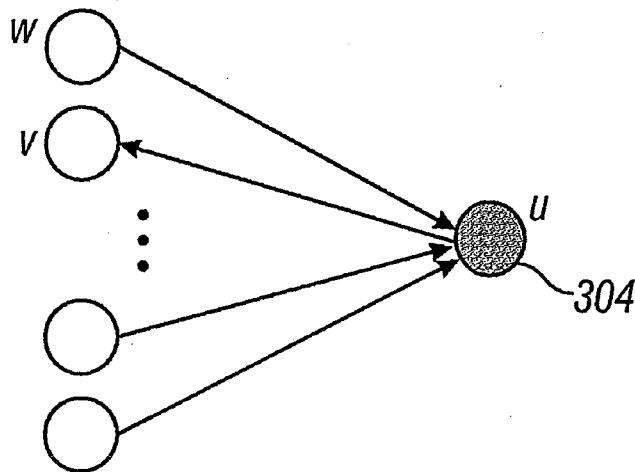


FIG. 5B

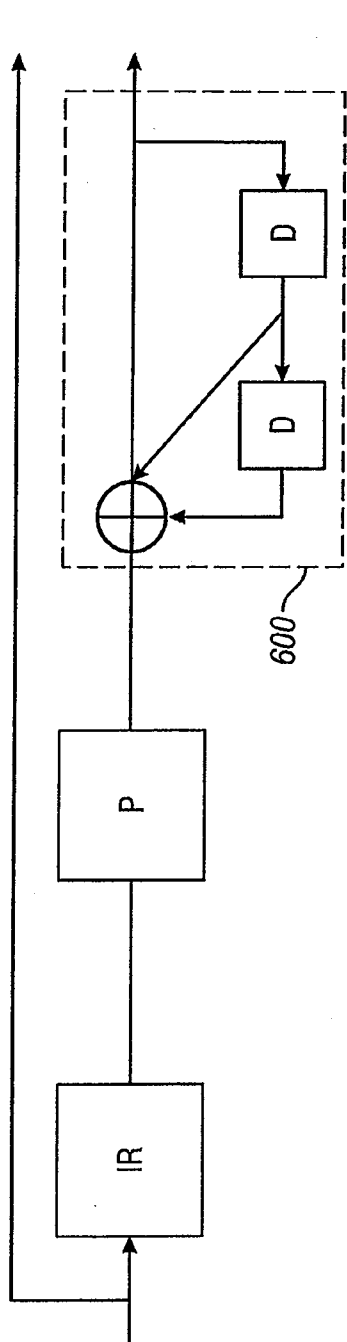


FIG. 6

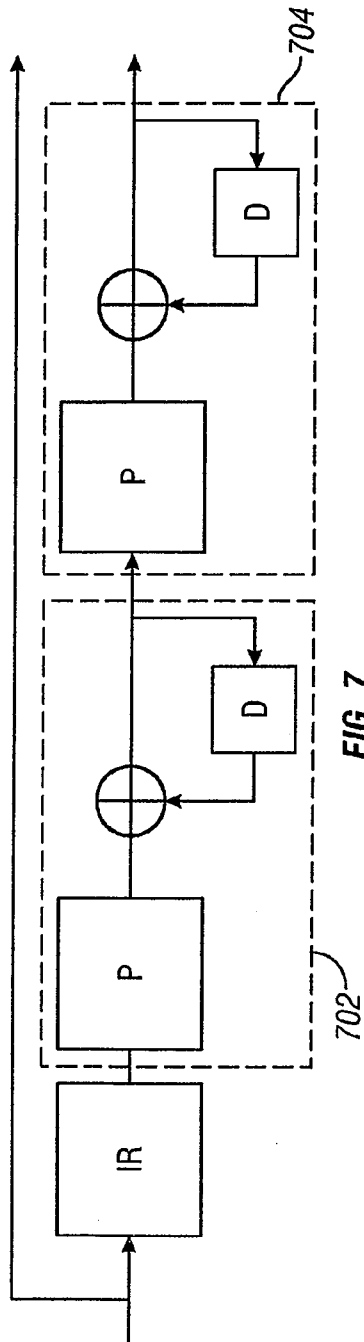


FIG. 7

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  $x = T_r w$ , where  $T_r$  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

$$y_1 = x_1$$

$$y_2 = x_1 \oplus x_2$$

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

$$\dots$$

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

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  $(f_1, \dots, f_j, a)$ , 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

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

$(v_1, \dots, v_{ra})$ , 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  $\lambda_i$  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  $\rho_i$  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  $(\lambda, \rho)$  is called a degree distribution. For  $L(x) = \sum_i f_i x^i$ ,

The rate of the systematic IRA code given by the

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

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

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.

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

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

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
$\lambda_2$	0.139025	0.078194	0.054485
$\lambda_3$	0.2221555	0.128085	0.104315
$\lambda_5$		0.160813	
$\lambda_6$	0.638820	0.036178	0.126755
$\lambda_{10}$			0.229816
$\lambda_{11}$			0.016484
$\lambda_{12}$		0.108828	
$\lambda_{13}$		0.487902	
$\lambda_{14}$			
$\lambda_{16}$			
$\lambda_{27}$			0.450302
$\lambda_{28}$			0.017842
Rate	0.333364	0.333223	0.333218
$\sigma_{GA}$	1.1840	1.2415	1.2615
$\sigma^*$	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.).

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.

\* \* \* \* \*



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**

(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 0 days.

(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

(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
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.

(Continued)

**OTHER PUBLICATIONS**

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

(Continued)

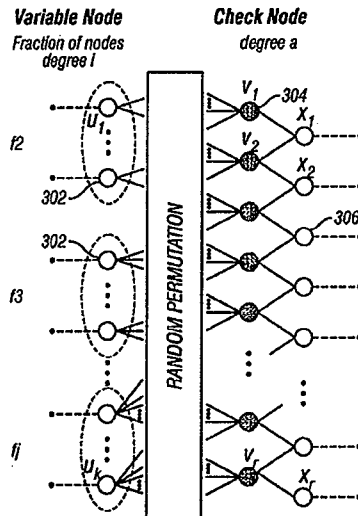
*Primary Examiner* — Dac Ha

(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



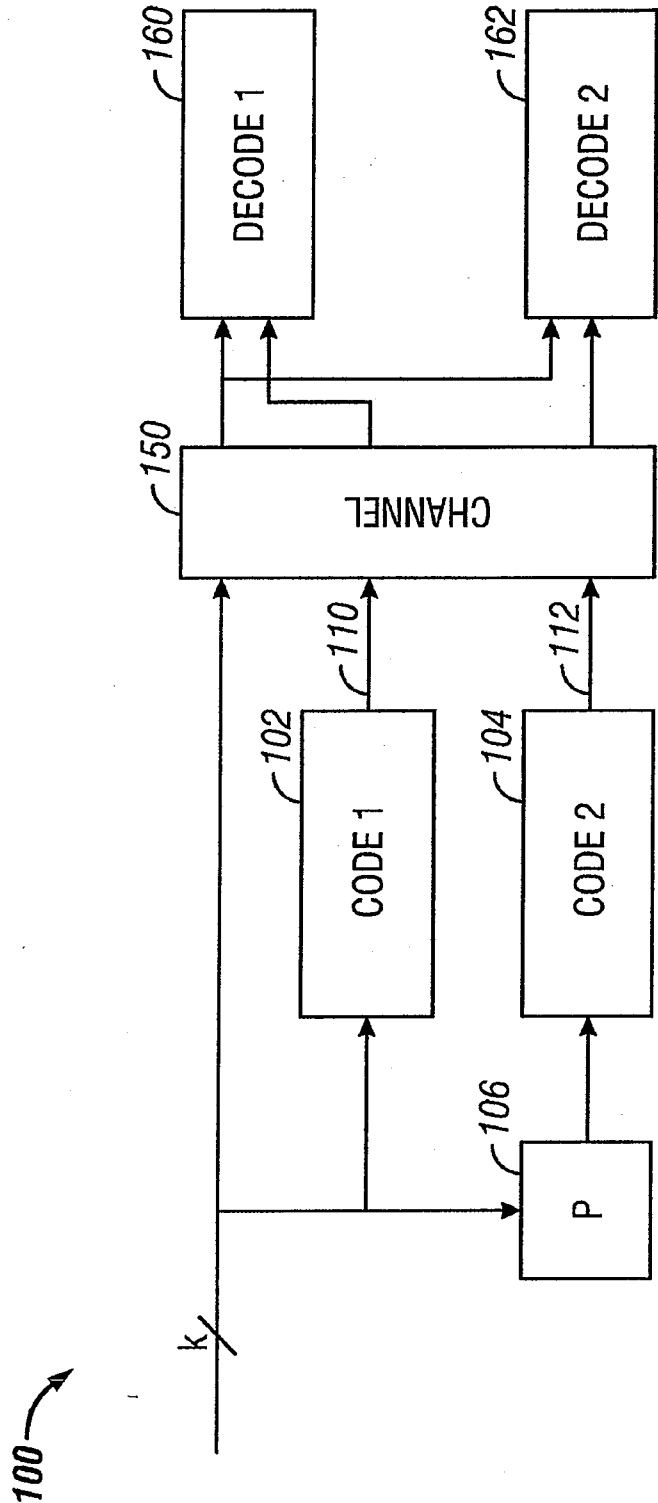


FIG. 1  
(Prior Art)

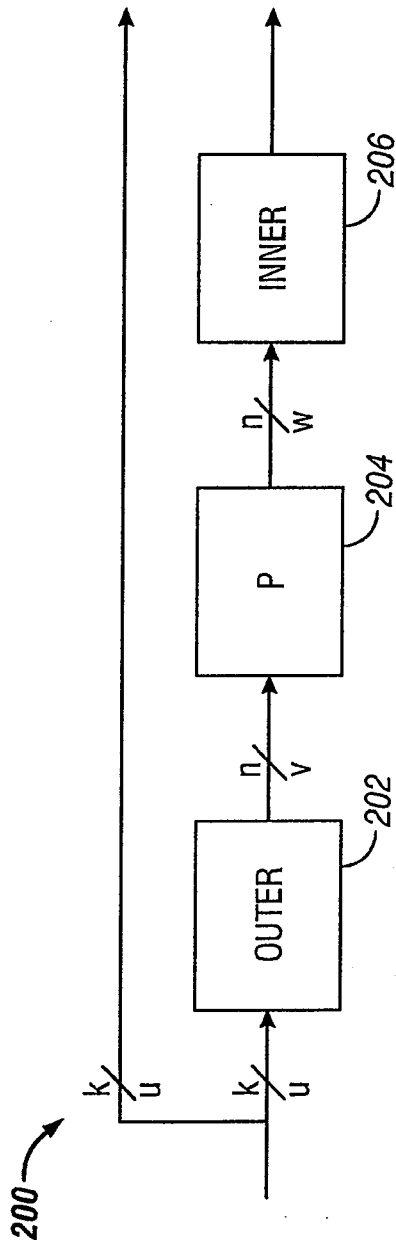


FIG. 2

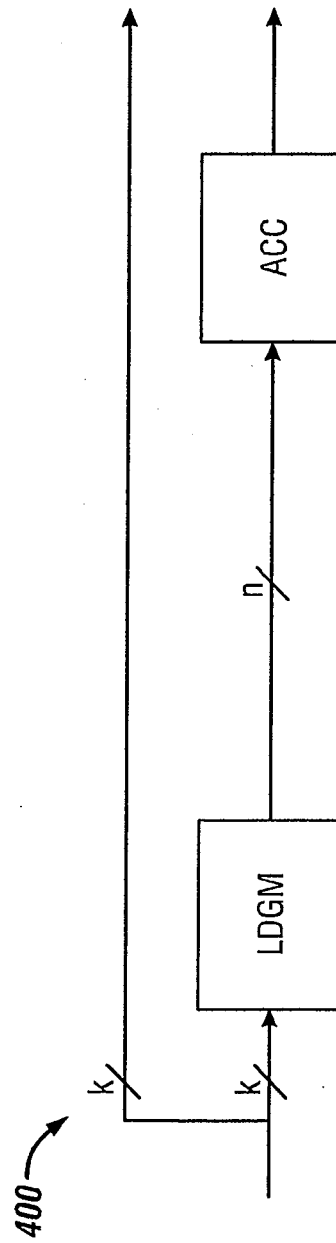


FIG. 4

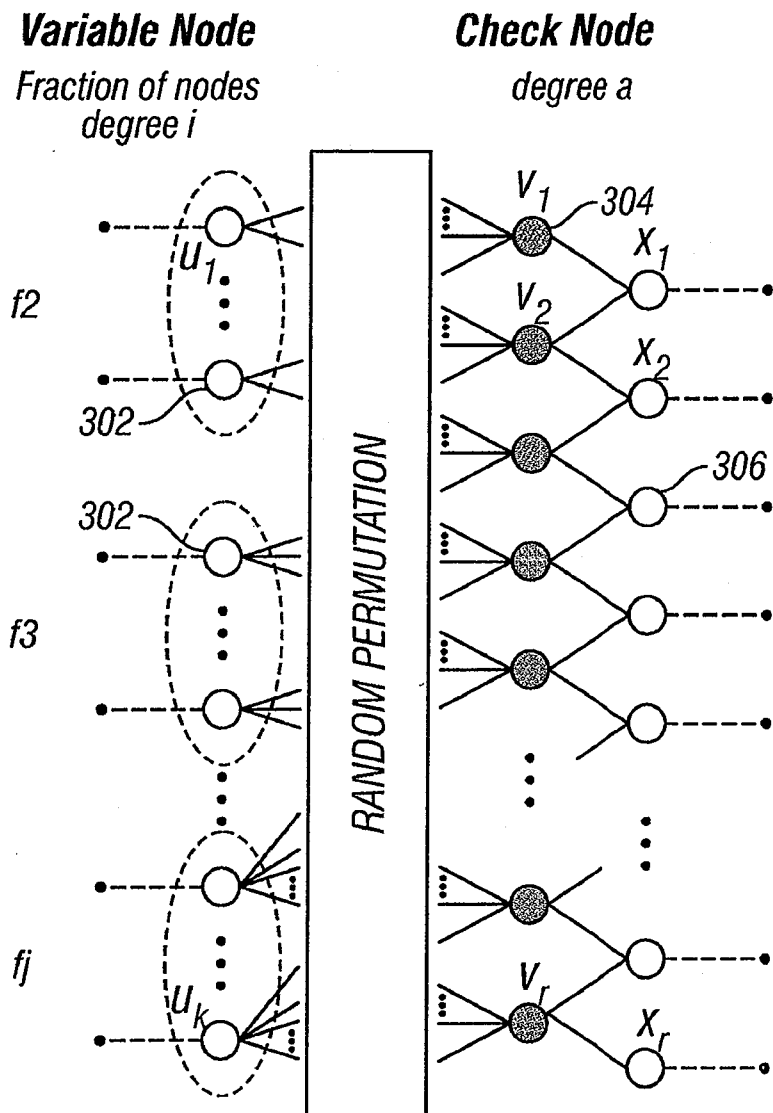


FIG. 3

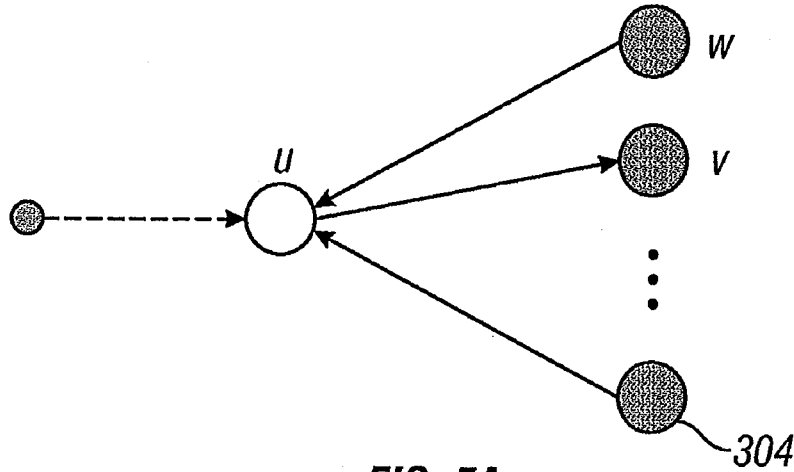


FIG. 5A

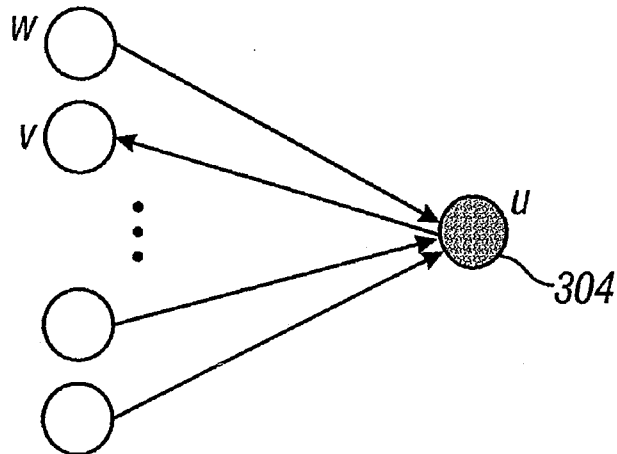


FIG. 5B

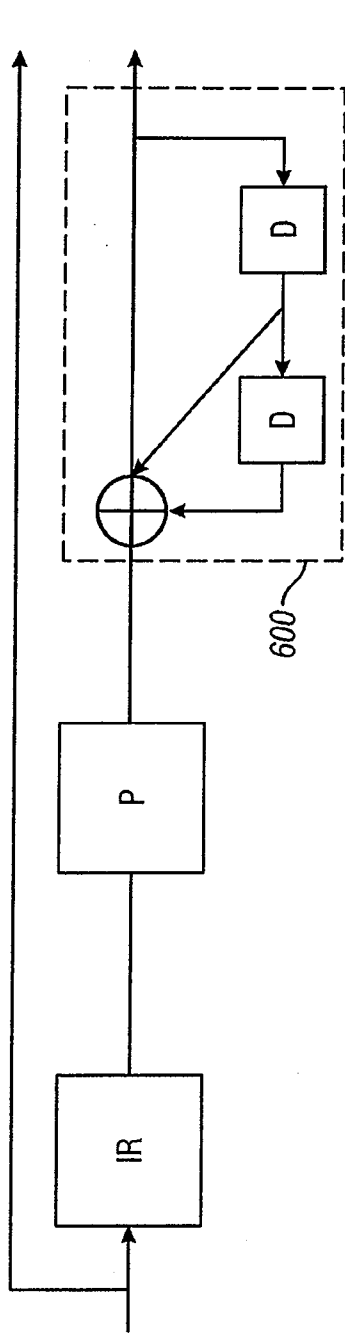


FIG. 6

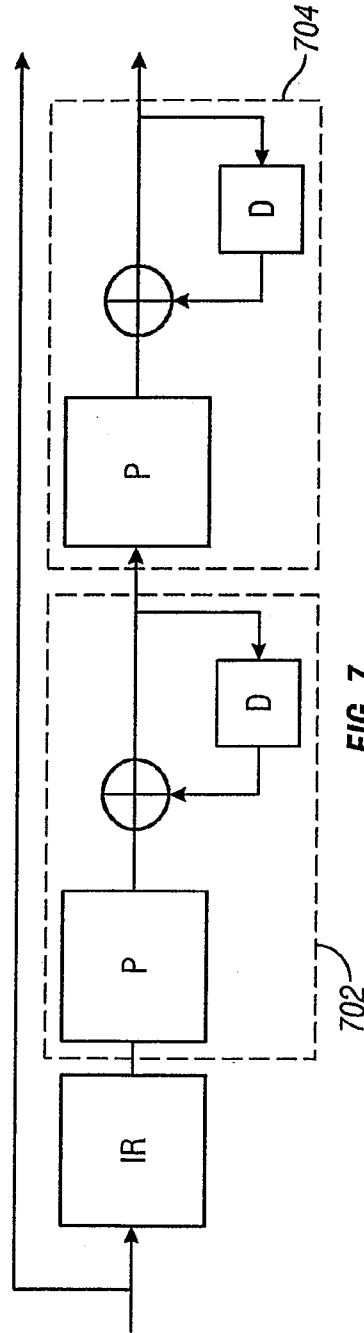


FIG. 7

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  $v = T_0 u$ , 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  $x = T_1 w$ ,



3

where  $T_i$  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  $(u_1, \dots, u_k, x_1, \dots, x_r)$ .

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

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  $R_{\text{sys}}$  simplifies to  $R=1/q$ .

The IRA code may be represented using an alternate notation. Let  $\lambda_i$  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  $\rho_i$  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  $(\lambda, \rho)$  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 (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

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  $y \in R$ , 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
$\lambda_2$		0.139025	0.078194	0.054485
$\lambda_3$		0.2221555	0.128085	0.104315
$\lambda_5$			0.160813	
$\lambda_6$		0.638820	0.036178	0.126755
$\lambda_{10}$				0.229816
$\lambda_{11}$				0.016484
$\lambda_{12}$			0.108828	
$\lambda_{13}$			0.487902	
$\lambda_{14}$				
$\lambda_{16}$				0.450302
$\lambda_{27}$				0.017842
$\lambda_{28}$				0.333218
Rate		0.333364	0.333223	0.333218
$\sigma_{GA}$		1.1840	1.2415	1.2615
$\sigma^*$		1.1981	1.2607	1.2780
$(E_b/N_0) * \text{ (dB)}$		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

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 "Berrou" and insert -- Berrou --, therefor.

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

# Irregular Repeat–Accumulate Codes <sup>1</sup>

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

<sup>1</sup>This 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.



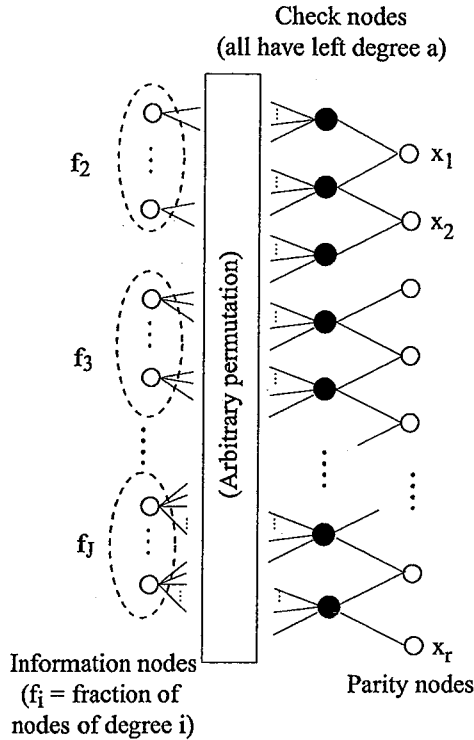


Figure 1: Tanner graph for IRA code with parameters  $(f_1, \dots, f_J; a)$ .

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  $\lambda_i$  be the fraction of *edges* between the information and the check nodes that are adjacent to an information node of degree  $i$ , and let  $\rho_i$  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  $\lambda_i$  and  $\rho_i$  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  $(\lambda, \rho)$  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



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

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

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

We now make the following new definitions

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

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

$$g_p(x) \triangleq h_p^{-1}(x) \tag{19}$$

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

$$p\lambda(f_p(x)) < x, \quad \forall x \neq 0 \tag{20}$$

which can be rewritten as

$$\lambda(x) < \frac{f_p^{-1}(x)}{p} \tag{21}$$

We make the following choice of  $\lambda(x)$ :

$$\lambda(x) = \frac{1}{p} \left( \sum_{i=1}^{N-1} g_{p,i} x^i + \epsilon x^N \right) \tag{22}$$

where  $0 < \epsilon < g_{p,N}$  and  $\sum_{i=1}^{N-1} g_{p,i} + \epsilon = p$ . Such a choice of  $N$  and  $\epsilon$  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

$$p\lambda(x) < g_p(x) = h_p^{-1}(x) < f_p^{-1}(x) \quad \forall x \neq 0 \tag{23}$$

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,

$$\lim_{a \rightarrow \infty} a \sum_i \frac{\lambda_i}{i} = \lim_{a \rightarrow \infty} a \left( \sum_{i=1}^{N-1} \frac{g_{p,i}}{i} + \frac{\epsilon}{N} \right) \tag{24}$$

We also have

$$\lim_{a \rightarrow \infty} a \sum_{i=N}^{\infty} \frac{g_{p,i}}{i} \leq \lim_{a \rightarrow \infty} \frac{a}{N} \sum_{i=N}^{\infty} g_{p,i} \leq \lim_{a \rightarrow \infty} \frac{a}{N} = 0 \tag{25}$$

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

$$\begin{aligned} &\frac{a}{a+1} \int_0^1 \left( \frac{1-p}{1-px^a} \right)^2 x^{a-1} dx \\ &< 1 - \int_0^1 h_p(x) dx \\ &< \int_0^1 \left( \frac{1-p}{1-px^a} \right)^2 x^{a-1} dx. \end{aligned} \tag{26}$$

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  $1/2$ .  $\delta$  is the code threshold (maximum allowable value of  $p$ ),  $N$  the number of terms in  $\lambda(x)$ , and  $R$  the rate of the code.

$a$	$\delta$	$N$	$1 - R$	$\delta/(1 - R)$
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  $\sigma^2$  Gaussian random variable. For a given noise variance  $\sigma^2$ , 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  $\mu$  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  $\mu$  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  $\mu_o$  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  $\mu_R$  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  $\mu_L$  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
$\lambda_2$	0.139025	0.078194	0.054485
$\lambda_3$	0.222155	0.128085	0.104315
$\lambda_5$		0.160813	
$\lambda_6$	0.638820	0.036178	0.126755
$\lambda_{10}$			0.229816
$\lambda_{11}$			0.016484
$\lambda_{12}$		0.108828	
$\lambda_{13}$		0.487902	
$\lambda_{14}$			
$\lambda_{16}$			
$\lambda_{27}$			0.450302
$\lambda_{28}$			0.017842
rate	0.333364	0.333223	0.333218
$\sigma_{GA}$	1.1840	1.2415	1.2615
$\sigma^*$	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

$\sigma_{GA}$  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
$\lambda_2$		0.0577128
$\lambda_3$	0.252744	0.117057
$\lambda_7$		0.2189922
$\lambda_8$		0.0333844
$\lambda_{11}$	0.081476	
$\lambda_{12}$	0.327162	
$\lambda_{18}$		0.2147221
$\lambda_{20}$		0.0752259
$\lambda_{46}$	0.184589	
$\lambda_{48}$	0.154029	
$\lambda_{55}$		0.0808676
$\lambda_{58}$		0.202038
rate	0.50227	0.497946
$\sigma^*$	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.

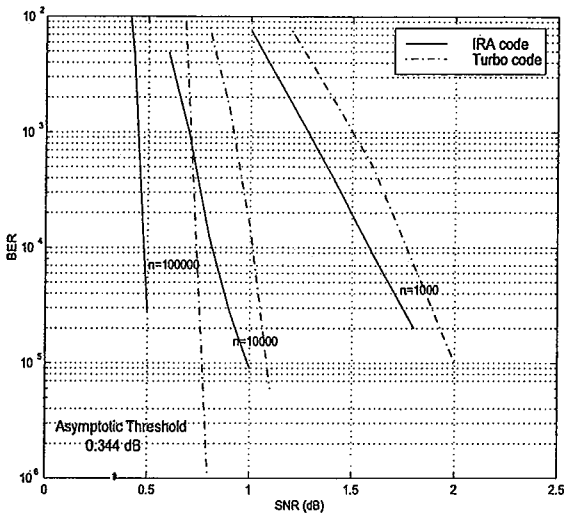


Figure 2: Comparison between turbo codes (dashed curves) and IRA codes (solid curves) of lengths  $n = 10^3$ ,  $10^4$ ,  $10^5$ . All codes are of rate one-half.

### 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.

# 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

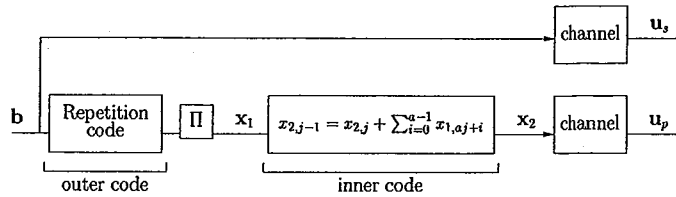


Fig. 1. IRA encoder.

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,

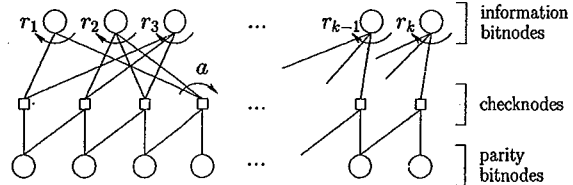


Fig. 2. Tanner graph of an IRA code.

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.<sup>1</sup>

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  $\alpha$  and checknode  $\beta$  if the symbol corresponding to  $\alpha$  participates in the parity-check equation corresponding to  $\beta$ .

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

<sup>1</sup>If 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  $\alpha$  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  $y_\alpha$  is the channel output corresponding to the transmission of the code symbol  $x_\alpha$ .

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  $\alpha$  of degree  $i$  and let  $m_1, \dots, m_{i-1}$  denote the messages received from the  $i - 1$  incoming edges and  $u_\alpha$  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  $\beta$  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  $\ell$  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  $\ell$  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  $\mathcal{F}_{\text{sym}}$ .

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  $\Delta_0$ , for which  $\text{Pe}(\Delta_0) = 1/2$ , and the “delta at infinity” distribution, denoted by  $\Delta_\infty$ , 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  $\Delta_\infty$  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  $P_\ell$  (respectively,  $\tilde{P}_\ell$ ) denote the average distribution of messages passed from an information bitnode (respectively, parity bitnode) to a checknode, at iteration  $\ell$ . Let  $Q_\ell$  (respectively,  $\tilde{Q}_\ell$ ) denote the average distribution of messages passed from a checknode to an information bitnode (respectively, parity bitnode), at iteration  $\ell$ .

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  $\Gamma^{-1}$  denotes the inverse mapping of  $\Gamma$ , 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 trajectory

of (10)–(13) are sequences of pairs of symmetric distributions  $(P_\ell, \tilde{P}_\ell)$ ). For this system, the BER at iteration  $\ell$  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  $\ell$  tends to infinity. On the contrary, if  $\lambda_2$  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  $\nu$  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  $\ell$

$$\nu \leq \nu' \Leftrightarrow \text{Pe}(P_\ell) \leq \text{Pe}(P'_\ell)$$

where  $P_\ell$  and  $P'_\ell$  are the message distributions at iteration  $\ell$  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*  $\nu^*$  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  $\nu$ , 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  $(x_\ell, \tilde{x}_\ell)$  are the system state variables.

By eliminating the intermediate distributions  $Q_\ell$  and  $\tilde{Q}_\ell$ , 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  $\Phi$  and  $\Psi$  and the functions  $\phi$  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)  $\phi$  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  $\phi(x, \tilde{x}) - \tilde{x}$  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

$$x < \phi(x, \tilde{x}(x)), \quad \forall x \in [0, 1). \quad (24)$$

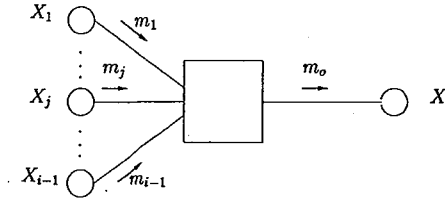


Fig. 3. EXIT model.

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  $\Phi$  and  $\Psi$ , and in the way the *intermediate* distributions  $Q_\ell$  and  $\tilde{Q}_\ell$  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 sub-graph 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 capacity<sup>2</sup> 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  $F_{\text{in}}$  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  $F_{\text{in}}$ . 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  $p_{Y|X}(u|0) = F(u)$  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  $\mu$ , i.e.,  $\mathcal{N}_{\text{sym}}(\mu) \triangleq \mathcal{N}(\mu, 2|\mu|)$ .

<sup>2</sup>Recall that the capacity of a binary-input symmetric-output memoryless channel is achieved by uniform i.i.d. inputs.

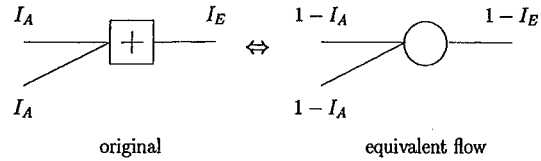


Fig. 4. Reciprocal channel approximation.

For a distribution  $F \in \mathcal{F}_{\text{sym}}$ , we let the mapping  $\Phi$  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,  $\Psi$  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  $\mu_\ell$  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,  $\mu_\ell$  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

$$F_u \otimes \lambda(Q_\ell) = \sum_{i=2}^d \lambda_i \mathcal{N}_{\text{sym}}(2/\sigma^2 + (i-1)\mu_\ell) \quad \diamond$$

$$F_u \otimes \tilde{Q}_\ell = \mathcal{N}_{\text{sym}}(2/\sigma^2 + \tilde{\mu}_\ell). \quad (37)$$

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.

### 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  $\Phi$  be equal to  $\mathcal{I}$  defined in (28) and the mapping  $\Psi$  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

$$Q_\ell = \mathcal{E}_{\text{sym}}(1 - x_{\ell-1}^{a-1} \tilde{x}_{\ell-1}^2) \\ \tilde{Q}_\ell = \mathcal{E}_{\text{sym}}(1 - x_{\ell-1}^a \tilde{x}_{\ell-1}) \quad (40)$$

where we used the fact that, as it can be easily seen from the definitions of  $\Gamma$  and  $\Gamma^{-1}$  in (46)–(48)

$$\Gamma^{-1}(\Gamma(\mathcal{E}_{\text{sym}}(\epsilon_1)) \otimes \Gamma(\mathcal{E}_{\text{sym}}(\epsilon_2))) \\ = \mathcal{E}_{\text{sym}}(1 - (1 - \epsilon_1)(1 - \epsilon_2)).$$

Notice that, while in Method 1 we assumed  $Q_\ell$  and  $\tilde{Q}_\ell$  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

$$x_\ell = 1 - (1 - \mathcal{I}(F_u)) \sum_{i=2}^d \lambda_i (1 - x_{\ell-1}^{a-1} \tilde{x}_{\ell-1}^2)^{i-1} \\ \tilde{x}_\ell = 1 - (1 - \mathcal{I}(F_u)) (1 - x_{\ell-1}^a \tilde{x}_{\ell-1}). \quad (41)$$

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  $\Phi$  and  $\Psi$ .

In this case, the fixed-point equation corresponding to (23) is obtained in closed form as

$$x = 1 - (1 - \mathcal{I}(F_u)) \\ \times \sum_{i=2}^d \lambda_i \left( 1 - \frac{x^{a-1} \mathcal{I}(F_u)^2}{(1 - (1 - \mathcal{I}(F_u))x^a)^2} \right)^{i-1} \quad (42)$$

(for details, see [19]).

---


$$\begin{cases} x_\ell = 1 - \sum_{i=2}^d \sum_{j=1}^D \lambda_i p_j \frac{1}{\sqrt{\pi}} \int_{-\infty}^{\infty} e^{-z^2} \log_2 \left( 1 + e^{-2\sqrt{(i-1)\mu_\ell z - (i-1)\mu_\ell - v_j}} \right) dz \\ \tilde{x}_\ell = 1 - \sum_{j=1}^D p_j \frac{1}{\sqrt{\pi}} \int_{-\infty}^{\infty} e^{-z^2} \log_2 \left( 1 + e^{-2\sqrt{\mu_\ell z - \tilde{\mu}_\ell - v_j}} \right) dz. \end{cases} \quad (35)$$


---

$$x_\ell = \sum_{i=2}^d \lambda_i J \left( \frac{2}{\sigma^2} + (i-1)J^{-1} \left( 1 - J \left( (a-1)J^{-1}(1 - x_{\ell-1}) + 2J^{-1}(1 - \tilde{x}_{\ell-1}) \right) \right) \right) \\ \tilde{x}_\ell = J \left( \frac{2}{\sigma^2} + J^{-1} \left( 1 - J \left( aJ^{-1}(1 - x_{\ell-1}) + J^{-1}(1 - \tilde{x}_{\ell-1}) \right) \right) \right). \quad (38)$$

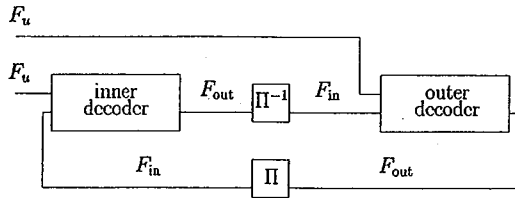


Fig. 5. Turbo-like IRA decoder.

#### 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  $Q_\ell$  and  $\tilde{Q}_\ell$ .

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 have<sup>3</sup>

$$h(I_A) = 1 - \sum_{i=2}^d \sum_{j=1}^D \lambda_i p_j \frac{1}{\sqrt{\pi}} \int_{-\infty}^{\infty} e^{-z^2} \times \log_2 \left( 1 + e^{-2\sqrt{(i-1)J^{-1}(I_A)}z - (i-1)J^{-1}(I_A) - v_j} \right) dz. \quad (44)$$

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.  $\square$

<sup>3</sup>Just 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.

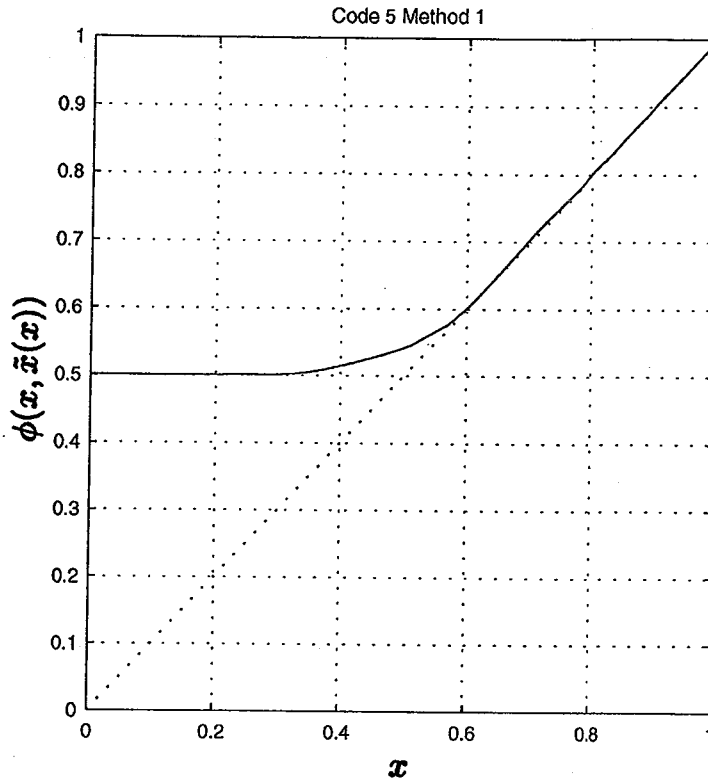


Fig. 6. Mutual information EXIT functions for BIAWGNC and Method 1.

TABLE I  
 OPTIMIZATION FOR THE BIAWGNC

	Method 1		Method 2		Method 3		Method 4	
	$i$	$\lambda_i$	$i$	$\lambda_i$	$i$	$\lambda_i$	$i$	$\lambda_i$
	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 <sub>gap</sub> (DE)	0.059		0.406		0.075		0.306	
SNR <sub>gap</sub> (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  $\nu^*$ , evaluated by exact DE, is worse than the channel parameter  $\nu$  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	
	$i$	$\lambda_i$	$i$	$\lambda_i$	$i$	$\lambda_i$	$i$	$\lambda_i$
	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  $\text{SNR}_{\text{gap}}$  (DE) or  $p_{\text{gap}}$  (DE) in the tables). Then, we compare it to the threshold of the approximated DE ( $\text{SNR}_{\text{gap}}$  (approximately) and  $p_{\text{gap}}$  (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

Rate	Method 1			Method 3			
	$a$	$d$	$\text{SNR}_{\text{gap}}$	Rate	$a$	$d$	$\text{SNR}_{\text{gap}}$
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

Rate	Method 2			Method 4			
	$a$	$d$	$\text{SNR}_{\text{gap}}$	Rate	$a$	$d$	$\text{SNR}_{\text{gap}}$
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).

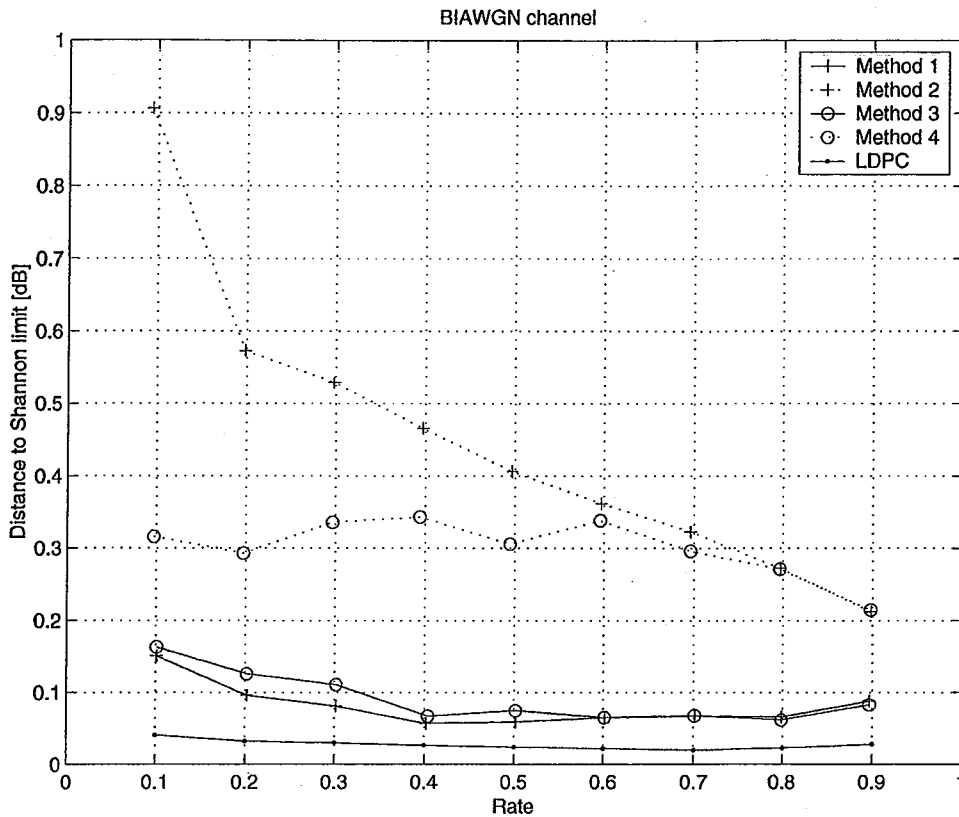


Fig. 7. Gap to Shannon limit (obtained by DE) versus rate for BIAWGN.

TABLE V  
 IRA CODES, DESIGNED WITH METHODS 1 AND 3, EVALUATED WITH DE, FOR BSC

Method 1				Method 3			
Rate	$a$	$d$	$p_{gap}$	Rate	$a$	$d$	$p_{gap}$
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	$a$	$d$	$p_{gap}$	Rate	$a$	$d$	$p_{gap}$
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.

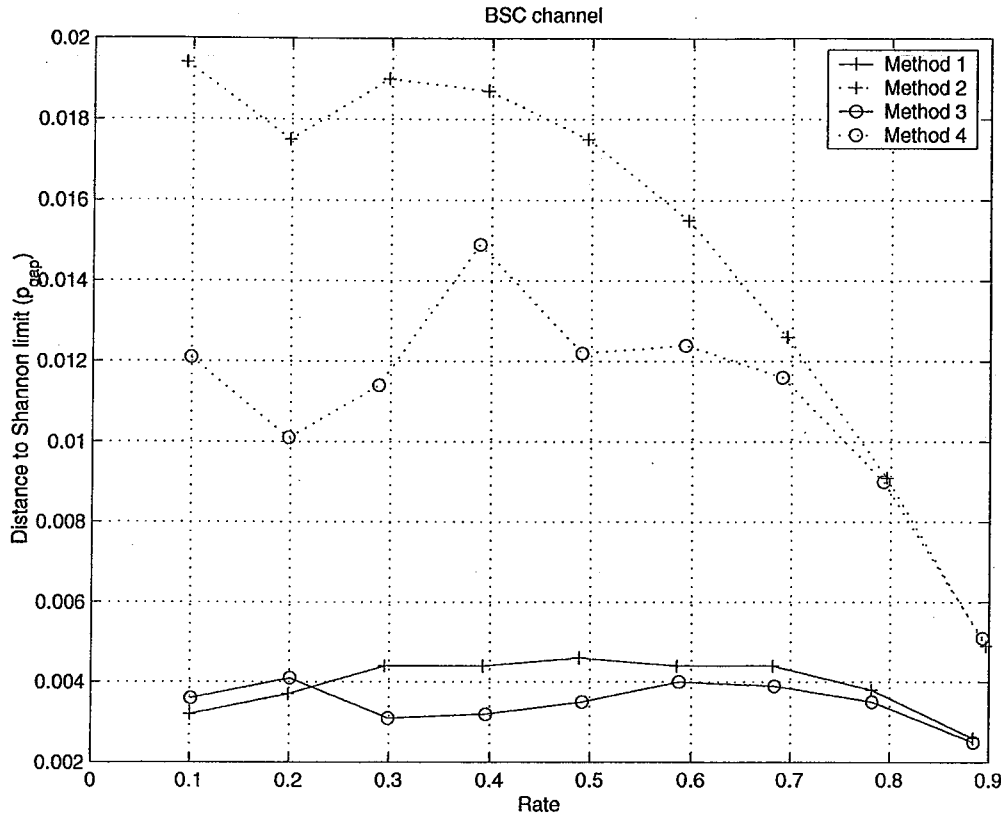


Fig. 8. Gap to Shannon limit (obtained by DE) versus rate for BSC.

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  $\Gamma$  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  $\Gamma$  applied to  $\Delta_0$  and  $\Delta_\infty$  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  $\Gamma^{-1}$  yields

$$\begin{aligned} \Gamma^{-1}(G)(z) &= \chi_{\{z>0\}}\left(1 - G_0\left(-\log \tanh \frac{z}{2}\right)\right) \\ &+ \chi_{\{z<0\}}G_1\left(-\log \tanh \frac{-z}{2}\right). \end{aligned} \quad (48)$$

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

$$\begin{aligned} &(\chi_0 G_0(z) + \chi_1 G_1(z))^{\otimes m} \\ &= \chi_0 \left( \sum_{j=0}^{\lfloor \frac{m}{2} \rfloor} \binom{m}{2j} G_0^{\otimes(m-2j)} \otimes G_1^{\otimes 2j} \right) \\ &+ \chi_1 \left( \sum_{j=0}^{\lfloor \frac{m-1}{2} \rfloor} \binom{m}{2j+1} G_0^{\otimes(m-2j-1)} \otimes G_1^{\otimes 2j+1} \right) \end{aligned} \quad (49)$$

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

$$\begin{aligned} \Gamma(P_0)^{\otimes n} &= \chi_0((1 - 2n\epsilon)\Delta_0 + n\epsilon\Delta_\infty) \\ &+ \chi_1(n\epsilon\Delta_\infty) + O(\epsilon^2) \\ \Gamma(\tilde{P}_0)^{\otimes 2} &= \chi_0((1 - 4\delta)\Delta_0 + 2\delta\Delta_\infty) \\ &+ \chi_1(2\delta\Delta_\infty) + O(\delta^2). \end{aligned}$$

By applying  $\Gamma^{-1}$  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

$$\begin{bmatrix} P_1 \\ \tilde{P}_1 \end{bmatrix} = A \begin{bmatrix} 2\epsilon \\ 2\delta \end{bmatrix} F_u + \left( \begin{bmatrix} 1 \\ 1 \end{bmatrix} - A \begin{bmatrix} 2\epsilon \\ 2\delta \end{bmatrix} \right) \Delta_\infty + \begin{bmatrix} O(\epsilon^2) \\ O(\delta^2) \end{bmatrix}$$

where

$$A = \begin{bmatrix} (a-1)\lambda_2 & 2\lambda_2 \\ a & 1 \end{bmatrix}. \quad (51)$$

After  $\ell$  iterations we obtain

$$\begin{bmatrix} P_\ell \\ \tilde{P}_\ell \end{bmatrix} = A^\ell \begin{bmatrix} 2\epsilon \\ 2\delta \end{bmatrix} F_u^{\otimes \ell} + \left( \begin{bmatrix} 1 \\ 1 \end{bmatrix} - A^\ell \begin{bmatrix} 2\epsilon \\ 2\delta \end{bmatrix} \right) \Delta_\infty + \begin{bmatrix} O(\epsilon^2) \\ O(\delta^2) \end{bmatrix}. \quad (52)$$

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  $P_\ell$  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  $Ae^{-r}$  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  $\lambda_2$ , 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  $\phi$  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

$$J'_{v_j}(\mu) = \frac{\log_2(e)}{\sqrt{\mu}} \frac{1}{\sqrt{\pi}} \int_{-\infty}^{+\infty} (z + \sqrt{\mu}) \times e^{-v_j} \frac{e^{-(z+\sqrt{\mu})^2}}{1 + e^{-2\sqrt{\mu}z - \mu - v_j}} dz.$$

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  $\mu$

$$\forall \mu > 0, \forall z \in \mathbb{R}, \quad \left| \frac{e^{-\frac{z^2}{\mu}}}{\cosh(z)} \right| \leq \frac{1}{\cosh(z)} \in L^1(\mathbb{R}).$$

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  $\mu$

$$f_{v_j}(\mu) \sim \frac{\sqrt{\pi}}{2} \frac{e^{-\frac{\mu}{4}} e^{-\frac{v_j}{2}}}{\sqrt{\mu}}.$$

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  $\phi$  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

$$\int_{-\infty}^0 e^{-z/2} dF_u(z) = \int_0^{+\infty} e^{-z/2} dF_u(z)$$

$$\int_{-\infty}^0 \log_2(1+e^{-z}) dF_u(z) = \int_0^{+\infty} e^{-z} \log_2(1+e^z) dF_u(z).$$

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

$$\forall z \geq 0, (1+e^{-z}) \log(1+e^{-z}) + ze^{-z} \leq 2(\log 2)e^{-z/2} \quad (64)$$

we get

$$\forall F_u \in \mathcal{F}_{\text{sym}}, 1 - \mathcal{I}(F_u) \leq e^{-r}$$

and the conclusion follows.  $\square$

In the following, we show inequality (64). Letting  $x = e^{-z}$ , (64) becomes equivalent to

$$\forall x \in [0, 1], f(x) \leq 0$$

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

$$\begin{aligned} I_A &= \frac{1}{n} \sum_{i=1}^n I(x_{1,i}; z_i) \\ I_E &= \frac{1}{n} \sum_{i=1}^n I(x_{1,i}; m_{o,i}) \end{aligned}$$

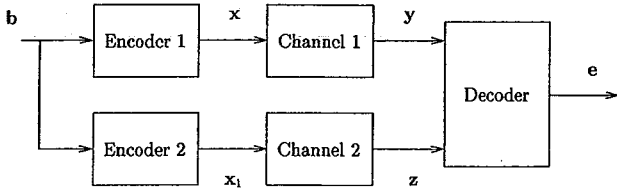


Fig. 9. General decoding model.

$$\stackrel{(a)}{=} \frac{1}{n} \sum_{i=1}^n I(x_{1,i}; \mathbf{y}, \mathbf{z}_{[i]}) \quad (67)$$

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).

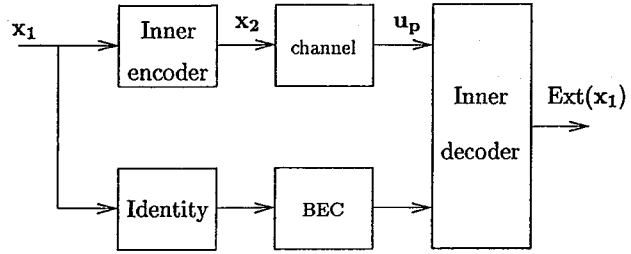
The outer MAP decoder receives channel observations  $\mathbf{u}_s$  for the information bits and input messages for the symbols of  $\mathbf{x}_1$ , and produces output messages for the symbols of  $\mathbf{x}_1$ . The general decoding model of Fig. 9, applied to the outer decoder, yields the model 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  $R_{in} = n/m$  and  $R_{out} = k/n$ , and the overall IRA coding rate (3) is given by

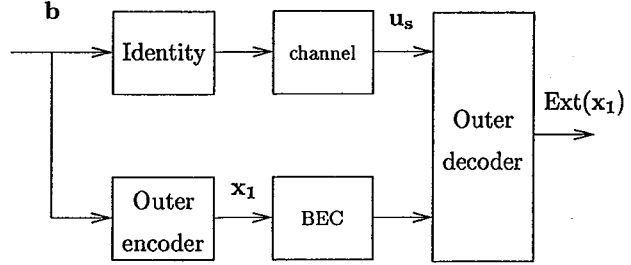
$$R = \frac{k}{k+m} = \frac{R_{in}R_{out}}{1 + R_{in}R_{out}}.$$

By applying Lemma 1 to the inner code model (Fig. 10(a)), we obtain

$$\begin{aligned} \mathcal{A}_{in} &= 1 - \frac{1}{n} H(\mathbf{x}_1 | \mathbf{u}_p) \\ &= 1 - \frac{1}{n} (H(\mathbf{x}_1) - I(\mathbf{x}_1; \mathbf{u}_p)) \\ &\stackrel{(a)}{=} \frac{1}{n} I(\mathbf{x}_1; \mathbf{u}_p) \\ &\stackrel{(b)}{=} \frac{m}{n} I(x_{2,i}; \mathbf{u}_{p,i}) = \mathcal{I}(F_u) / R_{in} \end{aligned} \quad (69)$$



(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

$$\begin{aligned} \mathcal{A}_{out} &= 1 - \frac{1}{n} H(\mathbf{x}_1 | \mathbf{u}_s) \\ &= 1 - \frac{1}{n} (H(\mathbf{x}_1) - I(\mathbf{x}_1; \mathbf{u}_s)) \\ &\stackrel{(a)}{=} 1 - \frac{k}{n} + \frac{1}{n} I(\mathbf{x}_1; \mathbf{u}_s) \\ &\stackrel{(b)}{=} 1 - \frac{k}{n} + \frac{k}{n} I(b_i; \mathbf{u}_{s,i}) \\ &= 1 - R_{out} + R_{out} \mathcal{I}(F_u) \end{aligned} \quad (70)$$

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  $I(\mathbf{x}_1; \mathbf{u}_s) = I(\mathbf{b}; \mathbf{u}_s) = kI(b_i; \mathbf{u}_{s,i}) = k\mathcal{I}(F_u)$ .

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

$$\mathcal{A}_{in} = \int_0^1 g(x) dx > \int_0^1 h^{-1}(x) dx = 1 - \mathcal{A}_{out}.$$

By using (69) and (70), we obtain

$$\begin{aligned} \mathcal{I}(F_u) / R_{in} &> R_{out} - R_{out} \mathcal{I}(F_u) \\ &\Downarrow \\ \mathcal{I}(F_u) &> \frac{R_{in}R_{out}}{1 + R_{in}R_{out}} = R. \end{aligned} \quad (71)$$

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

$$h(x) = 1 - (1 - \mathcal{I}(F_u)) \sum_{i=2}^d \lambda_i (1 - x)^{i-1}$$

$$g(x) = \frac{x^{\alpha-1} \mathcal{I}(F_u)^2}{(1 - (1 - \mathcal{I}(F_u))x^\alpha)^2}$$

and the area under these functions are again

$$A_{\text{out}} = \int_0^1 h(x) dx = 1 - (1 - \mathcal{I}(F_u)) \sum_{i=2}^d \lambda_i / i$$

$$= 1 - R_{\text{out}} + R_{\text{out}} \mathcal{I}(F_u)$$

$$A_{\text{in}} = \int_0^1 g(x) dx = \mathcal{I}(F_u) / \alpha = \mathcal{I}(F_u) / R_{\text{in}}$$

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 LPDC 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.

## A synthesizable IP Core for DVB-S2 LDPC Code Decoding

Frank Kienle, Torben Brack, Norbert Wehn  
Microelectronic System Design Research Group  
University of Kaiserslautern  
Erwin-Schrödinger-Straße  
67663 Kaiserslautern, Germany  
{kienle, brack, wehn}@eit.uni-kl.de

### Abstract

*The new standard for digital video broadcast DVB-S2 features Low-Density Parity-Check (LDPC) codes as their channel coding scheme. The codes are defined for various code rates with a block size of 64800 which allows a transmission close to the theoretical limits.*

*The decoding of LDPC is an iterative process. For DVB-S2 about 300000 messages are processed and reordered in each of the 30 iterations. These huge data processing and storage requirements are a real challenge for the decoder hardware realization, which has to fulfill the specified throughput of 255MBit/s for base station applications.*

*In this paper we will show, to the best of our knowledge, the first published IP LDPC decoder core for the DVB-S2 standard. We present a synthesizable IP block based on ST Microelectronics 0.13 $\mu$ m CMOS technology.*

### 1 Introduction

The new DVB-S2 standard [1] features a powerful forward error correction (FEC) system which enables transmission close to the theoretical limit (Shannon limit). This is enabled by using Low-Density Parity-Check (LDPC) codes [2] one of the most powerful codes known today which can even outperform Turbo-Codes [3]. To provide flexibility 11 different code rates ranging from ( $R = 1/4$  up to  $9/10$ ) are specified with a codeword length up to 64800 bits. This huge maximum codeword length is the reason for the outstanding communications performance ( $\sim 0.7$ dB to Shannon) of this DVB-S2 LDPC code proposal, so in this paper we only focus on the codeword length of 64800 bits. To yield this performance, the decoder has to iterate 30 times. At each iteration up to 300 000 data are scrambled and calculated. This huge data processing, storage and network/interconnect requirements is a real challenge for the decoder realization.

A LDPC code can be represented by a bipartite graph. For the DVB-S2 code 64800 so called variable nodes (VN) and  $64800 * (1 - R)$  check nodes (CN) exist. The connectivity of these two type of nodes is specified in the standard [1]. For decoding the LDPC code messages are exchanged iteratively between this two type of nodes, while the node processing is of low complexity. Within one iteration first the variable nodes are processed, then the check nodes.

For a fully parallel hardware realization each node is instantiated and the connections between them are hard-wired. This was shown in [4] for a 1024 bit LDPC code. But even for this relatively short block length severe routing congestion problems exist. Therefore a partly parallel architecture becomes mandatory for larger block length, where only a subset of nodes are instantiated. A network has to provide the required connectivity between VN and CN nodes. But realizing any permutation pattern is very costly in terms of area, delay and power.

To avoid this problem a decoder first design approach was presented in [5]. First an architecture is specified and afterwards a code is designed which fits this architecture. This approach is only suitable for regular LDPC code where each VN has the same number of incident edges, the CN respectively. But for an improved communications performance so called irregular LDPC codes are mandatory [6], where the VN nodes are of varying degrees. This is the case for the DVB-S2 code. In [7] we have presented a design method for irregular LDPC codes which can be efficiently processed by the decoder hardware. We used so called irregular repeat accumulate (IRA) codes [8] which are within the class of LDPC codes with the advantage of a very simple (linear) encoding complexity. In general, LDPC code encoder are very difficult to implement due to the inherent complex encoding scheme.

The LDPC codes as defined in the DVB-S2 standard are IRA codes, thus the encoder realization is straight forward. Furthermore, the DVB-S2 code shows regularities which can be exploited for an efficient hardware realization.

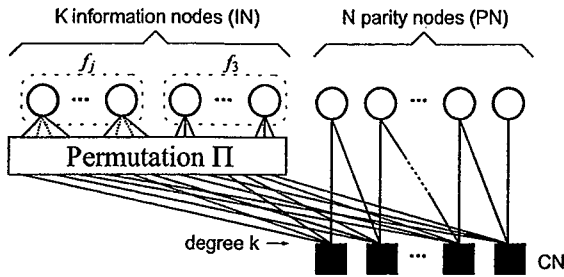


Figure 1. Tanner graph for the DVB-S2 LDPC code

These regularities are also the base for our methodology introduced in [7].

In this paper we show how to exploit these regularities and present an efficient mapping of VN and CN nodes to hardware instances. Memory area and access conflicts are most critical in this mapping process. Thus we used simulated annealing to minimize memory requirements and avoidance of RAM access conflicts.

We present to the best of our knowledge the first IP core capable to process all specified code rates in the DVB-S2 standard. We show synthesis results using a 0.13 \$\mu\$m technology.

The paper is structured as follows: the DVB-S2 LDPC codes and the decoding algorithm are presented in Section 2. In Section 3 the mapping of nodes to hardware instances is explained. The overall decoder architecture is shown in Section 4. Section 5 gives synthesis results and Section 6 concludes the paper.

## 2 DVB-S2 LDPC Codes

LDPC codes are linear block codes defined by a sparse binary matrix (parity check matrix)  $H$ . The set of valid codewords  $x \in C$  have to satisfy

$$Hx^T = 0, \quad \forall x \in C. \quad (1)$$

A column in  $H$  is associated to a bit of the codeword and each row corresponds to a parity check. A nonzero element in a row means that the corresponding bit contributes to this parity check. The code can best be described by a Tanner graph [6], a graphical representation of the associations between code bits and parity checks. Code bits are shown as variable nodes (circles), and parity checks as check nodes (squares), with edges connecting them. The number of edges on each node is called the node degree. If the node degree is identical for all variable nodes, the corresponding parity check matrix is called regular, otherwise it's irregular.

By carefully inspection of the construction rules, the DVB-S2 parity check matrix consists of two distinctive

Rate	j	$f_j$	$f_3$	k	N	K
1/4	12	5400	10800	4	49600	16200
1/3	12	7200	14400	5	43200	21600
2/5	12	8640	17280	6	38880	25920
1/2	8	12960	19440	7	32400	32400
3/5	12	12960	25920	11	25920	38880
2/3	13	4320	38880	10	21600	43200
3/4	12	5400	43200	14	16200	48600
4/5	11	6480	45360	18	12960	51840
5/6	13	5400	48600	22	10800	54000
8/9	4	7200	50400	27	7200	57600
9/10	4	6480	51840	30	6480	58320

Table 1. Parameters describing the DVB-S2 LDPC Tanner graph for different coderates

parts: a random part dedicated to the systematic information, and a fixed part that belongs to the parity information. The Tanner graph for DVB-S2 is shown in Figure 1. There exist two types of variable nodes, the information (IN) and parity nodes (PN), corresponding to the systematic and parity bits respectively. The permutation  $\Pi$  represents the random matrix part of the connectivity between IN and CN nodes, while the PN nodes are all of degree two and are connected in a fixed zigzag pattern to the CN nodes. The  $N$  check nodes have a constant degree  $k$ . The  $K$  information nodes consist of two subsets  $f_j$  and  $f_3$ , with  $f$  the number of IN nodes of degree  $j$  and 3. Table 1 summarizes the code rate dependent parameters as defined in the standard [1].

The connectivity of the IN and CN nodes is defined by the DVB-S2 encoding rule

$$p_j = p_j \oplus i_m, \quad j = (x + q(m \bmod 360)) \bmod N. \quad (2)$$

$p_j$  is the  $j$ th parity bit,  $i_m$  the  $m$ th information code bit, and  $x$ ,  $q$ , and  $N$  are code rate dependent parameters specified by the DVB-S2 standard. Equation 2 determines the entries of the parity check matrix. The  $m$ th column has nonzero elements in each row  $j$ , thus the permutation  $\Pi$  generates one edge between every CN  $m$  and IN  $j$ .

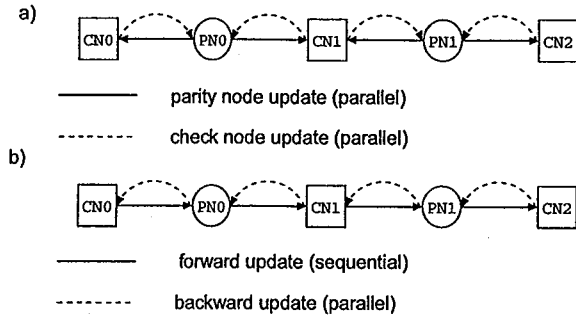
The fixed zigzag connectivity of the PN and CN nodes is defined by the encoding scheme:

$$p_j = p_j \oplus p_{j-1}, \quad j = 1, 2, \dots, N-1. \quad (3)$$

This is a simple accumulator. The corresponding part of the parity check matrix has two nonzero elements in each column, forming a square banded matrix. This type of LDPC codes with this simple encoding procedure are also called irregular repeat accumulate (IRA) codes [8].

### 2.1 Decoding Algorithm

LDPC codes can be decoded using the message passing algorithm [2]. It exchanges soft-information iteratively



**Figure 2. a) conventional message update scheme b) optimized message update scheme**

between the variable and check nodes. The update of the nodes can be done with a canonical scheduling [2]: In the first step all variable nodes must be updated, in the second step all check nodes respectively. The processing of individual nodes within one step is independent, and can thus be parallelized.

The exchanged messages are assumed to be log-likelihood ratios (LLR). Each variable node of degree  $i$  calculates an update of message  $k$  according to:

$$\lambda_k = \lambda_{ch} + \sum_{l=0, l \neq k}^{i-1} \lambda_l, \quad (4)$$

with  $\lambda_{ch}$  the corresponding channel LLR of the VN and  $\lambda_l$  the LLRs of the incident edges. The check node LLR updates are calculated according to

$$\tanh(\lambda_k/2) = \prod_{l=0, l \neq k}^{i-1} \tanh(\lambda_l/2). \quad (5)$$

For fixed-point implementations it was shown in [9] that the total quantization loss is  $\leq 0.1$ db when using a 6 bit message quantization compared to infinite precision. For a 5 bit message quantization the loss is 0.1-0.2 dB [6].

## 2.2 Optimized update of degree 2 Parity Nodes

The DVB standard supports LDPC codes ranging from code rate  $R = 1/4$  to  $R = 9/10$ . Each code has one common property: the connectivity of the check nodes caused by the accumulator of the encoder.  $CN_0$  is always connected to  $CN_1$  by a variable node of degree 2 and so on for all CN nodes. A variable node of degree 2 has the property that the input of the first incident edge is the output of the second incident edge (plus the received channel value) and vice versa. For a sequential processing of the check nodes (e.g. from left to right in Figure 1) an already updated message can directly passed to the next check node due to the simple zigzag connectivity. This immediate message update changes the con-

Rate	q	$E_{PN}$	$E_{IN}$	Addr
1/4	135	97199	97200	270
1/3	120	86399	129600	360
2/5	108	77759	155520	432
1/2	90	64799	162000	450
3/5	72	51839	233280	648
2/3	60	43199	172800	480
3/4	45	32399	194400	540
4/5	36	25919	207360	576
5/6	30	21599	216000	600
8/9	20	14399	180000	500
9/10	18	12959	181440	504

**Table 2. Code rate dependent parameters, with  $E$  the number of incident edges of IN and PN nodes and Addr the number of values required to store the code structure**

ventional update scheme between CN and VN nodes (Equation 4).

The difference in the update scheme is presented in Figure 2. Only the connectivity between check nodes and parity nodes is depicted, the incident edges from the information nodes are omitted. Figure 2a) shows the standard belief propagation with the two phase update. In the first phase all messages from the PN to CN nodes are updated, in the second phase the messages from CN to PN nodes respectively. The message update within one phase is commutative and can be fully parallelized. Figure 2b) shows our new message update scheme in which the new CN message is directly passed to the next CN node. This data flow is denoted as a forward update and corresponds to a sequential message update. The backwards update from the PN to CN nodes is again a parallel update. Note that a sequential backwards update would result in a maximum a posteriori (MAP) algorithm.

This new update scheme improves the communications performance. For the same communications performance 10 iterations can be saved i.e. 30 iterations instead of 40 have to be used. Furthermore we need to store only one message instead of two messages for the next iteration, which is explained in more detail in Section 4.

## 3 Hardware mapping

As already mentioned only partly parallel architectures are feasible. Hence only a subset  $P$  of the nodes are instantiated. The variable and check nodes have to be mapped on these  $P$  functional units. All messages have to be stored during the iterative process, while taking care of RAM access conflicts. Furthermore we need a permutation networks which provides the connectivity of the Tanner graph. -



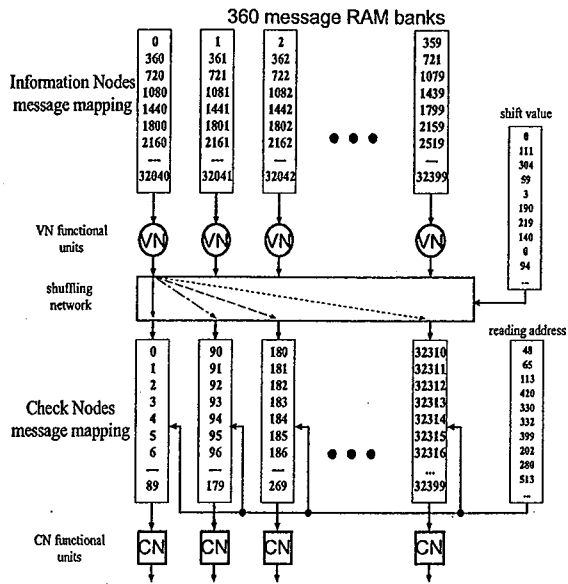


Figure 3. Message and functional unit mapping for  $R = 1/2$

We can split the set of edges  $E$  connecting the check nodes in two subsets  $E_{IN}$  and  $E_{PN}$ , indicating the connections between CN/IN nodes and CN/PN nodes respectively. The subsets are shown in Table 2 for each code rate. Furthermore the  $q$  factor of Equation 2 is listed. The implementation of  $E_{IN}$  is the challenging part, since this connectivity ( $\Pi$ ) changes for each code rate. The realization of  $E_{PN}$  is straightforward, thus we focus on the mapping of the IN and CN nodes.

Due to the varying node degrees the functional nodes process all incoming messages in a serial manner. Thus a functional node can accept one message per clock cycle and produces at most one updated message per clock cycle.

A careful analysis of Equation 2 shows that the connectivity of 360 edges of distinct information nodes are determined by just one value  $x$ , while  $q$  is a code rate dependent constant, see Table 2.

These 360 edges can be processed simultaneously by  $P = 360$  functional units. Within a half iteration a functional unit has to process  $q * (k - 2)$  edges.  $(k - 2)$  is the number of edges between one check node and information nodes. For each code rate  $q$  was chosen to satisfy the constraint

$$E_{IN}/360 = q * (k - 2). \tag{6}$$

It guarantees that each functional unit has to process the same amount of nodes which simplifies the node mapping. Figure 3 shows the mapping of the IN and CN nodes for the LDPC code of rate  $R = 1/2$ . Always 360 consecutive VN nodes are mapped to 360 functional units. To each func-

tional unit a RAM is associated to hold the corresponding messages (edges). Please note that for each IN of degree 8, 8 storage places are allocated to this VN, because each incident edge has to be stored.

The check nodes mapping depends on the rate dependent factor  $q$ . For  $R = 1/2$  the first  $q = 90$  CN nodes are mapped to the first functional unit. The next 90 CN nodes are mapped to the next producer and so on. Again the CN number corresponds to CN degree storage locations.

This node mapping is the key for an efficient hardware realization, since it enables to use a simple shuffling network to provide the connectivity of the Tanner graph. The shuffling network ensures that at each cycle 360 input messages are shuffled to 360 distinct target memories. Thus we have to store  $E_{IN}/390 = 450$  shuffling and addressing information for the  $R = 1/2$  code, see Table 2 for the other code rates. The shuffling offsets and addresses can be extracted from the  $x$  tables provided by [1].

#### 4 Decoder Architecture

Based on the message mapping described in the previous chapter, the basic architecture of the DVB-S2 LDPC decoder is shown in Figure 4. It consists of functional units which can process the functionality of variable and check nodes. This is possible, since only one type of the node are processed during one half iteration. The IN message memories banks hold the messages which are exchanged between information and check nodes. Furthermore we have memories for storing the exchanged messages for the parity nodes (PN message memories), which are all of degree two. The address and shuffling RAM together with the shuffling network provides the connectivity of the Tanner graph.

As mentioned the decoder processes 360 nodes in parallel so 360 messages have to be provided per cycle. All 360 messages are read from the same address from the IN message memory bank. Though, for the information node processing we just increment the reading address. The functional unit can accept each clock cycle new data, while a control flag just labels the last message belonging to a node and starts the output processing. The newly produced 360 messages are then written back to the same address location but with a cyclic shift, caused by the shuffling network. To process the check nodes we have to read from dedicated addresses, provided by the address RAM. These addresses were extracted from node mapping as described in the previous chapter. Again 360 messages are read per clock cycle and written back to the same address after the processing via the shuffling network. This ensures that the messages are shuffled back to their original position.

The processing of the parity nodes can be done concurrently during the check node processing, by using the update scheme described in Section 2.2. Each functional

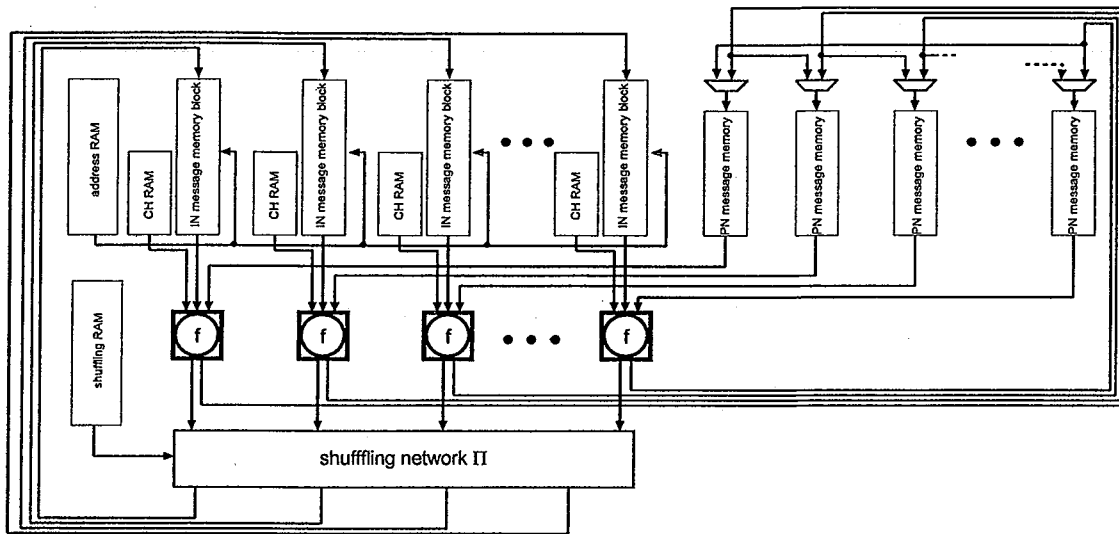


Figure 4. Basic architecture of our LDPC decoder

unit processes consecutive check nodes (Figure 3). The message which is passed during the forward update of the check nodes is kept in the functional unit. Only the messages of the backward update has to be stored which reduces the memory requirements for the zigzag connectivity to  $E_{PN}/2$  messages. The PN message memories are only read and written during the check node phase, while the channel (CH) RAMs delivers the corresponding received channel value.

We use single port SRAMs due to area and power efficiency. Hence we have to take care of read/write conflicts during the iterative process. Read/write conflicts occur, since data are continuously read from the 360 RAMs and provided to the functional units, while new processed messages have to be written back.

The check node processing is the most critical part. We have to read from dedicated addresses extracted during the mapping process. Therefore, the IN message memory block is partitioned in 4 RAMs which is shown in Figure 5. Even if the commutativity of the message processing within a check node is exploited all write conflicts can not be avoided. Therefore a buffer is required to hold a message if writing is not possible due to a conflict. We use simulated annealing to find the best addressing scheme to reduce RAM access conflicts and hence to minimize the buffer overhead. This optimization step ensures that only one buffer is required which holds for all code rates. Per clock cycle we read data from one RAM, and write at most 2 data back to two distinct RAMs, coming from the buffers or the shuffling network. The two least significant bits of the addresses determines the assignment to a partition. This

allows a simple control flow, which just has to compare the reading and the writing addresses of the current clock cycle.

The resulting decoder throughput  $T$  is

$$T = \frac{I}{\#cyc} \cdot f_{cyc}, \quad (7)$$

with  $I$  the number of information bits to be decoded and  $\#cyc$  the number of cycles to decode one block including the input/output (I/O) processing.

The number of cycles is calculated as  $\frac{C}{P_{IO}} + It \cdot \left(2 \cdot \frac{E_{IN}}{P}\right)$ . Thus Equation 7 yields:

$$T = \frac{I}{\frac{C}{P_{IO}} + It \cdot \left(2 \cdot \left(\frac{E_{IN}}{P} + T_{latency}\right)\right)} \cdot f_{cycle}. \quad (8)$$

The part  $\frac{C}{P_{IO}}$  is the number of cycles for input/output (I/O) processing. The decoder is capable to receive 10 channel values per clock cycle. Reading a new codeword of size  $C$  and writing the result of the prior processed block can be done in parallel with reading/writing  $P_{IO}$  data concurrently. The latency  $T_{latency}$  for each iteration depends on the processing units and the shuffling network.

## 5 Results

The LDPC decoder is implemented as a synthesizable VHDL model. Results are obtained with the Synopsis Design Compiler based on a ST Microelectronics  $0.13\mu m$  CMOS technology. The maximum clock frequency is 270 MHz under worst case conditions. The decoder is capable to process all specified code rates of the DVB standard with



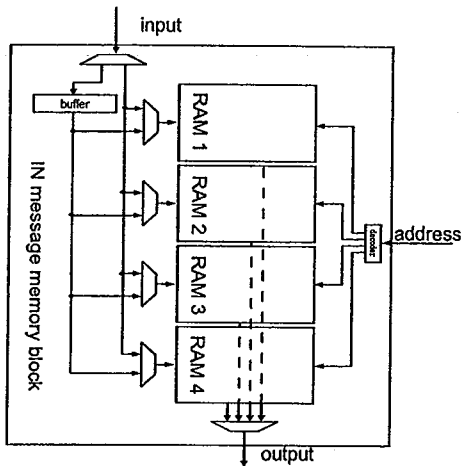


Figure 5. Hierarchical RAM structure

the required throughput of 255Mbit/s. 30 iterations are assumed.

Table 3 shows the synthesis results for a 6 bit message quantization of the channel values and the exchanged messages. The overall area is  $22.74mm^2$ . The area is determined by different code rates.

$R = 1/4$  has the largest set of parity nodes and defines the size of the PN message memories. While the rate  $R = 3/5$  has the most edges to the information nodes and hence determines the size of the IN message memory banks. The size of a functional node depends on the maximum IN and PN degree, respectively ( $R = 2/3$  and  $R = 9/10$ ).

The area is splitted in three parts: RAMs, logic and the shuffling network. Storing the messages yields the major part of the RAM area with  $9.12mm^2$ . It is important to note, that only an area of  $0.075mm^2$  is required to store the connectivity of the Tanner graph. This shows the efficiency of our architectural approach. The logic area of the functional nodes with  $10.8mm^2$  is a major part of the overall area. This is due to the required flexibility of the different code rates. We also placed and routed the shuffling network to test routing congestions. Due to its regularity no congestions resulted, its area is dominated by the logic cells.

## 6 Conclusion

Low-Density Parity-Check codes are part of the new DVB-S2 standard. In this paper we presented to the best of our knowledge the first published IP core for DVB-S2 LDPC decoding. We explained how to explore the code structure for an efficient hardware mapping and presented a decoder architecture which can process all specified code rates ranging from  $R = 1/4$  to  $R = 9/10$ .

0.13 $\mu m$ technologie		AREA [mm <sup>2</sup> ]
RAMs	channel LLRs	1.997
	Messages	9.117
	Address/Shuffling	0.075
Logic	Functional Nodes	10.8
	control logic	0.2
Shuffling Network		0.55
Total Area [mm <sup>2</sup> ]		22.739

Table 3. Synthesis Results for the DVB-S2 LDPC code decoder

## 7 Acknowledgments

The work presented in this paper was supported by the European IST project 4More 4G MC-CDMA multiple antenna system On chip for Radio Enhancements [10].

Our special thanks goes to Friedbert Berens from the Advanced System Technology Group of STM, Geneva, Switzerland, for many valuable discussions.

## References

- [1] European Telecommunications Standards Institute (ETSI). Digital Video Broadcasting (DVB) Second generation framing structure for broadband satellite applications; EN 302 307 V1.1.1. [www.dvb.org](http://www.dvb.org).
- [2] R. G. Gallager. *Low-Density Parity-Check Codes*. M.I.T. Press, Cambridge, Massachusetts, 1963.
- [3] C. Berrou. The Ten-Year-Old Turbo Codes are Entering into Service. *IEEE Communications Magazine*, 41:110–116, Aug. 2003.
- [4] A. Blanksby and C. J. Howland. A 690-mW 1-Gb/s, Rate-1/2 Low-Density Parity-Check Code Decoder. *IEEE Journal of Solid-State Circuits*, 37(3):404–412, Mar. 2002.
- [5] E. Boutillon, J. Castura, and F. Kschischang. Decoder-first code design. In *Proc. 2nd International Symposium on Turbo Codes & Related Topics*, pages 459–462, Brest, France, Sept. 2000.
- [6] T. Richardson and R. Urbanke. The Renaissance of Gallager’s Low-Density Parity-Check Codes. *IEEE Communications Magazine*, 41:126–131, Aug. 2003.
- [7] F. Kienle and N. Wehn. Design Methodology for IRA Codes. In *Proc. 2004 Asia South Pacific Design Automation Conference (ASP-DAC '04)*, Yokohama, Japan, Jan. 2004.
- [8] H. Jin, A. Khandekar, and R. McEliece. Irregular Repeat-Accumulate Codes. In *Proc. 2nd International Symposium on Turbo Codes & Related Topics*, pages 1–8, Brest, France, Sept. 2000.
- [9] T. Zhang, Z. Wang, and K. Parhi. On finite precision implementation of low-density parity-check codes decoder. In *Proc. International Symposium on Circuits and Systems (ISCAS '01)*, Antwerp, Belgium, May 2001.
- [10] <http://ist-4more.org>.

## Factorizable modulo $M$ parallel architecture for DVB-S2 LDPC decoding

Marco Gomes, Gabriel Falcão, Vitor Silva, Vitor Ferreira, Alexandre Sengo and Miguel Falcão\*

Instituto de Telecomunicações, Pólo II da Universidade de Coimbra, 3030-290 Coimbra, Portugal

\*Chipidea Microelectrónica S.A., Rua Frederico Ulrich, n. 2650, 4470-605 Moreira da Maia, Portugal

e-mail: marco@co.it.pt, gff@co.it.pt, vitor@co.it.pt, vitorhugo@co.it.pt, sengo@co.it.pt, mfalcao@chipidea.com

**Abstract** — State-of-the-art decoders for DVB-S2 low-density parity-check (LDPC) codes explore semi-parallel architectures based on the periodicity  $M=360$  factor of the special type of LDPC-IRA codes adopted. This paper addresses the generalization of a well known hardware  $M$ -kernel parallel structure and proposes an efficient partitioning by any factor of  $M$ , without addressing overhead and keeping unchanged the efficient message memory mapping scheme. Our method provides a simple and efficient way to reduce the decoder complexity. Synthesizing the decoder for an FPGA from Xilinx shows a minimum throughput above the minimal 90Mbps.

### I. INTRODUCTION

The recent Digital Video Satellite Broadcast Standard (DVB-S2) [1] [2] has adopted a powerful FEC scheme based on the serial concatenation of BCH and Low Density Parity Check (LDPC) codes. This new FEC structure, combined with the adoption of high order modulations (QPSK, 8PSK, 16APSK and 32APSK), is able to provide capacity gains of about 30% over the previous DVB-S standard [2], with the LDPC codes playing a fundamental role in this raise of performance.

LDPC codes are linear block codes defined by sparse parity-check matrices [3] [4] [5],  $\mathbf{H}$  and, usually, represented by Tanner graphs [6]. A Tanner graph is a bi-partite graph formed by two types of nodes. Check nodes ( $v^c$ ), one per each code constraint, and bit nodes one per each codeword bit (information and parity, respectively,  $v^i$  and  $v^p$ ), with the connection edges between them being given by  $\mathbf{H}$ .

They are decoded using low complexity iterative belief propagation algorithms operating over the Tanner graph description [7]. However, a major drawback is their high encoding complexity caused by the fact that the generator matrix,  $\mathbf{G}$ , is, in general, not sparse. In order to overcome this problem, DVB-S2 standard has adopted a special class of LDPC codes, with linear encoding complexity, known by Irregular Repeat-Accumulate (IRA) [8] [9].

An important issue in the design of LDPC encoder and decoder architectures for DVB-S2 is the fact that the standard supports two different frame lengths (16200 bits for low delay applications and 64800 bits otherwise) and a set of different code rates ( $1/4$ ,  $1/3$ ,  $2/5$ ,  $1/2$ ,  $3/5$ ,  $2/3$ ,  $3/4$ ,  $4/5$ ,  $5/6$ ,  $8/9$  and  $9/10$ ) for both frame lengths and different modulation schemes [1] [9]. For each mode of operation is defined a different LDPC code and, although they share a similar structure and properties, this still poses

an enormous challenge on the development of an encoder and a decoder fully compliant with all operating modes.

The decoder state-of-the-art is based on a flexible partial parallel architecture that explores the  $M=360$  periodicity nature of DVB-S2 LDPC codes [10]. Although capable of providing a throughput far above from the minimum mandatory rate of 90 Mbps, this architecture requires a huge ASIC area of  $22.74 \text{ mm}^2$  on a ST Microelectronics  $0.13 \mu\text{m}$  technology, mainly due to the high number (360) of computation kernels or functional units (FU) and the wide length of the barrel shifter. In order to decrease the number of computation kernels to only 45 FU's and to reduce the length of the barrel shifter, an alternative solution was proposed [11] which uses a re-structured version of  $\mathbf{H}$ . As a consequence, this approach increases the complexity of the DVB-S2 de-interleaver and doubles (almost) the input memory in terms of [10].

In this paper we generalize the architecture [10] and surpass its disadvantages. We will show that it is possible to reduce the number of computation kernels to any integer factor of  $M=360$ , without addressing overhead and keep unchanged the efficient message memory mapping scheme [10]. Our strategy also reduces the length of the barrel shifter by the same factor and considerably simplifies the routing problem. The throughput is reduced by the same factor but this does not represent a real problem since the architecture [10] is able to provide a throughput far above from the mandatory minimum rate. Thus, we provide a simple and efficient method to reduce the decoder complexity without losing the throughput goals.

The next section briefly describes DVB-S2 LDPC-IRA codes. Section III addresses the LDPC decoding for DVB-S2 using a partial parallel architecture and its generalization by sub-sampling it by a factor of  $M$ . Synthesis results are presented in section IV and final conclusions are pointed out in section V.

### II. DVB-S2 LDPC-IRA CODES

The new DVB-S2 [1] [9] standard adopted a special class of LDPC codes known by IRA codes [8] as the main solution for the FEC system. An IRA code is characterized by a parity check matrix,  $\mathbf{H}$ , of the form,

$$\mathbf{H}_{(n-k) \times n} = \left[ \begin{array}{c|c} \mathbf{A}_{(n-k) \times k} & \mathbf{B}_{(n-k) \times (n-k)} \end{array} \right] = \begin{bmatrix} a_{0,0} & a_{0,1} & \dots & a_{0,k-1} & 1 & 0 & \dots & \dots & \dots & 0 \\ a_{1,0} & a_{1,1} & \dots & a_{1,k-1} & 1 & 1 & 0 & \dots & \dots & \vdots \\ \vdots & \vdots & \dots & \vdots & 0 & 1 & 1 & \dots & \dots & \vdots \\ \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \vdots & \dots & 0 & \vdots \\ a_{k-1,0} & a_{k-1,1} & \dots & a_{k-1,k-1} & 0 & \dots & \dots & 1 & 1 & 0 \\ a_{k+1,0} & a_{k+1,1} & \dots & a_{k+1,k-1} & 0 & \dots & \dots & 0 & 1 & 1 \end{bmatrix}, \quad (1)$$

where  $\mathbf{B}$  is a staircase lower triangular matrix. By restricting  $\mathbf{A}$  to be sparse, it is obtained an LDPC-IRA code [9].

The  $\mathbf{H}$  matrices of the DVB-S2 LDPC codes have other properties beyond being of IRA type. Some periodicity constraints were put on the pseudo-random design of the  $\mathbf{A}$  matrices, which allows a significant reduction on the storage requirement without code performance loss.

The matrix  $\mathbf{A}$  construction technique is based on dividing the  $v^l$  nodes in disjoint groups of  $M$  consecutives ones. All the  $v^l$  nodes of a group  $l$  should have the same weight,  $w_l$ , and it is only necessary to choose the  $v^c$  nodes that connect to the first  $v^l$  of the group in order to specify the  $v^c$  nodes that connect to each one of the remaining  $M-1$  nodes. The connection choice for the first element of group  $l$  is pseudo-random with the restrictions that the resulting LDPC code is cycle-4 free, the number of length 6 cycles is the shortest possible and all the  $v^c$  nodes must connect to the same number of  $v^l$  nodes.

Denoting by,  $r_1, r_2, \dots, r_{w_l}$ , the indices of the  $v^c$  nodes that connect to the first  $v^l$  of group  $l$ , the indices of the  $v^c$  nodes that connect to  $v_i^l$ , with  $0 \leq i \leq M-1$ , of group  $l$  can be obtained by,

$$\{(r_1 + i \times q) \bmod (n-k), (r_2 + i \times q) \bmod (n-k), \dots, (r_{w_l} + i \times q) \bmod (n-k)\}, \quad (2)$$

with  $q = (n-k)/M$  and  $M=360$  (a common factor for all DVB-S2 supported codes).

Another property of matrix  $\mathbf{A}$  is that for each supported code, there are a set of groups of  $v^l$  nodes of constant weight  $w > 3$  ( $w$  is code dependent) and the remaining have all weight 3.

### III. DVB-S2 LDPC DECODING

The huge dimensions of the LDPC-IRA codes adopted by the DVB-S2 standard, turns impractical the adoption of a fully parallel architecture that maps the Tanner graph structure [12]. Besides that, such solution is code dependent, which means that is required a different full parallel decoder for each code defined in the standard.

Best known solutions are based on highly vectorized partial parallel architectures [10] [11], that explore the particular characteristics of the DVB-S2 LDPC-IRA codes, namely, the periodic nature ( $M=360$ ) shared by all the codes. One solution was proposed in [10], whose architecture uses  $M$

functional units working in parallel. In this paper we will show that it is possible to reduce the number of functional units by any integer factor of  $M$ , without addressing overhead, keeping unchanged its efficient memory mapping scheme. Our approach does not only surpass the architecture [10] disadvantages, but also makes the architecture flexible and easy reconfigurable according with the decoder constraints.

#### A. Modulo $M$ parallel architecture

As previously described, DVB-S2 adopted a special class of structured LDPC-IRA codes with the properties stated in (2). This turns possible the simultaneous processing of  $v^l$  and  $v^c$  node sets, whose indices are given by,

$$\begin{aligned}
 \mathbf{C}^{(c)} &= \{c, c+1, \dots, c+M-1\}, \text{ with, } c \bmod M = 0, \text{ and} \\
 \mathbf{R}^{(r)} &= \{r, r+q, r+2q, \dots, r+(M-1)q\}, \text{ with, } 0 \leq r \leq q-1, \quad (3)
 \end{aligned}$$

respectively, (the superscript is the index of the first element of the set and, 'r' and 'c' mean row and column of  $\mathbf{H}$ ), which significantly simplifies the decoder control. In fact, according to (2), if  $v_i^l$  is connected to  $v_r^c$ , then,  $v_{r+iq}^c$ , with  $0 \leq i \leq M-1$ , will be connected to,  $v_{c+(i-c) \bmod M}^l$ , where,  $c = M \times (\bar{c} \div M)$  is the index of the first  $v^l$  of the group  $\mathbf{C}^{(c)}$  to which  $v_i^l$  belongs.

The architecture shown in Fig. 1 is based on  $M$  functional units (FU) working in parallel with shared control signals [12], that process both  $v^c$  (in check mode) and  $v^l$  nodes (in bit mode) in a flooding schedule manner [13] [14]. Attending to the zigzag connectivity between  $v^l$  and  $v^c$  nodes, they are updated jointly in check mode following a horizontal schedule approach [15]. A detailed description of the FU operation can be found in [12].

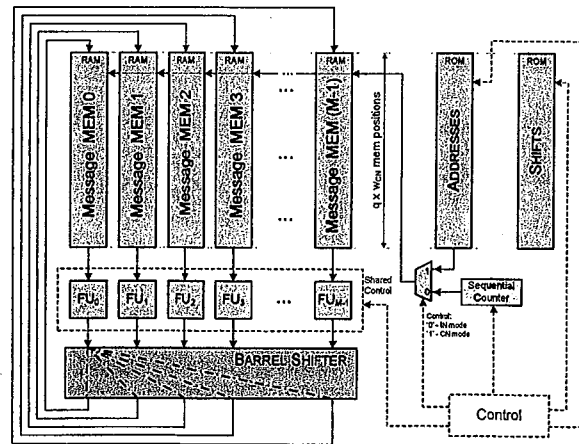


Figure 1. Modulo  $M$  parallel architecture for DVB-S2 LDPC decoding.

#### Memory mapping and shuffling mechanism

As mentioned before, a single FU unit is shared by a constant number of  $v^l$ ,  $v^c$  and  $v^p$  nodes (the last two are

processed jointly), depending on the code length and rate. More precisely, for a  $(n, k)$  DVB-S2 LDPC-IRA code, the FU $_i$ , with  $0 \leq i \leq M-1$ , updates sequentially in bit mode the  $v_{\{i+M, i+2M, \dots, i+(\alpha-1)M\}}^i$  nodes, with  $\alpha = k/M$ . In check mode, the same FU updates the  $v_{\{j, j+1, \dots, j+q-1\}}^c$  and  $v_{\{j, j+1, \dots, j+q-1\}}^p$  nodes, with  $j = i \times q$ . This guarantees that when processing simultaneously the group  $C^{(c)}$ , the computed messages have as destination a set  $R^{(r)}$ , where each one of them will be processed by a different FU. Considering (2), the new computed messages only need to be right rotated to be handled by the correct  $v^c$  nodes. The same happens when processing each  $R^{(r)}$  set, where according to (2), the right rotation must be reversed in order to the new computed messages have as destination the exact  $v^i$  nodes. The shuffling network (barrel shifter) is responsible for the correct message exchange between  $v^c$  and  $v^i$  nodes, emulating the code Tanner graph. The shift values stored in ROM (Fig. 1) can be easily obtained from the annexes B and C of DVB-S2 standard tables [1].

The messages sent along the Tanner graph edges are stored in RAM (see Fig. 1). If we adopt a sequential RAM access in bit mode, then, the access in check mode must be indexed or vice-versa. Both options are valid, so, without loss of generalization, we assume sequential access in bit mode. Denoting by,  $r_i = [r_{i1} \ r_{i2} \ \dots \ r_{i\alpha}]^T$ , the vector of  $v^c$  node indices connected to the  $v_i^i$  node of weight,  $w_i$ , then, the message memory mapping can be obtained using the following matrix,

$$\mathbf{R} = \begin{bmatrix} \mathbf{r}_0 & \mathbf{r}_1 & \dots & \mathbf{r}_{M-1} \\ \mathbf{r}_M & \mathbf{r}_{M+1} & \dots & \mathbf{r}_{2M-1} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{r}_{(q-1)M} & \mathbf{r}_{(q-1)M+1} & \dots & \mathbf{r}_{\alpha M-1} \end{bmatrix}_{(q\alpha w_c) \times M}, \quad (4)$$

where,  $w_c$ , is a code constant ( $v^c$  weight is  $w_c + 2$ , except for the first one (1)).

In order to process each  $R^{(r)}$  set in check mode, the required memory addresses can be obtained by finding the matrix  $\mathbf{R}$  rows where the index  $r$  appears.

### B. Sub-sampling by a factor of $M$

The simplicity of the shuffling mechanism and the efficient memory mapping scheme, constitute the major strengths of the architecture just described [10]. However, the high number of FU's and the long width of the barrel shifter require a huge silicon area. Since this architecture is able to provide a throughput far above from the minimum mandatory rate of 90 Mbps, we may reduce the number of FU's. In fact, we will show that this can be done by any factor of  $M$ .

Let be  $L, N \in \mathbb{N}$  factors of  $M$ , with,  $M = L \times N$ , and consider a  $C^{(c)}$  set (3). This group can be decomposed by down-sampling in  $L$  subgroups as:

$$\begin{aligned} C_0^{(c)} &= \{c, c+L, c+2L, \dots, c+(N-1) \times L\} \\ C_1^{(c)} &= \{c+1, c+1+L, c+1+2L, \dots, c+1+(N-1) \times L\} \\ &\vdots \\ C_{L-1}^{(c)} &= \{c+L-1, c+2L-1, c+3L-1, \dots, c+N \times L-1\} \end{aligned} \quad (5)$$

Each sub-group,  $C_\gamma^{(c)}$ , with  $0 \leq \gamma \leq L-1$ , can be described in terms of the first node of the subgroup (2),  $v_{c+\gamma}^i$ . If  $v_r^c$  is connected to the first information node of the subgroup,  $C_\gamma^{(c)}$ , then,  $v_{\{r+(\beta L+q) \bmod (n-k)\}}^c$  is connect to the  $i$ -th  $v^i$  node, with  $0 \leq i \leq N-1$ , of the referred subgroup.

Equally, the same down-sampling process by  $L$  can be done on each  $R^{(r)}$  group as:

$$\begin{aligned} R_0^{(r)} &= \{r, r+L \times q, r+2L \times q, \dots, r+(N-1) \times L \times q\} \\ R_1^{(r)} &= \{r+q, r+(L+1) \times q, r+(2L+1) \times q, \dots, r+((N-1) \times L+1) \times q\} \\ &\vdots \\ R_{L-1}^{(r)} &= \{r+(L-1) \times q, r+(2L-1) \times q, r+(3L-1) \times q, \dots, r+(N \times L-1) \times q\} \end{aligned} \quad (6)$$

and, in a similar way, each subgroup,  $R_\beta^{(r)}$ , with  $0 \leq \beta \leq L-1$ , can be described just in terms of the first element,  $v_{r+\beta \times q}^c$ . If  $v_c^i$  is connected to the first node of sub-set  $R_\beta^{(r)}$ , then,  $v_{\{c+(\tilde{c}-c+\beta L) \bmod M\}}^i$ , with  $c = M \times (\tilde{c} \div M)$ , is connected to the  $i$ -th  $v^c$ , with  $0 \leq i \leq N-1$ , of the considered subgroup.

From the framework just described in (5) and (6), we conclude that the down-sampling approach preserves the key modulo  $M$  properties and, thus, we can process individually each  $C_\gamma^{(c)}$  and  $R_\beta^{(r)}$  subgroup and the same architecture [10] can be used with only  $N$  processing units as shown in Fig. 2. In fact, when processing simultaneously a group  $C_\gamma^{(c)}$ , the computed messages have as destination a set  $R_\beta^{(r)}$  and, vice-versa.

### Memory mapping and shuffling mechanism

The down-sampling strategy allows a linear reduction (by a factor of  $L$ ) of the hardware resources occupied by the FU's blocks, reduces significantly the complexity of the barrel shifter ( $O(N \log_2 N)$ ) and simplifies the routing problem. Yet, at first glance, it may seem that this strategy implies an increase by  $L$  in the size of the system ROM (*Shifts* and *Addresses*). Fortunately, if we know the properties of the subgroups  $C_0^{(c)}$  and  $R_0^{(r)}$ , we automatically know the properties of the remaining subgroups,  $C_\gamma^{(c)}$  and  $R_\beta^{(r)}$  respectively, with  $0 \leq \gamma, \beta \leq N-1$ . By a proper message memory mapping based on a convenient reshape by  $L$  of the matrix  $\mathbf{R}$  (4), we can keep unchanged the size of the system ROM and compute on the fly the new shifts and addresses values as functions of the ones stored in the ROM of Fig. 2, i.e., for all  $C_0^{(c)}$  and  $R_0^{(r)}$  groups.



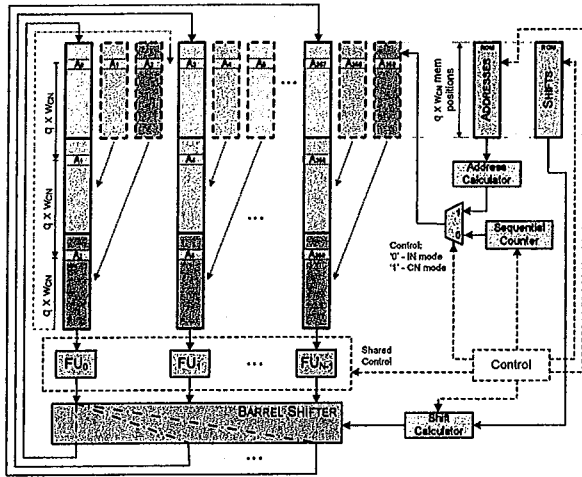


Figure 2. Factorizable modulo  $M$  parallel architecture for DVB-S2 LDPC decoding.

For the configuration shown in Fig. 2, each  $FU_i$ , with  $0 \leq i \leq N-1$ , is now responsible for processing  $L \times \alpha$  information nodes in the following order

$$\begin{aligned} & \{i, i+M, i+2M, \dots, i+(\alpha-1)M\}; \\ & \{i+1, i+1+M, \dots, i+1+(\alpha-1)M\}; \dots; \\ & \{i+L-1, i+L-1+M, \dots, i+L-1+(\alpha-1)M\} \end{aligned} \quad (7)$$

and  $L \times q$  check and parity nodes,  $\{j, j+1, \dots, j+L \times q-1\}$ , with  $j = i \times L \times q$ .

#### IV. SYNTHESIS RESULTS

The architecture of Fig. 2 was synthesized on Virtex-II Pro FPGAs (XC2VP) from Xilinx. For XC2VPxx family it is necessary to use a factor  $L=8$  (45 FU's) due to internal memory limitations. In fact, synthesis results show that it is mandatory to use at least the FPGA XC2VP50 in order to guarantee the minimum memory resources required to implement all code rates and lengths. However, this particular choice uses less than 50% of the FPGA available slices. Using external memory, it would be possible to choose the lower cost FPGA XC2VP30.

The XC2VP100 FPGA allows the implementation of the architecture of Fig. 2 with 90 FUs, which doubles the throughput.

#### V. CONCLUSIONS

This paper addresses the generalization of a state-of-the-art  $M$ -kernel parallel structure for LDPC-IRA DVB-S2 decoding, for any integer factor of  $M=360$  by mean of sub-sampling, keeping unchanged the efficient message memory mapping structure without addressing overheads. This architecture proves to be flexible and easily reconfigurable

according to the decoder constraints and represents a trade off between silicon area and decoder throughput.

Synthesis results show that the implementation of a complete LDPC-IRA DVB-S2 decoder is possible with 45 functional units for Xilinx XC2VP FPGAs family.

#### REFERENCES

- [1] ETSI, *Digital video broadcasting (DVB); Second generation framing structure, channel coding and modulation systems for broadcasting, interactive services, news gathering and other broad-band satellite applications: EN 302 307 V1.1.1*, 2005.
- [2] A. Morello and V. Mignone, "DVB-S2: The second generation standard for satellite broad-band services," *Proceedings of the IEEE*, vol. 94, pp. 210-227, 2006.
- [3] R. G. Gallager, "Low-Density Parity-Check Codes," *Ire Transactions on Information Theory*, vol. 8, pp. 21-&, 1962.
- [4] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Transactions on Information Theory*, vol. 45, pp. 399-431, 1999.
- [5] S. Y. Chung, G. D. Forney, T. J. Richardson, and R. Urbanke, "On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit," *IEEE Communications Letters*, vol. 5, pp. 58-60, 2001.
- [6] R. M. Tanner, "A Recursive Approach to Low Complexity Codes," *IEEE Transactions on Information Theory*, vol. 27, pp. 533-547, 1981.
- [7] J. H. Chen and M. P. C. Fossorier, "Near optimum universal belief propagation based decoding of low-density parity check codes," *IEEE Transactions on Communications*, vol. 50, pp. 406-414, 2002.
- [8] H. Jin, A. Khandekar, and R. McEliece, "Irregular repeat-accumulate codes," In *Proc. 2nd International Symposium on Turbo Codes & Related Topics*, Brest, France, Sept 2000.
- [9] M. Eroz, F. W. Sun, and L. N. Lee, "DVB-S2 low density parity check codes with near Shannon limit performance," *International Journal of Satellite Communications and Networking*, vol. 22, pp. 269-279, 2004.
- [10] F. Kienle, T. Brack, and N. Wehn, "A Synthesizable IP Core for DVB-S2 LDPC Code Decoding," In *Proc. Design, Automation and Test in Europe (DATE'05)*, Munich, Germany, Mar. 2005.
- [11] J. Dielissen, A. Hekstra, and V. Berg, "Low cost LDPC decoder for DVB-S2," In *Proc. Design, automation and test in Europe: Designers' forum (DATE'06)*, Munich, Germany, Mar. 2006.
- [12] M. Gomes, G. Falcão, J. Gonçalves, V. Silva, M. Falcão, and P. Faia, "HDL Library of Processing Units for Generic and DVB-S2 LDPC Decoding," In *Proc. International Conference on Signal Processing and Multimedia Applications (SIGMAP2006)*, Setúbal, Portugal, Aug. 2006.
- [13] J. T. Zhang and M. P. C. Fossorier, "Shuffled iterative decoding," *IEEE Transactions on Communications*, vol. 53, pp. 209-213, 2005.
- [14] H. Xiao and A. H. Banihashemi, "Graph-based message-passing schedules for decoding LDPC codes," *IEEE Transactions on Communications*, vol. 52, pp. 2098-2105, 2004.
- [15] E. Sharon, S. Litsyn, and J. Goldberger, "An efficient message-passing schedule for LDPC decoding," *Electrical and Electronics Engineers in Israel, 2004. Proceedings. 2004 23rd IEEE Convention of*, pp. 223-226, 2004.

# Design of LDPC Codes: A Survey and New Results

Gianluigi Liva, Shumei Song, Lan Lan, Yifei Zhang, Shu Lin, and William E. Ryan

**Abstract**—This survey paper provides fundamentals in the design of LDPC codes. To provide a target for the code designer, we first summarize the EXIT chart technique for determining (near-)optimal degree distributions for LDPC code ensembles. We also demonstrate the simplicity of representing codes by protographs and how this naturally leads to quasi-cyclic LDPC codes. The EXIT chart technique is then extended to the special case of protograph-based LDPC codes. Next, we present several design approaches for LDPC codes which incorporate one or more accumulators, including quasi-cyclic accumulator-based codes. The second half of the paper then surveys several algebraic LDPC code design techniques. First, codes based on finite geometries are discussed and then codes whose designs are based on Reed-Solomon codes are covered. The algebraic designs lead to cyclic, quasi-cyclic, and structured codes. The masking technique for converting regular quasi-cyclic LDPC codes to irregular codes is also presented. Some of these results and codes have not been presented elsewhere. The paper focuses on the binary-input AWGN channel (BI-AWGNC). However, as discussed in the paper, good BI-AWGNC codes tend to be universally good across many channels. Alternatively, the reader may treat this paper as a starting point for extensions to more advanced channels. The paper concludes with a brief discussion of open problems.

## I. INTRODUCTION

The class of low-density parity-check (LDPC) codes represents the leading edge in modern channel coding. They have held the attention of coding theorists and practitioners in the past decade because of their near-capacity performance on a large variety of data transmission and storage channels and because their decoders can be implemented with manageable complexity. They were invented by Gallager in his 1960 doctoral dissertation [1] and were scarcely considered in the 35 years that followed. One notable exception is Tanner, who wrote an important paper in 1981 [2] which generalized LDPC codes and introduced a graphical representation of LDPC codes, now called Tanner graphs. Apparently independent of Gallager's work, LDPC codes were re-invented in the mid-1990's by MacKay, Luby, and others [3][4][5][6] who noticed the advantages of linear block codes which possess sparse (low-density) parity-check matrices.

This paper surveys the state-of-the-art in LDPC code design for binary-input channels while including a few new results as well. While it is tutorial in some aspects, it is not

entirely a tutorial paper, and the reader is expected to be fairly versed on the topic of LDPC codes. Tutorial coverages of LDPC codes can be found in [7][8]. The purpose of this paper is to give the reader a detailed overview of various LDPC code design approaches and also to point the reader to the literature. While our emphasis is on code design for the binary-input AWGN channel (BI-AWGNC), the results in [9][10][11][12] demonstrate that a LDPC code that is good on the BI-AWGNC tends to be universally good and can be expected to be good on most wireless, optical, and storage channels.

We favor code designs which are most appropriate for applications, by which we mean codes which have low-complexity encoding, good waterfall regions, and low error floors. Thus, we discuss quasi-cyclic (QC) codes because their encoders may be implemented by shift-register circuits [13]. We also discuss accumulator-based codes because low-complexity encoding is possible from their parity-check matrices, whether they are quasi-cyclic or not. The code classes discussed tend to be the ones (or related to the ones) used in applications or adopted for standards. Due to time and space limitations, we cannot provide a complete survey. The present survey is biased toward the expertise and interests of the authors.

Before a code can be designed, the code designer needs to know the design target. For this reason, Section II first briefly reviews the belief propagation decoder for LDPC codes and then presents the so-called extrinsic information transfer (EXIT) chart technique for this decoder. The EXIT chart technique allows one to obtain near-optimal parameters for LDPC code ensembles which guide the code designer. The EXIT technique is extended in Section III to the case of codes based on protographs. Section IV considers LDPC codes based on accumulators. The code types treated in that section are: repeat-accumulate, irregular repeat-accumulate, irregular repeat-accumulate-accumulate, generalized irregular repeat-accumulate, and accumulate-repeat-accumulate. That section also gives examples of quasi-cyclic code design using protograph (or base matrix) representations. Section V surveys the literature on cyclic and quasi-cyclic LDPC code design based on finite geometries. Section VI presents several LDPC code design techniques based on Reed-Solomon codes. Section VII presents the masking technique for converting regular QC codes to irregular QC codes to conform to prescribed code parameters. Section VIII contains some concluding remarks and some open problems.

## II. DESIGN VIA EXIT CHARTS

We start with an  $m \times n$  low-density parity-check matrix  $\mathbf{H}$ , which corresponds to a code with design rate  $(n - m)/n$ , which could be less than the actual rate,  $R = k/n$ , where  $k$  is the number of information bits per codeword.  $\mathbf{H}$  gives rise

Manuscript received July 04, 2006; revised August 25, 2006. This work was supported by the University of Bologna, NASA-Goddard, and NSF.

This paper has been approved by F. Chiaraluca.

Gianluigi Liva is with the University of Bologna (email: gliva@deis.unibo.it).

Shumei Song, Lan Lan, and Shu Lin are with the University of California at Davis (e-mail: ssmsong@ece.ucdavis.edu, squashlan@gmail.com, shulin@ece.ucdavis.edu).

Yifei Zhang and William E. Ryan are with the University of Arizona, U.S.A. (e-mail: {yifeiz, ryan}@ece.arizona.edu).



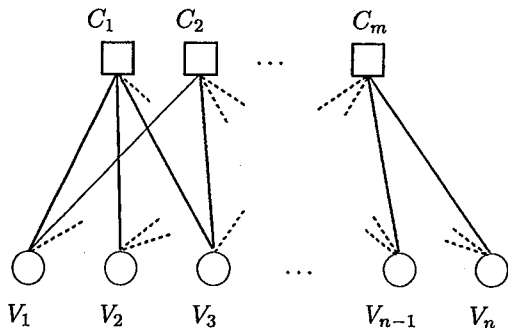


Fig. 1. Tanner graph representation of LDPC codes.

to a Tanner graph which has  $m$  check nodes, one for each row of  $\mathbf{H}$ , and  $n$  variable nodes, one for each column of  $\mathbf{H}$ . Considering the general case in which  $\mathbf{H}$  has non-uniform row and column weight, the Tanner graph can be characterized by degree assignments  $\{d_v(i)\}_{i=1}^n$  and  $\{d_c(j)\}_{j=1}^m$ , where  $d_v(i)$  is the degree of the  $i$ -th variable node and  $d_c(j)$  is the degree of the  $j$ -th check node. Such a graph, depicted in Fig. 1, is representative of the iterative decoder, with each node representing a soft-in/soft-out processor (or node decoder).

We shall assume the BI-AWGNC in our description of the LDPC iterative decoder. In this model, a received channel sample  $y$  is given by  $y = x + w$ , where  $x = (-1)^c \in \{\pm 1\}$  is the bipolar representation of the transmitted code bit  $c \in \{0, 1\}$  and  $w$  is a white Gaussian noise sample distributed as  $\eta(0, \sigma_w^2)$ , where  $\sigma_w^2 = N_0/2$ , following convention. The channel bit log-likelihood ratios (LLRs) are computed as

$$L_{ch} = \log \left( \frac{p(x = +1 | y)}{p(x = -1 | y)} \right) = \frac{2y}{\sigma_w^2}. \quad (1)$$

In one iteration of the conventional, flooding-schedule iterative decoder, the variable node decoders (VNDs) first process their input LLRs and send the computed outputs (messages) to each of their neighboring check node decoders (CNDs); then the CNDs process their input LLRs and send the computed outputs (messages) to each of their neighboring VNDs. More specifically, the message from the  $i$ -th VND to the  $j$ -th CND is

$$L_{i \rightarrow j} = L_{ch,i} + \sum_{j' \neq j} L_{j' \rightarrow i} \quad (2)$$

where  $L_{j' \rightarrow i}$  is the incoming message from CND  $j'$  to VND  $i$  and where the summation is over the  $d_v(i) - 1$  check node neighbors of variable node  $i$ , excluding check node  $j$ . The message from CND  $j$  to VND  $i$  is given by

$$L_{j \rightarrow i} = 2 \tanh^{-1} \left( \prod_{i' \neq i} \tanh(L_{i' \rightarrow j}) \right) \quad (3)$$

where  $L_{i' \rightarrow j}$  is the incoming message from VND  $i'$  to CND  $j$  and where the product is over the  $d_c(j) - 1$  variable node neighbors of check node  $j$ , excluding variable node  $i$ . This decoding algorithm is called the sum-product algorithm (SPA).

We now discuss the EXIT chart technique [14][15][11] for this decoder and channel model. The idea is that the VNDs and the CNDs work cooperatively and iteratively to make bit decisions, with the metric of interest generally improving with each half-iteration. A transfer curve which plots the input metric versus the output metric can be obtained for both the VNDs and the CNDs, where the transfer curve for the VNDs depends on the channel SNR. Further, since the output metric for one processor is the input metric for its companion processor, one can plot both transfer curves on the same axes, but with the abscissa and ordinate reversed for one processor. Such a chart aids in the prediction of the *decoding threshold* of the ensemble of codes characterized by given VN and CN degree distributions: the decoding threshold is the SNR at which the two transfer curves just touch, precluding convergence of the two processors. EXIT chart computations are thus integral to the optimization of Tanner graph node degree distributions for LDPC codes and are the main computation in the optimization process. We emphasize that decoding threshold prediction techniques such as EXIT charts or density evolution [16] assume a graph with no cycles, an infinite codeword length, and an infinite number of decoding iterations.

An EXIT chart example is depicted in Fig. 2 for the ensemble of regular LDPC codes on the BI-AWGNC with  $d_v(i) = d_v = 3$  for  $i = 1, \dots, n$ , and  $d_c(j) = d_c = 6$  for  $j = 1, \dots, m$ . In the figure, the metric used for the transfer curves is extrinsic mutual information, giving rise to the name extrinsic information transfer (EXIT) chart. (The notation used in the figure is explained below.) Also shown in the figure is the decoding trajectory corresponding to these EXIT curves. As the SNR increases, the top curve shifts upwards, increasing the "tunnel" between the two curves and thus the decoder convergence rate. The SNR for this figure is just above the decoding threshold for codes with  $(d_v, d_c) = (3, 6)$ ,  $(E_b/N_0)_{thres} = 1.1$  dB. Other metrics, such as SNR and mean [17][18] and error probability [19] are possible, but mutual information generally gives the most accurate prediction of the decoding threshold [14][20] and is a universally good metric across many channels [9][10][11][12].

To facilitate EXIT chart computations, the following Gaussian assumption is made. First, we note that the LLR  $L_{ch}$  in (1) corresponding to the BI-AWGNC is Gaussian with mean  $\mu_{ch} = 2x/\sigma_w^2$  and variance  $\sigma_{ch}^2 = 4/\sigma_w^2$ . From this and the usual assumption that the all-zeros codeword was transmitted (thus,  $x_i = +1$  for  $i = 1, \dots, n$ ),  $\sigma_{ch}^2 = 2\mu_{ch}$ . This is equivalent to the *symmetric condition* of [16] which states that the conditional pdf of an LLR value  $L$  must satisfy  $p_L(l | x) = p_L(-l | x) e^{xl}$ . Now, it has been observed that under normal operating conditions and after a few iterations, the LLRs  $L_{i \rightarrow j}$  and  $L_{j \rightarrow i}$  are approximately Gaussian and, further, if they are assumed to be symmetric-Gaussian, as is the case for  $L_{ch}$ , the decoding threshold predictions are very accurate (e.g., when compared to the more accurate, but more computationally intensive density evolution results [16]). Moreover, the symmetric-Gaussian assumption vastly

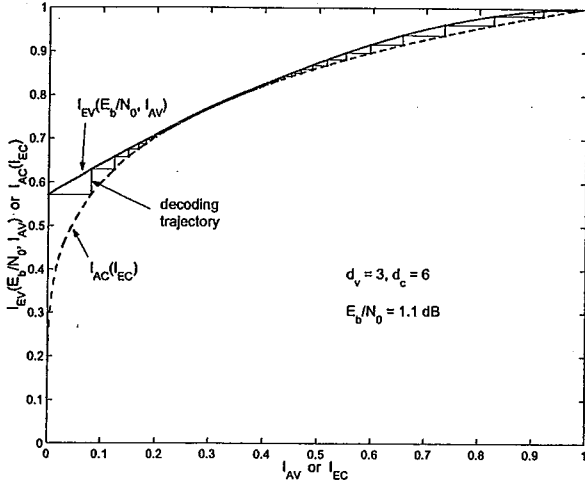


Fig. 2. EXIT chart example for  $(d_v, d_c) = (3, 6)$  regular LDPC code.

simplifies EXIT chart analyses.

We now consider the computation of EXIT transfer curves for both VNDs and the CNDs, first for regular LDPC codes and then for irregular codes. Following [14][15], excluding the inputs from the channel, we consider VND and CND inputs to be *a priori* information, designated by ‘A’, and their outputs to be extrinsic information, designated by ‘E’. Thus, an extrinsic information transfer curve for the VNDs plots the extrinsic information  $I_E$  as a function of its input *a priori* information,  $I_A$ , and similarly for the CNDs.

The VND EXIT curve,  $I_{E,V}$  versus  $I_{A,V}$ , under the symmetric-Gaussian assumption for VND inputs,  $L_{ch,i}$  and  $\{L_{j \rightarrow i}\}$ , and outputs,  $L_{i \rightarrow j}$ , can be obtained as follows. From (2) and an independent-message assumption,  $L_{i \rightarrow j}$  is Gaussian with variance  $\sigma^2 = \sigma_{ch}^2 + (d_v - 1)\sigma_A^2$  (hence, mean  $\sigma^2/2$ ). The mutual information between the random variable  $X$  (corresponding to the realization  $x_i$ ) and the extrinsic LLR  $L_{i \rightarrow j}$  is therefore (for simplicity, we write  $L$  for  $L_{i \rightarrow j}$ ,  $x$  for  $x_i$ , and  $p_L(l | \pm)$  for  $p_L(l | x = \pm 1)$ )

$$\begin{aligned} I_{E,V} &= H(X) - H(X | L) \\ &= 1 - E[\log_2(1/p_{X|L}(x | l))] \\ &= 1 - \sum_{x=\pm 1} \frac{1}{2} \int_{-\infty}^{\infty} p_L(l | x) \\ &\quad \cdot \log_2 \left( \frac{p_L(l | +) + p_L(l | -)}{p_L(l | x)} \right) dl \\ &= 1 - \int_{-\infty}^{\infty} p_L(l | +) \log \left( 1 + \frac{p_L(l | -)}{p_L(l | +)} \right) dl \\ &= 1 - \int_{-\infty}^{\infty} p_L(l | +) \log(1 + e^{-l}) dl \end{aligned}$$

where the last line follows from the symmetry condition and because  $p_L(l | x = -1) = p_L(-l | x = +1)$  for Gaussian densities.

Since  $L_{i \rightarrow j} \sim \eta(\sigma^2/2, \sigma^2)$  (when conditioned on  $x_i =$

+1), we have

$$I_{E,V} = 1 - \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\sigma}} e^{-(l-\sigma^2/2)^2/2\sigma^2} \log(1 + e^{-l}) dl. \quad (4)$$

For convenience we write this as

$$I_{E,V} = J(\sigma) = J\left(\sqrt{(d_v - 1)\sigma_A^2 + \sigma_{ch}^2}\right), \quad (5)$$

following [15]. To plot  $I_{E,V}$  versus  $I_{A,V}$ , where  $I_{A,V}$  is the mutual information between the VND inputs  $L_{j \rightarrow i}$  and the channel bits  $x_i$ , we apply the symmetric-Gaussian assumption to these inputs so that

$$I_{A,V} = J(\sigma_A) \quad (6)$$

and

$$I_{E,V} = J(\sigma) = J\left(\sqrt{(d_v - 1)[J^{-1}(I_{A,V})]^2 + \sigma_{ch}^2}\right). \quad (7)$$

The inverse function  $J^{-1}(\cdot)$  exists since  $J(\sigma_A)$  is monotonic in  $\sigma_A$ . Lastly,  $I_{E,V}$  can be parameterized by  $E_b/N_0$  for a given code rate  $R$  since  $\sigma_{ch}^2 = 4/\sigma_w^2 = 8R(E_b/N_0)$ . Approximations of the functions  $J(\cdot)$  and  $J^{-1}(\cdot)$  are given in [15].

To obtain the CND EXIT curve,  $I_{E,C}$  versus  $I_{A,C}$ , we can proceed as we did in the VND case, e.g., begin with the symmetric-Gaussian assumption. However, this assumption is not sufficient because determining the mean and variance for a CND output  $L_{j \rightarrow i}$  is not straightforward, as is evident from the computation for CNDs in (3). Closed-form expressions have been derived for the check node EXIT curves [21][22]. Computer-based numerical techniques can also be used to obtain these curves. However, the simplest technique exploits the following duality relationship (proven to be exact for the binary erasure channel [11]): the EXIT curve for a degree- $d_c$  check node (i.e., rate- $(d_c - 1)/d_c$  single-parity check (SPC) code) and that of a degree- $d_c$  variable node (i.e., rate- $1/d_c$  repetition code) are related as

$$I_{E,SPC}(d_c, I_A) = 1 - I_{E,REP}(d_c, 1 - I_A).$$

This relationship was shown to be very accurate for the BI-AWGNC in [21][22]. Thus,

$$\begin{aligned} I_{E,C} &= 1 - I_{E,V}(\sigma_{ch} = 0, d_v \leftarrow d_c, I_{A,V} \leftarrow 1 - I_{A,C}) \\ &= 1 - J\left(\sqrt{(d_c - 1)[J^{-1}(1 - I_{A,C})]^2}\right). \quad (8) \end{aligned}$$

For irregular LDPC codes,  $I_{E,V}$  and  $I_{E,C}$  are computed as weighted averages. The weighting is given by the coefficients of the “edge perspective” degree distribution polynomials  $\lambda(z) = \sum_{d=1}^{d_v} \lambda_d z^{d-1}$  and  $\rho(z) = \sum_{d=1}^{d_c} \rho_d z^{d-1}$ , where  $\lambda_d$  is the fraction of edges in the Tanner graph connected to degree- $d$  variable nodes,  $\rho_d$  is the fraction of edges connected to degree- $d$  check nodes, and  $\lambda(1) = \rho(1) = 1$ . Then, for irregular LDPC codes,

$$I_{E,V} = \sum_{d=1}^{d_v} \lambda_d I_{E,V}(d, I_{A,V}) \quad (9)$$

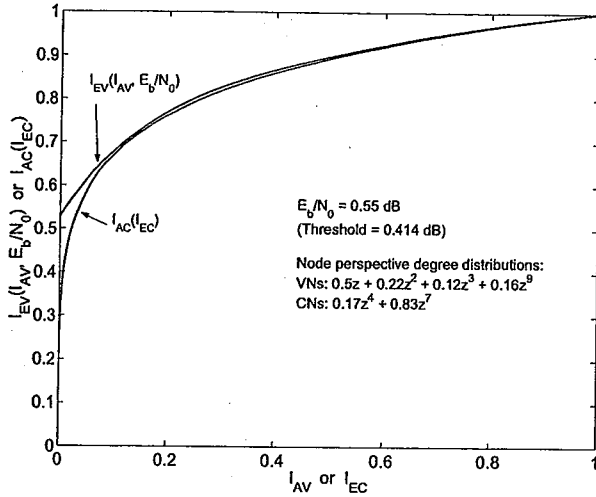


Fig. 3. EXIT chart for rate-1/2 irregular LDPC code. (Ack: S. AbuSurra)

where  $I_{E,V}(d)$  is given by (7) with  $d_v$  replaced by  $d$ , and

$$I_{E,C} = \sum_{d=1}^{d_c} \rho_d I_{E,C}(d, I_{A,C}) \quad (10)$$

where  $I_{E,C}(d)$  is given by (8) with  $d_c$  replaced by  $d$ .

It has been shown [11] that to optimize the decoding threshold on the binary erasure channel, the shapes of the VND and CND transfer curves must be well matched in the sense that the CND curve fits inside the VND curve (an example will follow). This situation has also been observed on the BI-AWGNC [15]. Further, to achieve a good match, the number of different VN degrees need only be about 3 or 4 and the number of different CN degrees need only be 1 or 2.

*Example 1:* We consider the design of a rate-1/2 irregular LDPC code with four possible VN degrees and two possible CN degrees. Given that  $\lambda(1) = \rho(1) = 1$  and  $R = 1 - \int_0^1 \rho(z) dz / \int_0^1 \lambda(z) dz$  [16],[4], only two of the four coefficients for  $\lambda(z)$  need be specified and only one of the two for  $\rho(z)$  need be specified. A non-exhaustive search yielded  $\lambda(z) = 0.267z + 0.176z^2 + 0.127z^3 + 0.430z^9$  and  $\rho(z) = 0.113z^4 + 0.887z^7$  with a decoding threshold of  $(E_b/N_0)_{thres} = 0.414$  dB. The EXIT chart for  $E_b/N_0 = 0.55$  dB is presented in Fig. 3. The figure also gives the "node perspective" degree distribution information. □

The references contain additional information on EXIT charts, including the so-called area property, EXIT charts for the Rayleigh channel, for higher-order modulation, and for multi-input/multi-output channels [14][15][11][23].

### III. DESIGN OF PROTOGRAPH-BASED CODES

#### A. Definition and Problem Statement

A protograph [24][25][26][27] is a relatively small bipartite graph from which a larger graph can be obtained by a copy-and-permute procedure: the protograph is copied  $Q$  times,

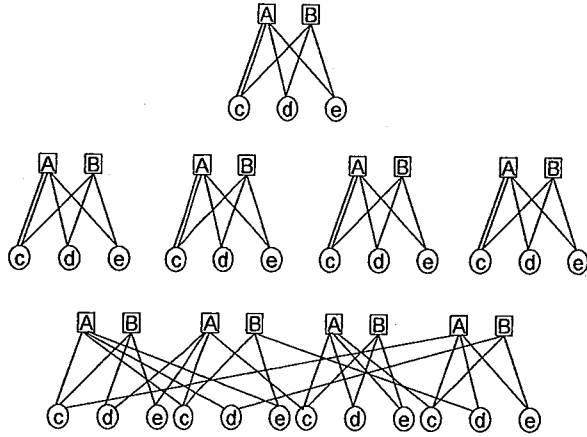


Fig. 4. Illustration of the protograph copy and permute procedure with  $q = 4$  copies.

and then the edges of the individual replicas are permuted among the replicas (under restrictions described below) to obtain a single, large graph. An example is presented in Fig. 4. The permuted edge connections are specified by the parity-check matrix  $H$ . Note that the edge permutations cannot be arbitrary. In particular, the nodes of the protograph are labeled so that if variable node  $V$  is connected to check node  $C$  in the protograph, then variable node  $V$  in a replica can only connect to one of the  $Q$  replicated  $C$  check nodes. Doing so preserves the decoding threshold properties of the protograph. A protograph can possess parallel edges, i.e., two nodes can be connected by more than one edge. For LDPC codes, the copy-and-permute procedure must eliminate such parallel connections in order to obtain a derived graph appropriate for a parity-check matrix.

It is convenient to choose the parity-check matrix  $H$  as an  $M \times N$  array of  $Q \times Q$  (weight-one) circulant permutation matrices (some of which may be the  $Q \times Q$  zero matrix). When  $H$  is an array of circulants, the LDPC code will be quasi-cyclic. Such a structure has a favorable impact on both the encoder and the decoder. The encoder for QC codes can be implemented with shift-register circuits with complexity linearly proportional to  $m$  for serial encoding and to  $n$  for parallel encoding [13]. By contrast, encoders for unstructured LDPC codes require much more work. The decoder for QC LDPC codes can be implemented in a modular fashion by exploiting the circulant-array structure of  $H$  [28][29].

Below we present an extension of the EXIT approach to codes defined by protographs. This extension is a multi-dimensional numerical technique and as such does not have a two-dimensional EXIT chart representation of the iterative decoding procedure. Still, the technique yields decoding thresholds for LDPC code ensembles specified by protographs. This multi-dimensional technique is facilitated by the relatively small size of protographs and permits the analysis of protograph code ensembles characterized by the presence of *critical node types*, i.e., node types which can lead to failed

EXIT-based convergence of code ensembles. Examples of critical node types are degree-1 variable nodes and punctured variable nodes.

A code ensemble specified by a protograph is a refinement (sub-ensemble) of a code ensemble specified simply by the protograph's (hence, LDPC code's) degree distributions. To demonstrate this, we introduce the adjacency matrix  $\mathbf{B} = [b_{ji}]$  for a protograph, also called a base matrix [25], where  $b_{ji}$  is the number of edges between CN  $j$  and VN  $i$ . As an example, for the protograph at the top of Fig. 4,

$$\mathbf{B} = \begin{pmatrix} 2 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}.$$

Consider also an alternative protograph and base matrix specified by

$$\mathbf{B}' = \begin{pmatrix} 2 & 0 & 2 \\ 1 & 2 & 0 \end{pmatrix}.$$

The degree distributions of both of these protographs are identical and are easily seen to be

$$\begin{aligned} \lambda(z) &= \frac{4}{7}z + \frac{3}{7}z^2 \\ \rho(z) &= \frac{3}{7}z^2 + \frac{4}{7}z^3. \end{aligned}$$

However, the ensemble corresponding to  $\mathbf{B}$  has a threshold of  $E_b/N_0 = 0.78$  dB and that corresponding to  $\mathbf{B}'$  has a threshold at 0.83 dB. (For reference, density evolution [16] applied to the above degree distributions gives 0.817 dB.)

As another example, let

$$\mathbf{B} = \begin{pmatrix} 1 & 2 & 1 & 1 & 0 \\ 2 & 1 & 1 & 1 & 0 \\ 1 & 2 & 0 & 0 & 1 \end{pmatrix}$$

and

$$\mathbf{B}' = \begin{pmatrix} 1 & 3 & 1 & 0 & 0 \\ 2 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 \end{pmatrix},$$

noting that they have identical degree distributions. We also puncture the bits corresponding to the second column in each base matrix. Using the multidimensional EXIT algorithm described below, the thresholds for  $\mathbf{B}$  and  $\mathbf{B}'$  in this case were computed to be 0.48 dB and  $+\infty$ , respectively.

Thus, standard EXIT analysis based on degree distributions is inadequate for protograph-based LDPC code design. In fact, the presence of degree-1 variable nodes as in our second example implies that there is a term in the summation in (9) of the form

$$\lambda_1 I_{E,V}(1, I_{A,V}) = J(\sigma_{ch}).$$

Since  $J(\sigma_{ch})$  is always less than one for  $0 < \sigma_{ch} < \infty$  and since  $\sum_{d=1}^{d_v} \lambda_d = 1$ , the summation in (9), that is,  $I_{E,V}$ , will be strictly less than one. Again, standard EXIT analysis implies failed convergence for codes with the same degree distributions as  $\mathbf{B}$  and  $\mathbf{B}'$ . This is in contrast with the fact that codes in the  $\mathbf{B}$  ensemble do converge when the SNR exceeds the threshold of 0.48 dB.

In the following, a multidimensional EXIT technique [30][31] will be presented which overcomes this issue and allows the determination of the decoding threshold for codes based on protographs (possibly with punctured nodes).

### B. Multidimensional EXIT Analysis

The algorithm presented in [30][31] eliminates the average in (9) and considers the propagation of the messages on a decoding tree which is specified by the protograph of the ensemble. Let  $\mathbf{B} = [b_{ji}]$  be the  $M \times N$  base matrix for the protograph under analysis. Let  $I_{E,V}^{i \rightarrow j}$  be the extrinsic mutual information between code bits associated with "type  $i$ " VNs and the LLRs  $L_{i \rightarrow j}$  sent from these VNs to "type  $j$ " CNs. Similarly, let  $I_{E,C}^{j \rightarrow i}$  be the extrinsic mutual information between code bits associated with "type  $i$ " VNs and the LLRs  $L_{j \rightarrow i}$  sent from "type  $j$ " CNs to these VNs. Then, because  $I_{E,C}^{j \rightarrow i}$  acts as *a priori* mutual information in the calculation of  $I_{E,V}^{i \rightarrow j}$ , following (7) we have (given an edge exists between CN  $j$  and VN  $i$ , i.e., given  $b_{ji} \neq 0$ )

$$I_{E,V}^{i \rightarrow j} = J \left( \sqrt{\sum_{c=1}^M (b_{ci} - \delta_{cj}) \left( J^{-1}(I_{E,C}^{c \rightarrow i}) \right)^2 + \sigma_{ch,i}^2} \right), \quad (11)$$

where  $\delta_{cj} = 1$  when  $c = j$  and  $\delta_{cj} = 0$  when  $c \neq j$ .  $\sigma_{ch,i}^2$  is set to zero if code bit  $i$  is punctured. Similarly, because  $I_{E,V}^{j \rightarrow i}$  acts as *a priori* mutual information in the calculation of  $I_{E,C}^{j \rightarrow i}$ , following (8) we have (when  $b_{ji} \neq 0$ )

$$I_{E,C}^{j \rightarrow i} = 1 - J \left( \sqrt{\sum_{v=1}^N (b_{jv} - \delta_{ci}) \left( J^{-1}(1 - I_{E,V}^{v \rightarrow j}) \right)^2} \right). \quad (12)$$

The multidimensional EXIT algorithm can now be presented as follows.

- 1) *Initialization.* Select  $E_b/N_0$ . Initialize a vector  $\sigma_{ch} = (\sigma_{ch,0}, \dots, \sigma_{ch,N-1})$  such that

$$\sigma_{ch,i} = 8R \left( \frac{E_b}{N_0} \right)_i$$

where  $(E_b/N_0)_i$  equals zero when  $x_i$  is punctured and equals the selected  $E_b/N_0$  otherwise.

- 2) *VN to CN.* For  $i = 0, \dots, N-1$  and  $j = 0, \dots, M-1$ , compute (11).
- 3) *CN to VN.* For  $i = 0, \dots, N-1$  and  $j = 0, \dots, M-1$ , compute (12).
- 4) *Cumulative mutual information.* For  $i = 0, \dots, N-1$ , compute

$$I_{CMI}^i = J \left( \sqrt{\sum_{c=1}^M \left( J^{-1}(I_{E,C}^{c \rightarrow i}) \right)^2 + \sigma_{ch,i}^2} \right).$$

- 5) If  $I_{CMI}^i = 1$  (up to desired precision) for all  $i$ , then stop; otherwise, go to step 2.

This algorithm converges only when the selected  $E_b/N_0$  is above the threshold. Thus, the threshold is the lowest



value of  $E_b/N_0$  for which all  $I_{CMI}^i$  converge to 1. As shown in [30][31], the thresholds computed by this algorithm are typically within 0.05 dB of those computed by density evolution. Recalling that many classes of multi-edge type (MET) [26] LDPC codes rely on simple protographs, the above algorithm provides an accurate threshold estimation for MET ensembles, with a remarkable reduction in computational complexity relative to the density evolution analysis proposed in [26].

#### IV. ACCUMULATOR-BASED CODE DESIGNS

##### A. Repeat-Accumulate Codes

This section provides an overview of the design of LDPC codes that can be considered to be a concatenation of a set of repetition codes with one or more accumulators, through an interleaver. The first example of accumulator-based codes were the so-called repeat-accumulate (RA) codes [32]. Despite their simple structure, they were shown to provide good performance and, more importantly, they paved a path toward the design of efficiently encodable LDPC codes. RA codes and other accumulator-based codes are LDPC codes that can be decoded as serial turbo codes or as LDPC codes.

An RA code consists of a serial concatenation of a single rate- $1/q$  repetition code through an interleaver with an accumulator having transfer function  $1/(1 \oplus D)$ . RA codes can be either non-systematic or systematic. In the first case, the accumulator output,  $\mathbf{p}$ , is the codeword and the code rate is  $1/q$ . For systematic RA codes, the information word,  $\mathbf{u}$ , is combined with  $\mathbf{p}$  to yield the codeword  $\mathbf{c} = [\mathbf{u} \ \mathbf{p}]$  and so that the code rate is  $1/(1+q)$ . RA codes perform reasonably well on the AWGN channel, and they tend to approach the channel capacity as their rate  $R \rightarrow 0$  and their block length  $n \rightarrow \infty$ . Their main limitations are the code rate, which cannot be higher than  $1/2$ , and the performance of short and medium-length RA codes. The following subsections will present a brief overview of the major enhancements to RA codes which permit operation closer to capacity for both high and low rates.

##### B. Irregular Repeat-Accumulate codes

The systematic irregular repeat-accumulate (IRA) codes generalize the systematic RA codes in that the repetition rate may differ across the  $k$  information bits and that a variable number of bits in the repeated word are combined (modulo 2) prior to sending them through the accumulator. Irregular repeat-accumulate [33] codes provide several advantages over RA codes. They allowing both flexibility in the choice of the repetition rate for each information bit so that high rate codes may be designed and capacity is more easily approached.

The Tanner graph for IRA codes is presented in Fig. 5(a) and the encoder structure (to be discussed further later) is depicted in Fig. 5(b). The variable repetition rate is accounted for in the graph by letting  $d_{b,i}$  vary with  $i$ . The accumulator is represented by the right-most part of the graph, where the dashed edge is added to include the possibility of a tail-biting trellis. Also, we see that  $d_{c,j}$  interleaver output bits are added

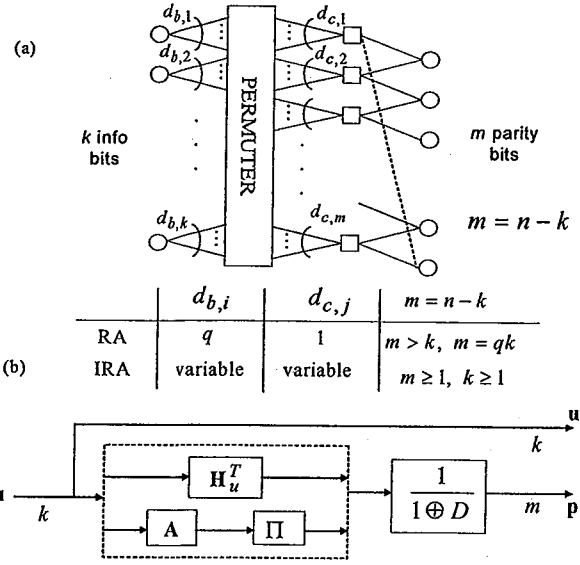


Fig. 5. Tanner graph (a) and encoder (b) for irregular repeat-accumulate codes.

(modulo 2) to produce the  $j$ -th accumulator input. Fig. 5 also includes the representation for RA codes. As indicated in the table in the figure, for an RA code, each information bit node connects to exactly  $q$  check nodes ( $d_{b,i} = q$ ) and each check node connects to exactly one information bit node ( $d_{c,j} = 1$ ). We remark that  $\{d_{b,i}\}$  and  $\{d_{c,j}\}$  can be related to our earlier notation,  $\{d_v(i)\}$  and  $\{d_c(j)\}$ , as follows:  $d_v(i) = d_{b,i}$  for  $i = 1, \dots, k$ ,  $d_v(i) = 2$  for  $i = k + 1, \dots, n$ , and  $d_c(j) = d_{c,j} + 2$  for  $j = 1, \dots, m$ .

To determine the code rate for an IRA code, define  $\bar{q}$  to be the average repetition rate of the information bits

$$\bar{q} = \frac{1}{k} \sum_{i=1}^k d_{b,i},$$

and  $\bar{a}$  as the average of the degrees  $\{d_{c,j}\}$ ,

$$\bar{a} = \frac{1}{m} \sum_{j=1}^m d_{c,j}.$$

Then the code rate for systematic IRA codes is

$$R = \frac{1}{1 + \bar{q}/\bar{a}}.$$

For non-systematic IRA codes,  $R = \bar{a}/\bar{q}$ .

The parity-check matrix for systematic RA and IRA codes has the form

$$\mathbf{H} = [\mathbf{H}_u \ \mathbf{H}_p], \quad (13)$$

where  $\mathbf{H}_p$  is an  $m \times m$  "dual-diagonal" square matrix,





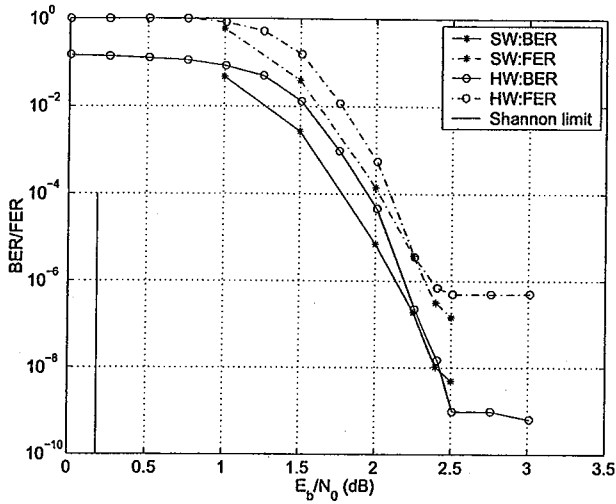


Fig. 6. Performance of a (2044,1024) S-IRA code on the BI-AWGN. HW=hardware simulator. SW=software simulator.

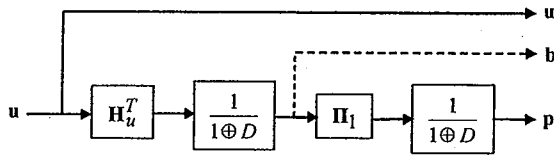


Fig. 7. IRAA encoder.

We now consider irregular repeat-accumulate-accumulate (IRAA) codes which are obtained by concatenating the parity arm of the IRA encoder of Fig. 5(b) with another accumulator, through a permuter, as shown in Fig. 7. (ARAA codes were considered in [49].) The IRAA codeword can be either  $c = [u \ p]$  or  $c = [u \ b \ p]$ , depending on whether the intermediate parity bits  $b$  are punctured or not. The parity-check matrix of the general IRAA code corresponding to Fig. 7 is

$$\mathbf{H}_{\text{IRAA}} = \begin{bmatrix} \mathbf{H}_u & \mathbf{H}_p & 0 \\ 0 & \mathbf{\Pi}_1^T & \mathbf{H}_p \end{bmatrix}, \quad (19)$$

where  $\mathbf{\Pi}_1$  is the interleaver between the two accumulators. When the parity bits  $b$  are not transmitted, they are considered to be punctured, that is, the log-likelihood ratios for these bits are initialized by zeros before decoding. When an IRAA code is structured, we use the notation S-IRAA.

*Example 2:* We compare the performance of rate-1/2 (2048, 1024) S-IRA and S-IRAA codes in this example. For the S-IRA code,  $d_{b,i} = 5$  for all  $i$  and for the S-IRAA code,  $d_{b,i} = 3$  for all  $i$ , and the intermediate parity vector  $b$  is not transmitted to maintain the code rate at 1/2. The protographs for these codes are given in Fig. 8. Because decoder complexity is proportional to the number of edges in a code's parity-check matrix, the complexity of the S-IRAA decoder is slightly greater than the complexity of the S-IRA decoder, even though the column weight in  $\mathbf{H}_u$  is 3 for the former versus 5 for the

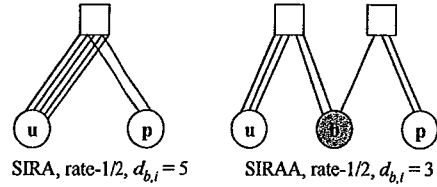


Fig. 8. Rate-1/2 SIRA and SIRAA protographs for the codes in Fig. 9. The shaded node in the SIRAA protograph represents punctured bits. SIRA:  $(E_b/N_0)_{\text{thres}} = 0.97$  dB. SIRAA:  $(E_b/N_0)_{\text{thres}} = 1.1$  dB.

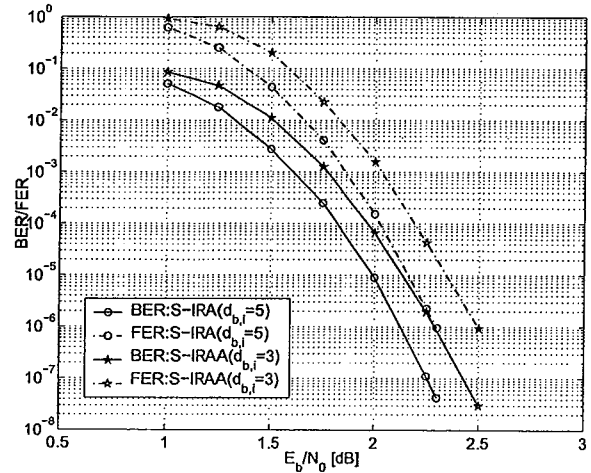


Fig. 9. Performance comparison between rate-1/2 S-IRA and S-IRAA codes on the BI-AWGN,  $n = 2048$  and  $k = 1024$ .

latter. We observe in Fig. 9 that, for both codes, there are no error floors in the BER curves down to  $\text{BER} = 5 \times 10^{-8}$  and in the FER curves down to  $\text{FER} = 10^{-6}$ . While the S-IRAA code is 0.2 dB inferior to the S-IRA code in the waterfall region, we conjecture that it has a lower floor (which is difficult to measure), which would be due to the second accumulator whose function is to increase minimum distance.  $\square$

*Example 3:* This second example is a comparison of rate-1/3 (3072,1024) S-IRA and S-IRAA codes, with  $d_{b,i} = 4$  for the S-IRA code and  $d_{b,i} = 3$  for the S-IRAA code. The protographs for these codes are given in Fig. 10. In this case,  $b$  is part of the transmitted S-IRAA codeword and the decoder complexities are the same. We see in Fig. 11 that, in the low SNR region, the performance of the S-IRA code is 0.4 dB better than the S-IRAA code. However, for high SNRs, the S-IRAA code will outperform the S-IRA code due to its lower error floor.  $\square$

#### D. Generalized IRA codes

Generalized IRA (G-IRA) codes [40][41] increase the flexibility in choosing degree distributions relative to IRA codes, allowing, for example, the design of near-regular efficiently



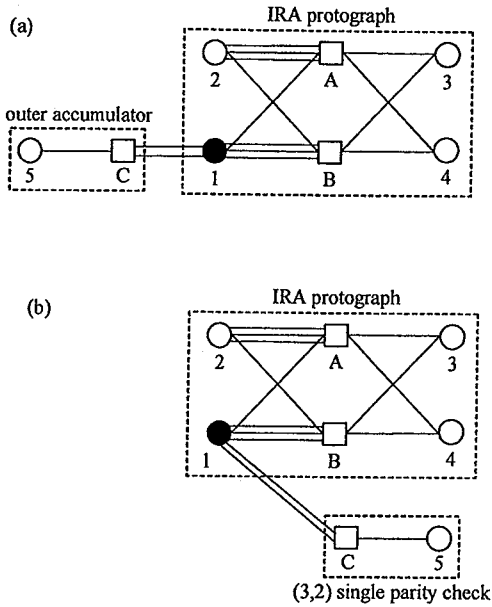


Fig. 13. AR4A protographs in (a) serial-concatenated form and (b) parallel-concatenated form.  $(E_b/N_0)_{thres} = 0.55$  dB.

outer accumulator. This encoding procedure corresponds to a systematic code.

A different code structure is represented by the protograph in Fig. 13(b), which has a parallel-concatenated form. In this case, half (node 2) of the information bits are encoded by the IRA encoder and the other half (node 3) are encoded by both the IRA encoder and a (3, 2) single-parity-check encoder. The node-3 information bits (corresponding to the dark circle in the protograph) are punctured and so codes corresponding to this protograph are non-systematic. While the codes (actually, code ensembles) specified by the protographs in Fig. 13(a) are the same in the sense that the same set of codewords are implied, the  $u \rightarrow c$  mappings are different. The advantage of the non-systematic protograph is that, although the node-3 information bits in Fig. 13(b) are punctured, the node degree is 6, in contrast with the node-1 information bits in Fig. 13(a), in which the node degree is only 1. Given that ARA code decoders converge so slowly, the faster-converging degree-6 node is to be preferred over the slowly converging degree-1 node.

To demonstrate this, we designed a (2048,1024) QC AR4A code whose  $H$  matrix is depicted in Fig. 14. The first group of 512 columns (of weight 6) correspond to variable node type 1 (Fig. 13) whose bits are punctured, and the subsequent four groups of 512 columns correspond, respectively, to node types 2, 3, 4, and 5. The first group of 512 rows correspond to check node type A, and the two subsequent groups of rows correspond to node types B and C, respectively. The performance of the code, with a maximum of  $I_{max} = 50$  iterations is shown in Fig. 15. We note that the (2048,1024) AR4A code reported in [47] achieves  $BER = 10^{-7}$  at  $E_b/N_0 = 2$  dB with

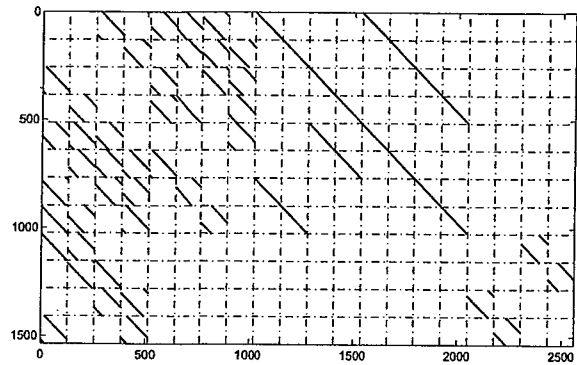


Fig. 14.  $H$  matrix for the (2048,1024) AR4A code.

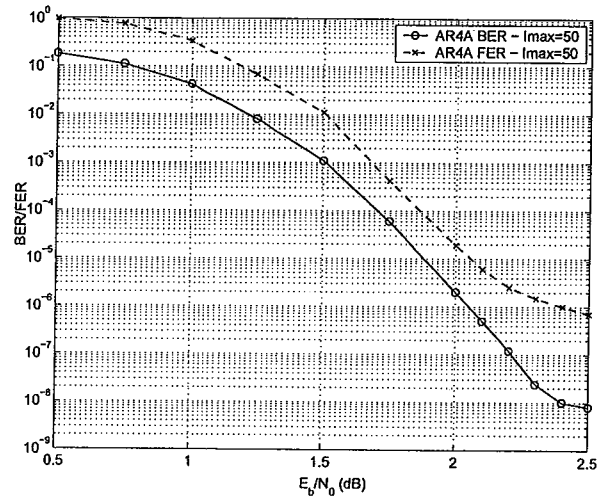


Fig. 15. BER and FER performance for an AR4A code.

200 iterations, whereas in the simulation here,  $BER = 10^{-7}$  is achieved at  $E_b/N_0 = 2.2$  dB with 50 iterations. In Fig. 16, we present the BER performance at  $E_b/N_0 = 2.25$  dB for the five node types that appear in Fig. 13 for  $I_{max}$  ranging from 5 to 20. With 20 iterations, we collected 400 error events, while with fewer iterations, the numbers of collected error events were larger. From the figure, we see that the high-degree variable nodes (node types 2 and 3) converge the fastest. We note also that, while type 3 nodes have degree 6 and type 2 nodes have degree 4, type 3 nodes initially converge slower because the bits corresponding to those nodes are punctured so that the decoder receives no channel LLRs for those bits. However, by 20 iterations, the type 3 bits become more reliable than the type 2 bits.

#### F. Accumulator-Based Codes in Standards

IRA codes and IRA-influenced codes are being considered for several communication standards. The ETSI DVB S2 [48] standard for digital video broadcast specifies two IRA code





and they do not pass through the origin. Since  $\alpha^{q^m-1} = 1$ ,  $\alpha^{n_{EG}} \mathcal{L} = \mathcal{L}$ . These  $n_{EG}$  lines form a cyclic class. The  $(q^{m-1} - 1)(q^m - 1)/(q - 1)$  lines in  $EG(m, q)$  not passing through the origin can be partitioned into  $K = (q^{m-1} - 1)/(q - 1)$  cyclic classes, denoted  $\mathcal{Q}_1, \mathcal{Q}_2, \dots, \mathcal{Q}_K$  where  $\mathcal{Q}_i = \{\mathcal{L}_i, \alpha \mathcal{L}_i, \dots, \alpha^{n_{EG}-1} \mathcal{L}_i\}$  with  $1 \leq i \leq K$ . For each cyclic class  $\mathcal{Q}_i$ , we form an  $n_{EG} \times n_{EG}$  matrix  $\mathbf{H}_{EG,i}$  over  $GF(2)$  with the incidence vectors  $\mathcal{L}_i, \alpha \mathcal{L}_i, \dots, \alpha^{n_{EG}-1} \mathcal{L}_i$  as rows.  $\mathbf{H}_{EG,i}$  is a circulant matrix with column and row weights equal to  $q$ . For  $1 \leq k \leq K$ , let

$$\mathbf{H}_{EG(m,q),k} = \begin{bmatrix} \mathbf{H}_{EG,1} \\ \mathbf{H}_{EG,2} \\ \vdots \\ \mathbf{H}_{EG,k} \end{bmatrix}. \quad (20)$$

Then  $\mathbf{H}_{EG(m,q),k}$  consists of a column of  $k$  circulants of the same size  $n_{EG} \times n_{EG}$ , and it has column and row weights,  $kq$  and  $q$ , respectively. Since no two lines in  $EG(m, q)$  have more than one point in common, it follows that no two rows or two columns in  $\mathbf{H}_{EG(m,q),k}$  have more than a single 1-element in common. We say that  $\mathbf{H}_{EG(m,q),k}$  satisfies the *RC-constraint*. The null space of  $\mathbf{H}_{EG(m,q),k}$  gives a cyclic EG-LDPC code of length  $n_{EG} = q^m - 1$  and minimum distance at least  $kq + 1$  [50][7], whose Tanner graph has a girth of at least 6.

Of particular interest is the two-dimensional Euclidean geometry,  $EG(2, q)$ , which is also called an affine plane over  $GF(q)$  [52]. This geometry has  $q^2$  points and  $q(q + 1)$  lines, and  $q^2 - 1$  of them do not pass through the origin. Each line has  $q$  points and each point lies on  $q + 1$  lines. Each nonorigin point lies on  $q$  lines that do not pass through the origin. If  $\mathcal{L}$  is a line in  $EG(2, q)$  not passing through the origin, then  $\mathcal{L}, \alpha \mathcal{L}, \dots, \alpha^{q^2-2} \mathcal{L}$ , where  $\alpha$  is a primitive element in  $GF(q^2)$ , are all the lines in the geometry not passing through the origin. Hence, all the lines in  $EG(2, q)$  not passing through the origin form a single cyclic class  $\mathcal{Q}$  (i.e.,  $K = 1$ ). Let  $\mathbf{H}_{EG(2,q)}$  denote the  $(q^2 - 1) \times (q^2 - 1)$  circulant formed by the incidence vectors of lines in  $\mathcal{Q}$ . It is a  $(q^2 - 1) \times (q^2 - 1)$  matrix over  $GF(2)$  with both column and row weights equal to  $q$ . The null space of  $\mathbf{H}_{EG(2,q)}$  gives a cyclic EG-LDPC code of length  $q^2 - 1$  and minimum distance at least  $q + 1$ . For  $q = 2^s$ , the parameters of the code with parity-check matrix  $\mathbf{H}_{EG(2,q)}$  are as follows [7]:

Length	$n = 2^{2s} - 1,$
Number of parity bits	$n - k = 3^s - 1,$
Dimension	$k = 2^{2s} - 3^s,$
Minimum distance	$d_{\min} \geq 2^s + 1,$
Size of the LDPC matrix	$(2^{2s} - 1) \times (2^{2s} - 1),$
Row weight	$2^s,$
Column weight	$2^s.$

Generators polynomials for these codes can be readily obtained from [7].

*Example 4:* The cyclic LDPC code constructed based on the two-dimensional Euclidean geometry  $EG(2, 2^6)$  over  $GF(2^6)$  is a (4095, 3367) LDPC code with rate 0.822 and

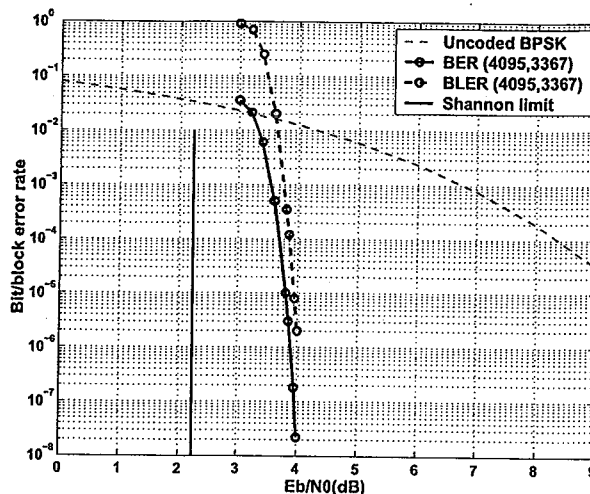


Fig. 17. Performance of the binary (4095,3367) cyclic EG-LDPC code given in Example 4 over the BI-AWGN.

minimum distance 65. The performance of this code with iterative decoding using the SPA is shown in Fig. 17. At a BER of  $10^{-6}$ , it performs 1.65 dB from the Shannon limit. Since it has a very large minimum distance, it has a very low error-floor.  $\square$

### B. Cyclic Projective Geometry LDPC Codes

The  $m$ -dimensional projective geometry over  $GF(q)$ , denoted by  $PG(m, q)$ , consists of  $n_{PG} = (q^{m+1} - 1)/(q - 1)$  points. Each point is represented by a non-zero  $(m + 1)$ -tuple  $\mathbf{a}$  over  $GF(q)$  such that all  $q - 1$  non-zero multiples  $\beta \mathbf{a}$ , where  $\beta$  is a non-zero element in  $GF(q)$ , represent the same point. A line in  $PG(m, q)$  consists of all points of the form  $\beta_1 \mathbf{a}_1 + \beta_2 \mathbf{a}_2$ , where  $\mathbf{a}_1$  and  $\mathbf{a}_2$  are two  $(m + 1)$ -tuples that are linearly independent over  $GF(q)$  and  $\beta_1$  and  $\beta_2$  are elements in  $GF(q)$ , with  $\beta_1$  and  $\beta_2$  not simultaneously equal to zero. There are  $(q^{m+1} - 1)(q^m - 1)/(q^2 - 1)(q - 1)$  lines in  $PG(m, q)$  and each line consists of  $q + 1$  points. Two points are connected by one and only one line and each point lies on  $(q^m - 1)/(q - 1)$  lines.

The extension field  $GF(q^{m+1})$  of  $GF(q)$  is a realization of  $PG(m, q)$  [7]. Let  $\alpha$  be a primitive element of  $GF(q^{m+1})$ . A point in  $PG(m, q)$  is represented by a non-zero element  $\alpha^i$ . Every nonzero element in the base field  $GF(q)$  can be written as  $\alpha^l$  for some  $l$  which is divisible by  $(q^{m+1} - 1)/(q - 1)$ . Hence, the elements  $\alpha^i$  and  $\alpha^j$  represent the same point in  $PG(m, q)$  if and only if  $i \equiv j \pmod{(q^{m+1} - 1)/(q - 1)}$ . Therefore, we can take the elements  $1, \alpha, \dots, \alpha^{n_{PG}-1}$  to represent all the points in  $PG(m, q)$ .

Let  $\mathcal{L}$  be a line in  $PG(m, q)$ . Define the  $n_{PG}$ -tuple over  $GF(2)$   $\mathbf{v}_{\mathcal{L}} = (v_0, v_1, \dots, v_{n_{PG}-1})$  whose components correspond to the  $n_{PG} = (q^{m+1} - 1)/(q - 1)$  points of  $PG(m, q)$ , where  $v_i = 1$  if the point represented by  $\alpha^i$  lies on  $\mathcal{L}$ , otherwise  $v_i = 0$ . The vector  $\mathbf{v}_{\mathcal{L}}$  is called the incidence

vector of  $\mathcal{L}$ . Clearly,  $\alpha\mathcal{L}$  is also a line in the geometry whose incidence vector  $\mathbf{v}_{\alpha\mathcal{L}}$  is the cyclic-shift of  $\mathbf{v}_{\mathcal{L}}$ .

For even  $m$ , the lines in  $\text{PG}(m, q)$  can be partitioned into  $K_1 = (q^m - 1)/(q^2 - 1)$  cyclic classes  $\mathcal{Q}_1, \mathcal{Q}_2, \dots, \mathcal{Q}_{K_1}$ , each class consisting of  $n_{\text{PG}}$  lines. For each cyclic class  $\mathcal{Q}_i$ , we can form an  $n_{\text{PG}} \times n_{\text{PG}}$  circulant  $\mathbf{H}_{\text{PG},i}$  with both column and row weights equal to  $q + 1$ . For  $1 \leq k \leq K_1$ , form the following matrix:

$$\mathbf{H}_{\text{PG}(m,q),k}^{(1)} = \begin{bmatrix} \mathbf{H}_{\text{PG},1} \\ \mathbf{H}_{\text{PG},2} \\ \vdots \\ \mathbf{H}_{\text{PG},k} \end{bmatrix}, \quad (21)$$

which has column and row weights  $k(q + 1)$  and  $q + 1$ , respectively. The null space of  $\mathbf{H}_{\text{PG}(m,q),k}^{(1)}$  gives a cyclic PG-LDPC code of length  $n_{\text{PG}} = (q^{m+1} - 1)/(q - 1)$  and minimum distance at least  $k(q + 1) + 1$  whose Tanner graph has a girth of at least 6. For odd  $m$ , the lines in  $\text{PG}(m, q)$  can be partitioned into  $K_2 + 1$  cyclic classes,  $\mathcal{Q}_0, \mathcal{Q}_1, \mathcal{Q}_2, \dots, \mathcal{Q}_{K_2}$ , where  $K_2 = q(q^{m-1} - 1)/(q^2 - 1)$ . Except for  $\mathcal{Q}_0$ , each cyclic class consists of  $n_{\text{PG}}$  lines. The cyclic class  $\mathcal{Q}_0$  consists of only  $\lambda = (q^{m+1} - 1)/(q^2 - 1)$  lines. For each cyclic class  $\mathcal{Q}_i$  with  $i \neq 0$ , we can form a  $n_{\text{PG}} \times n_{\text{PG}}$  circulant  $\mathbf{H}_{\text{PG},i}$  with the incidence vectors of the lines in  $\mathcal{Q}_i$  as rows. For  $1 \leq k \leq K_2$ , we can form a matrix  $\mathbf{H}_{\text{PG}(m,q),k}^{(2)}$  of the form given by (21). The null space of  $\mathbf{H}_{\text{PG}(m,q),k}^{(2)}$  gives a cyclic PG-LDPC code of length  $n_{\text{PG}}$  and minimum distance at least  $k(q + 1) + 1$  whose Tanner graph has a girth of at least 6.

As in the case of Euclidean geometries, the two-dimensional projective geometry,  $\text{PG}(2, q)$ , which is also called a projective plane over  $\text{GF}(q)$  [52], is of particular interest. This geometry has  $q^2 + q + 1$  points and  $q^2 + q + 1$  lines. Each line has  $q + 1$  points and each point lies on  $q + 1$  lines. If  $\mathcal{L}$  is a line in  $\text{PG}(2, q)$ , then  $\mathcal{L}, \alpha\mathcal{L}, \dots, \alpha^{q^2+q}\mathcal{L}$ , where  $\alpha$  is a primitive element in  $\text{GF}(q^2)$ , are all the lines in the geometry. Hence, all the lines in  $\text{PG}(2, q)$  form a single cyclic class  $\mathcal{Q}$  (i.e.,  $K_1 = 1$ ). Let  $\mathbf{H}_{\text{PG}(2,q)}$  denote the  $n_{\text{PG}} \times n_{\text{PG}}$  circulant formed by the incidence vectors of the lines in  $\mathcal{Q}$ . It is a  $(q^2 + q + 1) \times (q^2 + q + 1)$  matrix over  $\text{GF}(2)$  with both column and row weights equal to  $q + 1$ . The null space of  $\mathbf{H}_{\text{PG}(2,q)}$  gives a cyclic PG-LDPC code of length  $q^2 + q + 1$  and minimum distance at least  $q + 2$ . For  $q = 2^s$ , the parameters of the cyclic PG-LDPC code given by the null space of  $\mathbf{H}_{\text{PG}(2,q)}$  are as follows [7]:

Length	$n = 2^{2s} + 2^s + 1,$
Number of parity bits	$n - k = 3^s + 1,$
Dimension	$k = 2^{2s} + 2^s - 3^s,$
Minimum distance	$d_{\min} \geq 2^s + 2,$
Size of the LDPC matrix	$(2^{2s} + 2^s + 1) \times (2^{2s} + 2^s + 1)$
Row weight	$2^s + 1,$
Column weight	$2^s + 1.$

Generators polynomials for these codes can also be readily obtained from [7].

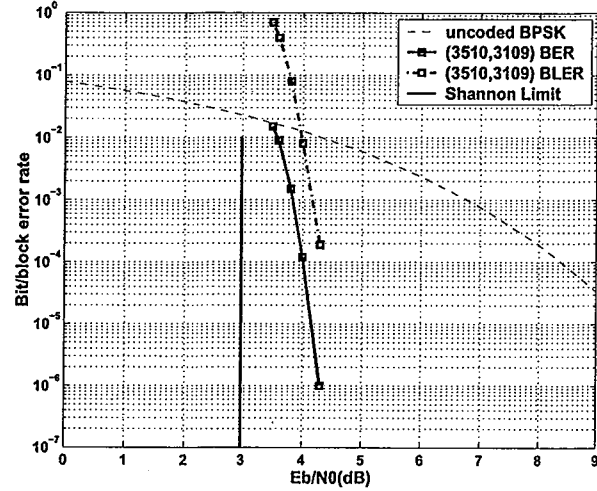


Fig. 18. Performance of the binary (3510,3109) quasi-cyclic PG-LDPC code given in Example 5 over the BI-AWGNC.

### C. Quasi-Cyclic Finite Geometry LDPC Codes

Let  $\mathbf{R}_{\text{EG}(m,q),k}$  be the transpose of the parity-check matrix  $\mathbf{H}_{\text{EG}(m,q),k}$  of a cyclic EG-LDPC code given by (20), i.e.,

$$\mathbf{R}_{\text{EG}(m,q),k} \triangleq \mathbf{H}_{\text{EG}(m,q),k}^T = [\mathbf{H}_1^T \mathbf{H}_2^T \cdots \mathbf{H}_k^T], \quad (22)$$

which consists of a row of  $k$  circulants of size  $n_{\text{EG}} \times n_{\text{EG}}$ . It is a  $(q^m - 1) \times k(q^m - 1)$  matrix with column and row weights  $q$  and  $kq$ , respectively. The null space of  $\mathbf{R}_{\text{EG}(m,q),k}$  gives a quasi-cyclic EG-LDPC code of length  $k(q^m - 1)$  and minimum distance at least  $q + 1$  whose Tanner graph has a girth of at least 6.

Similarly, let  $\mathbf{R}_{\text{PG}(m,q),k}^{(e)}$  be the transpose of  $\mathbf{H}_{\text{PG}(m,q),k}^{(e)}$  with  $e = 1$  or  $2$ . Then the null space of  $\mathbf{H}_{\text{PG}(m,q),k}^{(e)}$  gives a quasi-cyclic PG-LDPC code of length  $k(q^{m+1} - 1)/(q - 1)$  and minimum distance at least  $q + 2$ .

*Example 5:* Consider the 3-dimensional projective geometries  $\text{PG}(3, 2^3)$  over  $\text{GF}(2^3)$ . This geometry consists of 585 points and 4745 lines, each line consists of 9 points. The lines in this geometry can be partitioned into 9 cyclic classes,  $\mathcal{Q}_0, \mathcal{Q}_1, \dots, \mathcal{Q}_8$ , where  $\mathcal{Q}_0$  consists of 65 lines and each of the other 8 cyclic classes consists of 585 lines. For each  $\mathcal{Q}_i$  with  $1 \leq i \leq 8$ , we can form a  $585 \times 585$  circulant  $\mathbf{H}_{\text{PG},i}$  over  $\text{GF}(2)$  with the incidence vectors in  $\mathcal{Q}_i$  as the rows. Set  $k = 6$ . Form the following  $585 \times 3510$  matrix:  $\mathbf{R}_{\text{PG}(3,2^3),6}^{(2)} = [\mathbf{H}_{\text{PG},1}^T \mathbf{H}_{\text{PG},2}^T \cdots \mathbf{H}_{\text{PG},6}^T]$ , which has column and row weights 9 and 54, respectively. The null space of this matrix gives a (3510, 3109) quasi-cyclic PG-LDPC code with rate 0.8858 and minimum distance at least 10. The performance of this code decoded with iterative decoding using the SPA is shown in Fig. 18. At a BER of  $10^{-6}$ , it performs 1.3 dB from the Shannon limit.  $\square$

Other LDPC codes constructed based on finite geometries can be found in [53][54][55][56][57]. Finite geometry LDPC codes can also be effectively decoded with one-step majority-



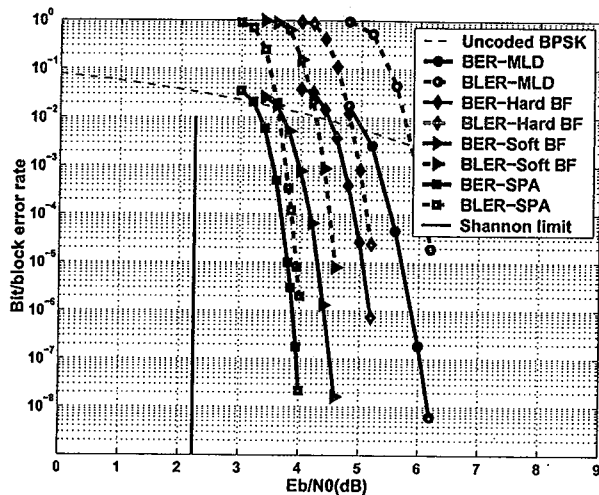


Fig. 19. Performance of the binary (4095,3367) EG-LDPC code given in Example 4 with various decoding techniques over the BI-AWGN. MLD = majority-logic decoding. BF = bit-flipping. SPA = sum-product algorithm.

logic decoding [7]; hard-decision bit-flipping (BF) decoding [1][50][7] and weighted BF decoding [50][58][59][60]. These decoding methods together with the soft-input and soft-output (SISO) iterative decoding based on belief propagation offer various trade-offs between performance and decoding complexity. The one-step majority-logic decoding requires the least decoding complexity while the (SISO) iterative decoding based on belief propagation requires the most decoding complexity and the other two decoding methods are in between. Fig. 19 shows the performances of the (4095,3367) cyclic EG-LDPC code given in Example 4 with various decoding methods.

## VI. REGULAR RS-BASED LDPC CODES

This section first gives a brief survey of a class of structured LDPC codes that are constructed from the codewords of Reed-Solomon (RS) codes with two information symbols. Then two new classes of Reed-Solomon-based quasi-cyclic LDPC codes are presented. Experimental results show that constructed codes perform very well over the AWGN channel with iterative decoding.

In [61], a class of structured regular LDPC codes was presented which were constructed from the codewords of RS codes with two information symbols. These codes are referred to as RS-based LDPC codes and their parity-check matrices are arrays of permutation matrices. RS-based LDPC codes perform well with iterative decoding over the AWGN channel. Most importantly, they have low error-floors and their decoding converges very fast. These features are important in high-speed communication systems where very low error rates are required, such as the 10G Base-T Ethernet. In this section, we first give a more general form of the RS-based LDPC codes presented in [61] and then we present two classes of RS-based QC LDPC codes.

Let  $\alpha$  be a primitive element of the finite field  $GF(q)$ . Then the following powers of  $\alpha$ ,  $\alpha^{-\infty} \triangleq 0, \alpha^0 = 1, \alpha, \dots, \alpha^{q-2}$ , form the  $q$  elements of  $GF(q)$  and  $\alpha^{q-1} = 1$ . For  $i = -\infty, 0, 1, \dots, q-2$ , represent each element  $\alpha^i$  of  $GF(q)$  by a  $q$ -tuple over  $GF(2)$ ,

$$\mathbf{z}(\alpha^i) = (z_{-\infty}, z_0, z_1, z_2, \dots, z_{q-2}), \quad (23)$$

with components corresponding to the  $q$  elements,  $\alpha^{-\infty}, \alpha^0, \dots, \alpha^{q-2}$ , of  $GF(q)$ , where the  $i$ -th component  $z_i = 1$  and all the other components equal to zero. This binary  $q$ -tuple  $\mathbf{z}(\alpha^i)$  is a unit-vector with one and only one 1-component and is called the *location vector* of  $\alpha^i$ . It is clear that the location vectors of two different elements in  $GF(q)$  have their 1-components at two different locations. Suppose we form a  $q \times q$  matrix  $\mathbf{A}$  over  $GF(2)$  with the location vectors of the  $q$  elements of  $GF(q)$  as rows arranged in any order. Then  $\mathbf{A}$  is a  $q \times q$  permutation matrix.

Consider an extended  $(q, 2, q-1)$  RS code  $C_b$  over  $GF(q)$  [7] of length  $q$  with two information symbols and minimum distance  $q-1$ . The nonzero codewords of  $C_b$  have two different weights,  $q-1$  and  $q$ . Because the minimum distance of  $C_b$  is  $q-1$ , two codewords in  $C_b$  differ in at least  $q-1$  places, i.e., they have at most one place where they have the same code symbols. Let  $\mathbf{v}$  be a nonzero codeword in  $C_b$  with weight  $q$ . Then, the set  $C_b^{(0)} = \{c\mathbf{v} : c \in GF(q)\}$  of  $q$  codewords in  $C_b$  of weight  $q$  forms a one-dimensional subcode of  $C_b$  with minimum distance  $q$  and is a  $(q, 1, q)$  extended RS code over  $GF(q)$ . Any two codewords in  $C_b^{(0)}$  differ at every location. Partition  $C_b$  into  $q$  cosets,  $C_b^{(0)}, C_b^{(1)}, \dots, C_b^{(q-1)}$ , based on the subcode  $C_b^{(0)}$ . Then two codewords in any coset  $C_b^{(i)}$  differ at every location and two codewords from two different cosets  $C_b^{(i)}$  and  $C_b^{(j)}$  with  $i \neq j$  differ in at least  $q-1$  locations. For  $0 \leq i < q$ , form a  $q \times q$  matrix  $\mathbf{G}_i$  over  $GF(q)$  with the codewords in  $C_b^{(i)}$  as rows. Then all the  $q$  entries in a column of  $\mathbf{G}_i$  are different and they form all the  $q$  elements of  $GF(q)$ . It follows from the structural properties of the cosets of  $C_b^{(0)}$  that any two rows from any matrix  $\mathbf{G}_i$  differ at every position and any two rows from two different matrices  $\mathbf{G}_i$  and  $\mathbf{G}_j$  with  $i \neq j$  can have at most one location where they have identical symbols.

For  $0 \leq i < q$ , replacing each entry in  $\mathbf{G}_i$  by its location vector, we obtain a  $q \times q^2$  matrix  $\mathbf{B}_i$  over  $GF(2)$  which consists of a row of  $q$  permutation matrices of size  $q \times q$ ,

$$\mathbf{B}_i = [\mathbf{A}_{i,0} \ \mathbf{A}_{i,1} \ \dots \ \mathbf{A}_{i,q}], \quad (24)$$

where  $\mathbf{A}_{i,j}$  has the location vectors of the  $q$  entries of the  $j$ -th column of  $\mathbf{G}_i$  as rows. Next, we form the following  $q \times q$  array of  $q \times q$  permutation matrices with  $\mathbf{B}_0, \mathbf{B}_1, \dots, \mathbf{B}_{q-1}$

as submatrices arranged in a column:

$$\mathbf{H}_{r,s,1} = \begin{bmatrix} \mathbf{B}_0 \\ \mathbf{B}_1 \\ \vdots \\ \mathbf{B}_{q-1} \end{bmatrix} \quad (25)$$

$$= \begin{bmatrix} \mathbf{A}_{0,0} & \mathbf{A}_{0,1} & \cdots & \mathbf{A}_{0,q-1} \\ \mathbf{A}_{1,0} & \mathbf{A}_{1,1} & \cdots & \mathbf{A}_{1,q-1} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}_{q-1,0} & \mathbf{A}_{q-1,1} & \cdots & \mathbf{A}_{q-1,q-1} \end{bmatrix}$$

$\mathbf{H}_{r,s,1}$  is a  $q^2 \times q^2$  matrix over  $\text{GF}(2)$  with both column and row weights  $q$ . For  $q > 7$ , each permutation matrix  $\mathbf{A}_{i,j}$  is a sparse matrix and hence  $\mathbf{H}_{r,s,1}$  is also a sparse matrix. It follows from the structural properties of the matrices  $\mathbf{G}_i$ 's that no two rows (or two columns) of  $\mathbf{H}_{r,s,1}$  can have more than one 1-component in common. This implies that there are no four 1-components at the four corners of a rectangle in  $\mathbf{H}_{r,s,1}$ , that is,  $\mathbf{H}_{r,s,1}$  satisfies the RC-constraint and, hence, has a girth of at least 6 [50][7].

For any pair of integers,  $(d_v, d_c)$ , with  $1 \leq d_v, d_c \leq q$ , let  $\mathbf{H}_{r,s,1}(d_v, d_c)$  be a  $d_v \times d_c$  subarray of  $\mathbf{H}_{r,s,1}$ . Then  $\mathbf{H}_{r,s,1}(d_v, d_c)$  is a  $d_v q \times d_c q$  matrix over  $\text{GF}(2)$  with column and row weights  $d_v$  and  $d_c$ , respectively. It is a  $(d_v, d_c)$ -regular matrix which also satisfies the RC-constraint. The null space of  $\mathbf{H}_{r,s,1}(d_v, d_c)$  gives a  $(d_v, d_c)$ -regular RS-based LDPC code  $C_{r,s,1}$  of length  $d_c q$  with rate at least  $(d_c - d_v)/d_c$  and minimum distance at least  $d_v + 1$  [50], [7], whose Tanner graph has a girth of at least 6. Since  $\mathbf{H}_{r,s,1}$  consists of an array of permutation matrices, no odd number of columns of  $\mathbf{H}_{r,s,1}$  can be added to zero. This implies that the RS-based regular LDPC code  $C_{r,s,1}$  has only even-weight codewords. Consequently, its minimum distance is even, at least  $d_v + 2$  for even  $d_v$  and  $d_v + 1$  for odd  $d_v$ . The above construction gives a class of regular LDPC codes whose Tanner graphs have girth at least 6. For each  $(q, 2, q-1)$  extended RS code  $C_b$  over  $\text{GF}(q)$ , we can construct a family of regular RS-based LDPC codes with various lengths, rates and minimum distances.  $C_b$  is referred to as the base code.

*Example 6:* Consider the  $(64, 2, 63)$  extended RS code  $C_b$  over  $\text{GF}(2^6)$ . Based on the codewords of this RS code  $C_b$ , we can construct a  $64 \times 64$  array  $\mathbf{H}_{r,s,1}$  of  $64 \times 64$  permutation matrices. Suppose we choose  $d_v = 6$  and  $d_c = 32$ . Take a  $6 \times 32$  subarray  $\mathbf{H}_{r,s,1}(6, 32)$  from  $\mathbf{H}_{r,s,1}$ , say the  $6 \times 32$  subarray at the upper left corner of  $\mathbf{H}_{r,s,1}$ .  $\mathbf{H}_{r,s,1}(6, 32)$  is a  $384 \times 2048$  matrix over  $\text{GF}(2)$  with column and row weights 6 and 32, respectively. The null space of this matrix gives a  $(2048, 1723)$  regular RS-based LDPC code with rate 0.841 and minimum distance at least 8. Assume transmission over the AWGN channel with BPSK signaling. The performance of this code with iterative decoding using the SPA (50 iterations) is shown in Fig. 20. At a BER of  $10^{-6}$ , the code performs 1.55 dB from the Shannon limit. The standard code for the IEEE 802.2 10G Base-T Ethernet is a  $(2048, 1723)$  regular RS-based LDPC code given by the null space of a  $6 \times 32$  subarray of the array  $\mathbf{H}_{r,s,1}$  constructed above.  $\square$

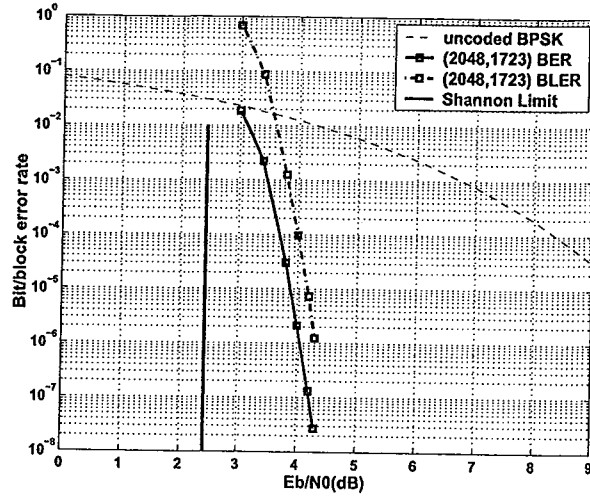


Fig. 20. Performance of the binary  $(2048,1723)$  regular RS-based LDPC code given in Example 6 over the BI-AWGNC.

#### A. Class-I RS-Based QC-LDPC Codes

RS codes were originally defined in polynomial form in frequency domain [63]. Using the polynomial form, arrays of circulant permutation matrices that satisfy the RC-constraint can be constructed from all the codewords of an RS code over a prime field  $\text{GF}(p)$  with two information symbols. Based on these arrays of circulant permutation matrices, a class of QC-LDPC codes can be constructed.

Let  $p$  be a prime. Consider the prime field  $\text{GF}(p) = \{0, 1, \dots, p-1\}$  under modulo- $p$  addition and multiplication. Let  $\mathcal{P} = \{\mathbf{a}(X) = a_1 X + a_0 : a_1, a_0 \in \text{GF}(p)\}$  be the set of  $p^2$  polynomials of degree one or less with coefficients from  $\text{GF}(p)$ . For each polynomial  $\mathbf{a}(X)$  in  $\mathcal{P}$ , define the following  $p$ -tuple over  $\text{GF}(p)$ :  $\mathbf{v} = (\mathbf{a}(0), \mathbf{a}(1), \dots, \mathbf{a}(p-1))$ , where  $\mathbf{a}(j) = a_1 \cdot j + a_0$  with  $j \in \text{GF}(p)$ . Then the set of  $p^2$   $p$ -tuples,

$$C_b = \{\mathbf{v} = (\mathbf{a}(0), \mathbf{a}(1), \dots, \mathbf{a}(p-1)) : \mathbf{a}(X) \in \mathcal{P}\}, \quad (26)$$

gives a  $(p, 2, p-1)$  RS code over  $\text{GF}(p)$  with two information symbols. The RS code  $C_b$  given by (26) is not cyclic.

Consider the subset  $\mathcal{P}_0 = \{\mathbf{a}(X) = a_0 : a_0 \in \text{GF}(p)\}$  of zero-degree polynomials in  $\mathcal{P}$ . Then the set of  $p$ -tuples,

$$C_b^{(0)} = \{(\mathbf{a}(0), \mathbf{a}(1), \dots, \mathbf{a}(p-1)) : \mathbf{a}(X) \in \mathcal{P}_0\} = \{(a_0, a_0, \dots, a_0) : a_0 \in \text{GF}(p)\} \quad (27)$$

constructed from the zero-degree polynomials in  $\mathcal{P}_0$  forms a one-dimensional subcode of  $C_b$  and is a  $(p, 1, p-1)$  RS code over  $\text{GF}(p)$  with minimum distance  $p$ . Partition  $C_b$  with respect to  $C_b^{(0)}$  into  $p$  cosets,  $C_b^{(0)}, C_b^{(1)}, \dots, C_b^{(p-1)}$ , where

$$C_b^{(i)} = \{(\mathbf{a}(0), \dots, \mathbf{a}(p-1)) : \mathbf{a}(X) = iX + a_0, a_0 \in \text{GF}(p)\}. \quad (28)$$

For  $0 \leq i < p$ ,  $C_b^{(i)}$  contains  $p$  codewords in  $C_b$  of the following form:

$$(i \cdot 0 + a_0, i \cdot 1 + a_0, \dots, i \cdot (p-1) + a_0). \quad (29)$$