White Rose Research Online

eprints@whiterose.ac.uk
https://eprints.whiterose.ac.uk
Universities of Leeds, Sheffield and York

Deposited via The University of Sheffield.

White Rose Research Online URL for this paper:
https://eprints.whiterose.ac.uk/id/eprint/185957/

Version: Published Version

Article:
Sholiyi, A.O. and O'Farrell, T. (2022) Irregular vector turbo codes with low complexity. EURASIP Journal on Wireless Communications and Networking, 2022. 32. ISSN: 1687-1472

https://doi.org/10.1186/s13638-022-02115-5

Reuse
This article is distributed under the terms of the Creative Commons Attribution (CC BY) licence. This licence allows you to distribute, remix, tweak, and build upon the work, even commercially, as long as you credit the authors for the original work. More information and the full terms of the licence here: https://creativecommons.org/licenses/

Takedown
If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.

UNIVERSITY OF LEEDS
University of Sheffield
UNIVERSITY of York

EURASIP Journal on Wireless Communications and Networking

RESEARCH Open Access

Irregular Vector Turbo Codes with low complexity

Abiodun O. Sholiyi[1*] and Timothy O'Farrell[2]

*Correspondence:
eiholiness@gmail.com
[1] Swansea University, Swansea, West Glamorgan, UK
Full list of author information is available at the end of the article

Abstract

The term Block Turbo Code typically refers to the iterative decoding of a serially concatenated two-dimensional systematic block code. This paper introduces a Vector Turbo Code that is irregular but with code rates comparable to those of a Block Turbo Code (BTC) when the Bahl Cocke Jelinek Raviv algorithm is used. In BTC's, the horizontal (or vertical) blocks are encoded first and the vertical (or horizontal) blocks second. The irregular Vector Turbo Code (iVTC) uses information bits that participate in varying numbers of trellis sections, which are organized into blocks that are encoded horizontally (or vertical) without vertical (or horizontal) encoding. The decoding requires only one soft-input soft-output decoder. In general, a reduction in complexity, in comparison to a BTC was achieved for the same very low probability of bit error (10^{-5}). Performance in the AWGN channel shows that iVTC is capable of achieving a significant coding gain of 1.28dB for a 64QAM modulation scheme, at a bit error rate of 10^{-5}over its corresponding BTC. Simulation results also show that some of these codes perform within 0.49dB of capacity for binary transmission over an AWGN channel.

Keywords: BCJR algorithm, Block turbo code, Extrinsic information transfer function, Log-MAP, Irregular coding

1 Introduction

The introduction of iterative decoding (i.e., Turbo Codes) by Berrou et al. [8] has significantly reduced the transmit power required to achieve negligible bit error rate (BER) with moderate to large complexity in digital communication systems. Originally designed for low code rates, i.e., code rates 1/3 and 1/2, the Turbo Code (TC) achieved a very low BER, e.g., 10^{-5} with average bit energy to noise energy ratio (E_b/N_o) close to Shannon's theoretical limit in an AWGN channel. For the decoding of the constituent codes, Berrou used a modified version of the Bahl Cocke Jelinek Raviv (BCJR) algorithm [17] known as the maximum a posteriori (MAP) algorithm. Due to its moderate complexity and high coding gain, the Turbo Code has been widely adopted as the channel codec in modern wireless systems [14]. Iterative decoding of product codes, also known as Turbo Product Codes (TPC) or Block Turbo Codes (BTC), using a soft-input soft-output decoder was later introduced by Pyndiah [7]. The BTC also achieved a very low BER (10^{-5}) with an E_b/N_o close to Shannon's theoretical limit in an AWGN channel. Pyndiah introduced a new decoding scheme, "the

Springer Open

© The Author(s) 2022. **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

Chase-Pyndiah" soft decoder, where a selected list of code words is used to produce soft-outputs from the Chase decoder, since Chase is a hard decision decoder. This is implemented by searching for the least reliable bits in a code word and then generating test vectors by flipping some of these least reliable bits. The complexity of this decoder is a function of the number of bits flipped. Of importance in this channel codec is its high code rate, making it very useful for high data rate communication systems. In the channel codecs described above, two soft-input soft-output (SISO) decoding components are used, which leads to a very low BER performance.

The unequal protection of information bits has recently been of interest, from the design of an irregular Low Density Parity Check (LDPC) code [6] to the design of an irregular Turbo Code [2]. Significant performance benefits stem from the extra protection on some of the transmitted bits. An irregular Turbo Code was first proposed by Brendan and McKay in [3], where a coding gain of 0.23dB at a BER of 10^{-4} was demonstrated over the corresponding regular Turbo Code in an AWGN channel using BPSK modulation. In [2], Sawaya improved the system BER performance of the scheme proposed by Brendan by using an alternative irregular bit pattern to obtain a coding gain of 0.24dB at 10^{-6} BER compared to the regular Turbo Code. The irregular Turbo Codes described in [2] and [3] utilized a single recursive systematic convolutional (RSC) encoder and a single SISO decoder. This reduced the complexity of the decoder compared to the regular Turbo Code but still required very large frame sizes and high numbers of iterations (i.e., about 100).

Recent publications have focused on irregular turbo codes such as in [4, 12] to achieve a very low BER. In this paper, a new irregular iterative coding scheme, termed an irregular Vector Turbo Code (iVTC) to distinguish it from a BTC, is presented. Unlike the BTC, which uses both row and column encoding, the iVTC uses only horizontal encoding without vertical encoding (or vice versa). Specifically, the new iVTC uses:

(1) Binary BCH codes as inner block codes.
(2) Row encoding only.
(3) A single SISO decoder.
(4) A single extrinsic information computational block.

Our results show that the iVTC exhibits greater design flexibility, enhanced system BER performance and lower implementation complexity than the best known BTCs. For a 64QAM modulation scheme in AWGN, the iVTC achieves a significant coding gain of 1.28dB at 10^{-5} BER over its corresponding BTC and is closer to the Shannon theoretical limit as determined by the Extrinsic Information Transfer (EXIT) function of the iVTC.

Notation: The following notation is used throughout this paper: let $\mathbf{V}$ denote a matrix, $\mathbf{v}$ a vector of length$|\nu|$ and v a scalar of magnitude$|\nu|$.

The paper is structured as follows. Section 3 describes the iVTC encoder, whereas Sect. 4 describes the corresponding decoder, including a detailed explanation of the extrinsic information exchange between the SISO decoding block and the extrinsic computation block. Simulation results of the system BER performance are presented and discussed in Sect. 5 together with an analysis of the decoding complexity. Finally, the conclusions are given in Sect. 6.

2 Methods/experimental

The experiments and simulations in this paper were done using MATLAB with an AWGN channel. In this study, a single decode iteration of an iVTC consists of only one Log-MAP calculation, whereas in a TPC a single decode iteration consists of two Log-MAP calculations.

Information regarding the designs used can be found in Sects. 3, 4 and 5 along with comparisons made between the TPC and the iVTC.

3 The irregular Vector Turbo Code (iVTC) encoder

In a BTC a block or matrix $\mathbf{K} \in \mathfrak{B}^{k \times k}$ of information bits is encoded horizontally first and vertically second (or vice versa) in a block format [7], where $\mathfrak{B}$ denotes a binary Galois field over $\mathfrak{F}_2$. By reading horizontally into the first encoder and vertically into the second encoder, a block interleaver is realized. In contrast, in an iVTC the bits of an information vector $\mathbf{k} \in \mathfrak{B}^{a\times 1}$ are repeated to generate a vector $\mathbf{g} \in \mathfrak{B}^{k\times 1}$, where $a < k$. Then k such vectors are concatenated to form a longer vector $\mathbf{h} \in \mathfrak{B}^{k^2\times 1}$, which is randomly interleaved to form the vector $\widetilde{\mathbf{h}} \in \mathfrak{B}^{k^2\times 1}$. Vector $\widetilde{\mathbf{h}}$ is partitioned into k shorter vectors $\widetilde{\mathbf{g}} \in \mathfrak{B}^{k\times 1}$ each of which is encoded by a systematic (n, k) linear block code giving k encoded vectors $\widetilde{\mathbf{c}} \in \mathfrak{B}^{n\times 1}$, where $n = k + p$ for p parity bits. The parity bits of each encoded vector $\widetilde{\mathbf{c}}$ are extracted to form a parity vector $\mathbf{p} \in \mathfrak{B}^{p\times 1}$, which is appended to the original information vector $\mathbf{k}$ to form the transmitted codeword $\boldsymbol{\nu} \in \mathfrak{B}^{(a+p)\times 1}$. The transmission of all k codewords corresponding to $\mathbf{k}$ information vectors, equates to the interleaver depth k^2. Figure 1 illustrates the general structure of the iVTC encoder.

When each information bit is repeated once (i.e., uniform repetition of degree 2), this is equivalent to the regular TPC. This is so because, each information bit in the degree 2 iVTC codeword has two extrinsic information values in the decoding process which is similar to the regular TPC with two extrinsic information values per information bit (one from the horizontal decoding and the other from the vertical decoding). It will be shown in Sect. 5 of this paper that the performance of the TPC in [7] and the uniformly repeated iVTC of degree 2 have similar BER versus E_b/N_0 performance for large block sizes in an AWGN channel with slight differences.

A fraction of the generated information bits is repeated d times (called the degree) where $d > 2$. For example, with $d = 5$ the bits of higher degrees are well protected because their *a posteriori* values include 5 extrinsic information bits instead of 2. The nonuniform repetition divides the information bits into groups indexed by $i = 1,\ 2,\ 3\ldots$, with each group having a certain number of repetitions d_i where $d_i = 2,\ 3, \ldots T$, and T is the maximum number of repetition. The number of bits in a group i is a fraction f_i of the total number of information bits. The bits belonging to group i are repeated d_i times by the nonuniform repetition. The output of the repeater is then randomly interleaved and passed onto an encoder. Information on the random interleaver, bit degree selection and their corresponding fractions used in this paper is explained in Sect. 4. The parity vector $\mathbf{p}\in \mathfrak{B}^{p\times 1}$ from the block encoder and the original $\mathbf{k}\in \mathfrak{B}^{a\times 1}$information vector before the nonuniform repetition are then transmitted in a block format, with the parity vector appended to the information vector for systematic encoding. If $\mathbf{k}$ is the information vector, $\mathbf{v}$ the transmitted vector and $\mathbf{p}$ the parity vector, then the following

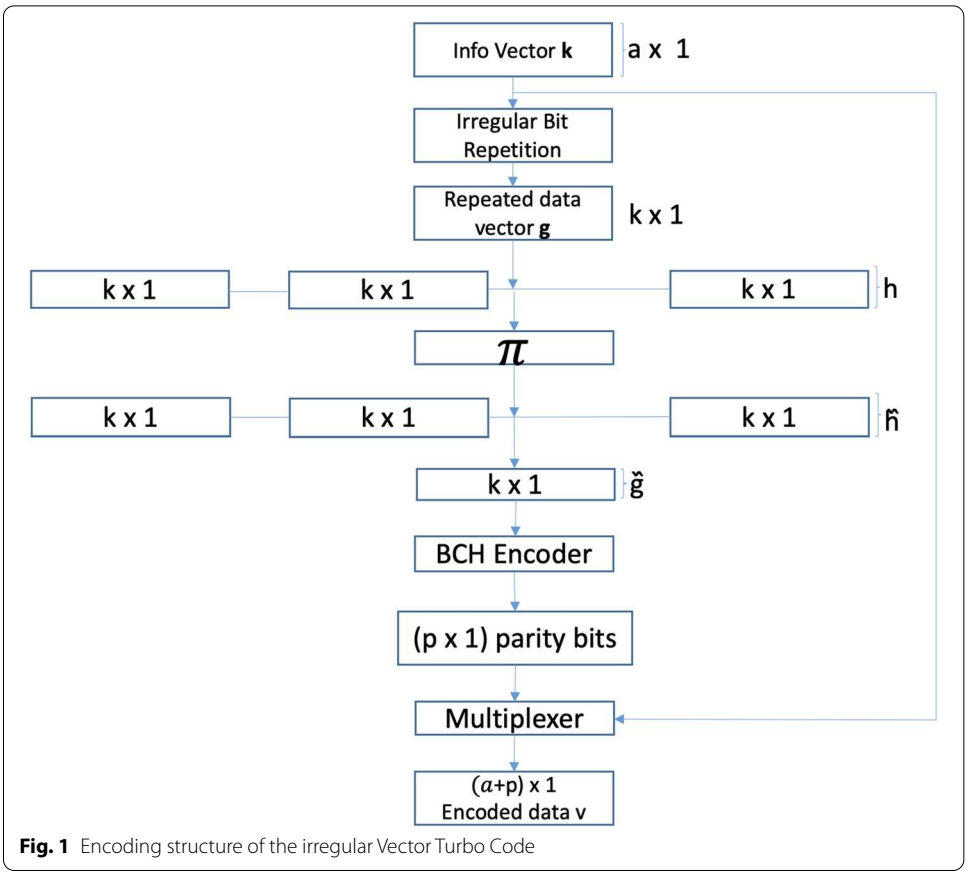

Fig. 1 Encoding structure of the irregular Vector Turbo Code

equations (1, 2 and 3) show the relationship between the average bit degree, the fractions f_i and their various nonuniform repetition degrees d_i.

$$\sum_{i=2}^{T} f_i = 1. \tag{1}$$

$$\sum_{d_i=2}^{T} d_i f_i = \ddot{d}, \tag{2}$$

where $\ddot{d}$ is the average bit degree.

$$r = \frac{|k|}{|k| + |p|}. \tag{3}$$

In (3), r is the code rate of the iVTC, while $|k|$ and $|p|$ represent the length of vectors **k** and **p**, respectively. It should be noted that, the degrees (d_i) used in constructing an iVTC are limited by the block size of its corresponding TPC. Let us consider a systematic BCH code $(n, k, \partial)$where n is the codeword length, k the number of information bits and ∂ the minimum hamming distance. In generating the **k** information vector, there is a limit to the dimension of **k** that could be generated and grouped before being repeated to give the required length for a given systematic BCH code. For example, an iVTC derived

from a (127, 120, 4) systematic BCH code could have its **k** information vector as 1 by 60 bits, i.e., one row and 60 columns. Using a degree 2 repetition on vector **k** produces **g**, a 1 by 120 vector (i.e., fraction $f_i = 1$ and degree $d_i = 2$) with i indicating the group index. In this case, we have only one group with its fraction f_i (i.e., $i = 1$ and f_i =1). This ensures that **g** retains the original dimension of the TPC information block before encoding. Before interleaving, $|g|$ (i.e., the length of repeated information bits) number of rows is stored with each row a 1 by 120 vector as explained earlier.

In this study, the number of rows of data bits stored equals the number of rows in a regular TPC for fair comparison. More details on this are given in Sect. 5, showing the sensitivity of the iVTC to the number of rows chosen. This then produces an array of dimension 120 by 120. Random interleaving of this array is then performed (for this example the interleaver depth is 120×120). Rows of the interleaved (120 by 120) array are individually read out as vector **h** and separately encoded using the (127,120, 4) systematic BCH code. Parity vector **p** from each encoding is then attached to the originally generated **k** information bit vector for transmission. Before transmission, $|h|$(i.e., length of repeated information bits) number of rows is again stored with each row a 1 by 67 vector. This example produces a high rate ($\frac{60}{67} = 0.896$) iVTC of degree 2 with the information vector **k**, a 1 by 60 vector and the transmit vector **v**, a 1 by 67 vector (parity vector **p** is 1 by 7). In general, the repeated **h** vector should be equal to the original dimensions of the TPC information block before encoding. In terms of block sizes, a (127,120)TPC, i.e., a TPC encoded horizontally and vertically with a (127, 120, 4) BCH encoder (i.e., each information bit is encoded twice), would require a block size of (127×127) with a code rate of 0.89, while a corresponding iVTC to the (127,120)TPC described above would require a transmit block size of (120×67) with code rate 0.896 (i.e., ~ 0.9), where $|h|$ equals 120 in this case.

4 Decoding the irregular Vector Turbo Code

This section gives a detailed description for the decoding of the iVTC consisting of a nonuniform repetition block, a Log-MAP and an Extrinsic Computational Block (ECB).

4.1 Iterative decoder structure

In this study, a modified BCJR algorithm, the Log-MAP algorithm, has been used as the decoder for the regular TPC and the iVTC as originally proposed in [17] and subsequently used in [11]. Firstly, the received codeword $\overrightarrow{\boldsymbol{v}} \in \mathfrak{B}^{(a+p)\times 1}$ is demultiplexed into vectors $\overrightarrow{\boldsymbol{k}} \in \mathfrak{B}^{a\times 1}$ and $\overrightarrow{\boldsymbol{p}} \in \mathfrak{B}^{p\times 1}$, corresponding to the transmitted **k** information and **p** parity bits. $\overrightarrow{\boldsymbol{k}}$ is then repeated using the same repetition pattern at the encoder to generate a vector $\overrightarrow{\boldsymbol{g}} \in \mathfrak{B}^{k\times 1}$. k such vector is concatenated into a longer vector $\overrightarrow{\boldsymbol{h}} \in \mathfrak{B}^{k^2\times 1}$, which is randomly interleaved using the same encoder interleaving pattern to form the vector $\widehat{\boldsymbol{h}} \in \mathfrak{B}^{k^2\times 1}$. Vector $\widehat{\boldsymbol{h}}$ is partitioned into k shorter vectors $\widehat{\boldsymbol{g}} \in \mathfrak{B}^{k\times 1}$ each of which is passed into the Log-MAP decoder with its corresponding parity bits $\overrightarrow{p}$ together with an initial a priori value of equal probability, i.e., zero log likelihood vector **a**. Extrinsic information vector $\boldsymbol{e} \in \mathfrak{B}^{k\times 1}$ is then gleaned out from the Log-MAP decoder. k such vector is concatenated into a longer vector $\boldsymbol{e} \in \mathfrak{B}^{k^2\times 1}$, which is de-interleaved (using the interleaving pattern from the encoder as a guide) to form the vector $\widetilde{\boldsymbol{e}} \in \mathfrak{B}^{k^2\times 1}$. Vector $\widetilde{\boldsymbol{e}}$ is partitioned into k shorter vectors $\widetilde{\boldsymbol{e}} \in \mathfrak{B}^{k\times 1}$ before being passed into the extrinsic computational block. At

every iteration, each information bit of degree $d_i$will have a new extrinsic information value which is the product, or sum when using the log likelihood values, of the other d_{i-1} extrinsic information values. k such vector is again concatenated into a longer vector $\widetilde{\boldsymbol{a}} \in \mathfrak{B}^{k^2\times 1}$, which is randomly interleaved to form the vector $\mathbf{a}\in \mathfrak{B}^{k^2\times 1}$. Vector **a** is partitioned into k shorter vectors $\boldsymbol{a} \in \mathfrak{B}^{k\times 1}$ as new *a priori* values for the next decoding iteration. After the final iteration, k such vectors are concatenated into a longer vector $\widetilde{\boldsymbol{m}} \in \mathfrak{B}^{k^2\times 1}$, for a final de-interleaving to form the vector $\boldsymbol{m} \in \mathfrak{B}^{k^2\times 1}$ and finally partitioned into k shorter vectors $\boldsymbol{k} \in \mathfrak{B}^{k\times 1}$. The repetition pattern is then removed from **k** for comparison with the originally generated $a \times 1$information bits **k.**

Figure 2 depicts the decoding process where π and $\pi^{'}$denote the interleaving and de-interleaving functions, respectively.

In this paper, a different random interleaver pattern has been used for each block of data to be interleaved. This implies that the random interleaver pattern used for the first data block differs from the random interleaver used for the second data block and so on, i.e., the random interleaver itself is seeded randomly.

In the extrinsic computational block depicted in Fig. 2, the extrinsic information at every k^{th} bit E_{jk} with a degree of repetition of d_i is recalculated in the log domain using (4).

$$E_{jk} = \sum_{\substack{l=1 \\ l\neq k}}^{d_i} E_{jl}. \tag{4}$$

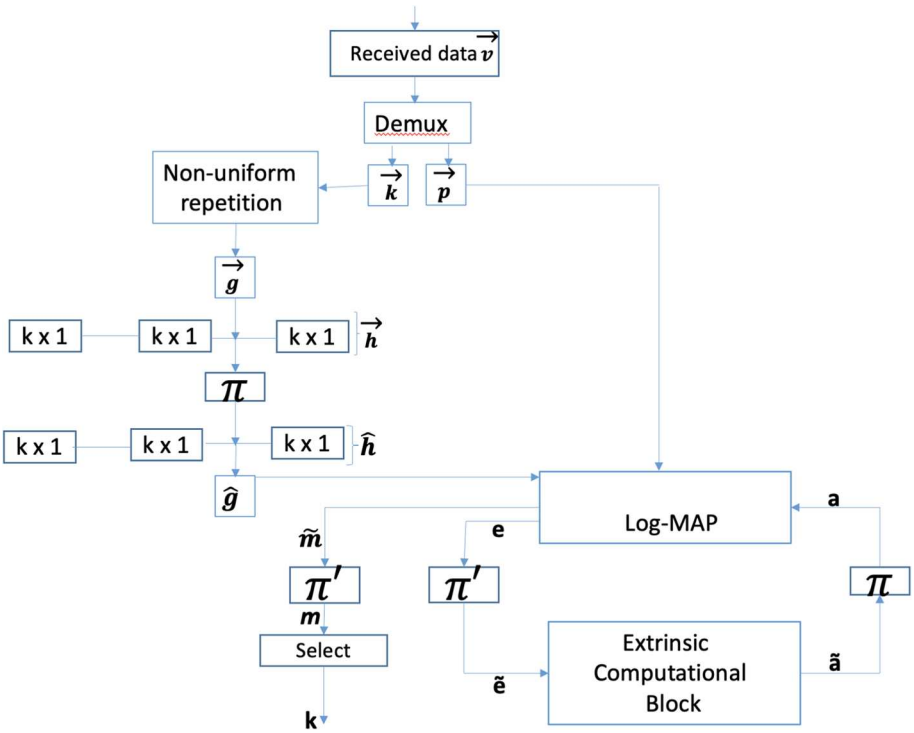

Fig. 2 Iterative decoding structure of an irregular Vector Turbo Code

4.2 The EXIT function of an irregular Vector Turbo Code and its area property

The distance to capacity for a channel codec determines how efficient the channel codec is. In this section, a description of the computation of the area underneath an EXIT function is given. As stated earlier, the iVTC requires only one decoder, as such, only one EXIT function can be derived from the decoding process, unlike a typical iterative decoding process that required two decoders. In the latter case, we derive two EXIT functions which are used to plot an EXIT chart. The EXIT function of an iVTC can be used to determine how close the iVTC is to capacity in bits per channel use. The area underneath the EXIT functions of the iVTCs and their corresponding TPCs are calculated in Sect. 5 together with their corresponding attainable capacities. These values were then used to compute the distance between the attainable capacities and their corresponding throughput in bits per channel use.

In Extrinsic Information Transfer charts (EXIT charts), transfer characteristics based on mutual information are used to describe the flow of extrinsic information through the soft-in/soft-out (SISO) constituent decoders of an iterative decoder. A decoding trajectory is then used to visualize the exchange of extrinsic information between the constituent decoders (i.e., the EXIT functions) [13]. In an iVTC, the constituent code is a single BCH encoder which can be seen as the inner encoder in a serially concatenated code, i.e., the BCJR decoder of the iVTC directly collects the channel observation as is the case for an inner code of a serially concatenated code. In [9], R.G Maunder and L. Hanzo showed that the EXIT chart of short block lengths codes such as a BCH gives results that are very close to infinite length codes with the use of EXIT band charts. That is, a slight variation occurs between the EXIT chart of an infinite length block code and a short length block code that is not significant.

Figure 3 is a schematic diagram showing how the EXIT function of an iVTC is computed where the letters "a" to "d" represent the various LLR ratios at certain points of the schematic diagram. A priori values of a predetermined mutual information between the transmitted and the received LLR is then sent into the decoder and its corresponding extrinsic information for each mutual information of the a priori is then calculated, given as $\mathrm{d}'_e - \mathrm{c}'_a$ in Fig. 3. The mutual information of these a priori values ranges from 0 to 1. A plot of all the predetermined mutual information on the abscissa against its corresponding extrinsic information on the ordinate gives the EXIT function.

In [16], the authors have shown that the area A underneath the EXIT function of an inner code is given by equation (5).

$$A = \int_0^1 I_{\mathrm{e}}(I_{\mathrm{a}})\mathrm{d}I_{\mathrm{a}} = \frac{I_{\max\,(X\,;\,Y)}}{R_{\mathrm{in}}}, \tag{5}$$

where I_{e} denotes the extrinsic output values, I_{a} is the a priori input values, I_{e} a function of I_{a}, i.e., $I_{\mathrm{e}}\ (I_{\mathrm{a}})$, $I_{\max}(\mathrm{X};\mathrm{Y})$ is the maximum mutual information transfer between the transmitted symbol X and received symbol Y, also known as the capacity, and R_{in} is the rate of the inner code. I_{e} which is a function of I_{a} is integrated with respect to I_{a} (i.e., $\mathrm{d}I_{\mathrm{a}}$). This implies that for a rate one inner code, the area underneath the inner code equals the capacity (C) of the communication channel. In cases where the inner code has a code rate less than unity, i.e., $R_{\mathrm{in}} < 1$, the area underneath the inner code is

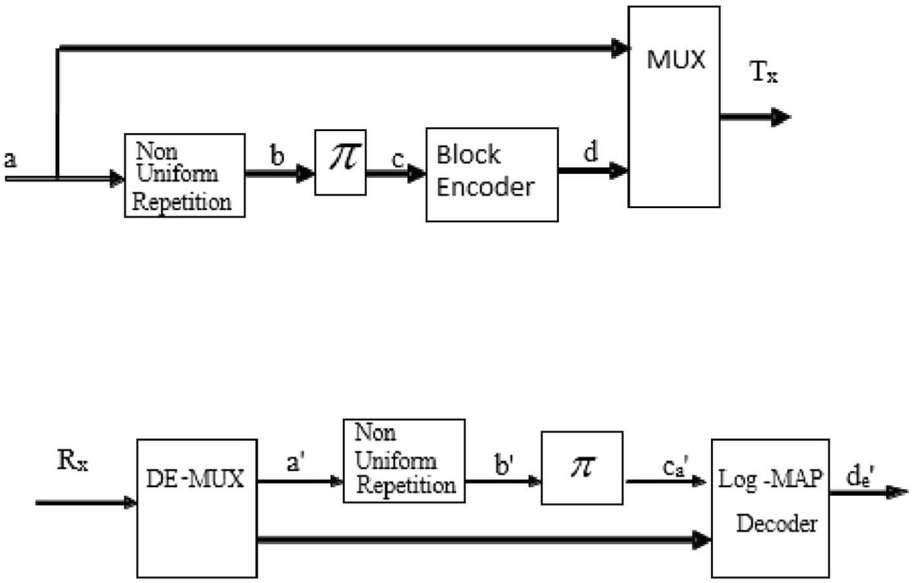

Fig. 3 Schematic of an irregular Vector Turbo Code arrangement in plotting an EXIT function. The corresponding LLR sequences used in the receiver are indicated using an apostrophe. A subscript of "a" indicates a priori information, while a subscript of "e" is used for extrinsic information

an attainable capacity (C_{A}) (i.e., a slightly lower capacity bound) which is slightly lower than the capacity of the communication channel as given by Maunder and Hanzo [9]. In higher order modulations, the attainable capacity (C_{A}) is given by (6) where M is the M-ary modulation order [15].

$$C_{\mathrm{A}} = A \times R_{\mathrm{in}} \times \log_2 M. \tag{6}$$

Also,

$$C_{\mathrm{A}} = C = A \times \log_2 M, \text{ when } R_{\mathrm{in}} = 1. \tag{7}$$

The attainable capacity is the upper capacity bound in bits per channel use for any inner code with its code rate less than one in a Gaussian modeled digital communications channel. The TPCs and the iVTCs used in this paper all possess inner code rates less than one. In Sect. 5, the effective throughputs in bits per channel use for each channel codec are compared to their corresponding attainable capacities to determine the capacity loss for each channel codec for different modulation schemes.

4.3 Bit degree combination

Selecting an appropriate bit degree profile and its corresponding fraction is not trivial since they depend on the block length of the code. At present, there is no algorithmic means to derive the bit degree combination. However, from observation it is highly recommended to have a fraction of the information bits repeated once (degree 2 bits) as a substantial part of the bit degree profile. This fraction f_i for the degree 2 bits should also be within a certain neighborhood range for best observed performance. Also for good performance, the number of groupings (see Sect. 3) should not exceed 3. A limiting

factor to the number of bit degrees (apart from degree 2) and their frequency of repetition is the number of information bits in the linear code also explained in Sect. 3. In general, it was observed that the degree 2 profile must have its fraction f_i between 75 and 99% of the original information bits, while the remaining fraction is shared between degrees which vary depending on the modulation scheme. A degree 2 repetition for say 60% of the information block matrix has been observed to produce a BER performance in the iVTC worse than its equivalent TPC. With respect to the above explanation a good bit degree combination would have 90% of the information block matrix with a degree profile of 2, another 6% of the information block matrix with a degree profile of 3, while the remaining 4% of the information block matrix would have a degree profile of 4. Table 2 in Sect. 5 shows the various iVTCs used in this paper together with their degree profiles and their corresponding fractions f_i. In recent publications on irregular codes, there is no algorithmic means of determining the initial bit degree selection, rather what exists are programs used to close the tunnel gaps in irregular codes. However, the iVTC is constructed from block codes of fixed length, the irregular repeat of bits must fit the available block length. This in turn significantly increases the challenge of developing an algorithm. The search for an algorithmic means of determining f_i is the topic of future research.

5 Simulation results and discussion

This section illustrates and discusses the system BER performance and the throughput curve of the iVTC in comparison to the regular TPC. Simulation of the system was done using MATLAB for an AWGN channel. In this study, a single decode iteration of an iVTC consists of only one Log-MAP calculation, whereas in a TPC a single decode iteration consists of two Log-MAP calculations. Thus two decode iterations in an iVTC are equivalent to a single decode iteration in the TPC. In this study, the block length of the regular TPC is twice the block length for the "degree 2" iVTC. As stated earlier in Sect. 3, the BER performance in AWGN of the Turbo Product Code in [7] and its equivalent uniform repetition of degree 2 is only the same for large block lengths (i.e., block lengths > 120 bits). Simulation results showing the BER versus E_b/N_0 curve have been plotted for an AWGN channel using the BPSK, QPSK, 16QAM and 64QAM modulation schemes as well as using different iVTC and TPC block sizes to illustrate the coding gain of the iVTC over its equivalent TPC. Also, plotted is the throughput (S) in bits per channel use for the various IVTCs and TPCs in different modulation schemes (BPSK, QPSK, 16QAM and the 64QAM) versus the signal power to noise power ratio (SNR), illustrating a more efficient bandwidth usage in the iVTC over its equivalent TPC. In terms of computational complexity, the number of operations required to achieve a low BER in the TPC and the iVTC was also compared. Each bit in the decoding Log-MAP trellis requires 23 mathematical operations. Firstly, an operation is required to add the received channel information to the a priori values. This is followed by a single operation required to calculate the branch transition probability for each bit. Seven operations are then required for the forward recursion calculation per bit. This includes the exact Jacobian logarithm for the Log-MAP approximation. The same number of operations is also required for the backward recursion. A single operation is then required to calculate the a posteriori transition log-confidences. Lastly six operations are required to calculate

the a posteriori log-likelihood values which also include a Jacobian operation. On the other hand, the extrinsic computational block requires four mathematical operations per bit, i.e., two operations to calculate the original length of the repeated bits, an operation to group the bits into the various groups i, and an operation to sum them. The above explanation then gives a total of 1449, 2921 and 5865 operations for a Log-MAP decoding in a (63, 57), (127, 120) and (255, 247)TPC, respectively. Table 1 shows the number of decoding operations required for convergence in the various iVTCs and TPCs taking into account the number of iterations. Table 1 also shows that, twelve iVTCs require fewer operations in comparison to the their corresponding TPCs, with a significant 41.3% reduction in the number of operations required in the QPSK (108, 100)iVTC coupled with a coding gain of up to 0.98dB over its corresponding TPC (see Table 2). Also, two iVTCs require about the same number of operations in comparison to their corresponding TPCs, with six iVTCs requiring a higher number of operations in comparison to their corresponding TPCs. This shows that a good number of the iVTCs require fewer operations to converge on a low BER.

Table 2 shows the length of information vector $\mathbf{k}$ and the transmit vector $\boldsymbol{\nu}$ used for the various fractions f_i and degrees d_i for the iVTCs together with the minimum E_b/N_0 in dB required to achieve a probability of error $\leq 10^{-5}$ for the TPC in comparison to the iVTC. The various fractions f_i and degrees d_i used for the iVTCs in this study are the most effective combinations found in terms of BER performance. The performance

Table 1 Number of operations required to achieve a low BER in the TPCs and the iVTCs

Code	Mod	Iterations		Operations		%Δ
		TPC	iVTC/R	TPC	iVTC	
(63,57)	BPSK	3	6/0.77	8694	10,206	− 17.3
	QPSK	7	9/0.77	20,286	15,309	24.5
	16QAM	8	7/0.77	23,184	11,907	48.6
	64QAM	2	4/0.77	5796	6804	− 17.3
(127,120)	BPSK	5	17/0.90	29,210	55,233	− 89.0
			10/0.88		32,490	− 11.2
			6/0.85		19,494	33.2
	QPSK	10	11/0.90	58,420	35,739	38.8
			38/0.88		123,462	− 111.3
			18/0.85		58,482	− 0.1
	16QAM	10	14/0.90	58,420	45,486	22.1
			13/0.88		42,237	27.7
			16/0.85		51,984	11.0
	64QAM	7	9/0.90	40,894	29,241	28.4
			13/0.88		42,237	3.2
			14/0.85		45,486	− 11.2
(255,247)	BPSK	11	10/0.93	129,030	68,850	46.6
	QPSK	15	15/0.93	175,950	103,275	41.3
	16QAM	10	17/0.93	117,300	117,045	0.2
	64QAM	8	11/0.93	93,840	75,735	19.2

[a] The (63, 57), (127, 120) and (255, 247)TPC have fixed code rates of 0.82, 0.89 and 0.94, respectively. %Δ in the table stands for the percentage change in the decoding operations with respect to TPC, R the iVTC code rate, operations as the number of operations and Mod as Modulation

Table 2 Comparison between the required E_b/N_0 for TPC and iVTC to achieve a low BER (10^{-5}) in an AWGN channel

Code	Modulation	TPC E_{b/N_0} (dB)	iVTC E_{b/N_0} (dB)	K, N (iVTC)	iVTC bit degree and (fractions)
(63,57)	BPSK	3.35	3.60	20, 26	2(0.9),10(0.05) 11(0.05)
	QPSK	3.55	3.50	20, 26	2(0.85), 5(0.1) 13(0.05)
	16QAM	7.04	6.40	20, 26	2(0.85), 7(0.05) 8(0.1)
	64QAM	12.55	11.18	20, 26	2(0.80), 7(0.15) 4(0.05)
(127,120)	BPSK	3.69	3.80	60, 67	2(1)
			3.60	50, 57	2(0.9) 6(0.1)
			3.40	40, 47	2(0.9), 7(0.05) 17(0.05)
	QPSK	5.10	5.15	60, 67	2(1)
			4.80	50, 57	2(0.9) 6(0.1)
			4.30	40, 47	2(0.9), 7(0.05) 17(0.05)
	16QAM	7.78	7.90	60, 67	2(1)
			7.30	50, 57	2(0.96) 12(0.04)
			7.10	40, 47	2(0.9) 12(0.1)
	64QAM	12.48	12.10	60, 67	2(1)
			11.50	50, 57	2(0.98) 22(0.02)
			11.20	40, 47	2(0.9), 9(0.05) 15(0.05)
(255,247)	BPSK	4.24	4.16	100,108	2(0.97),18(0.02) 17(0.01)
	QPSK	8.13	7.15	100,108	2(0.97),19(0.02) 15(0.01)
	16QAM	10.06	9.45	100,108	2(0.97),19(0.02) 15(0.01)
	64QAM	14.47	13.59	100,108	2(0.97),19(0.02) 15(0.01)

of the TPCs and their equivalent iVTCs were firstly, evaluated in terms of BER versus E_b/N_0 and then in terms of throughput in bits per channel use versus SNR. These metrics have been evaluated for four different digital modulation schemes, namely; BPSK, QPSK, 16QAM and 64QAM.

Figure 4 shows the BER performance of the (63, 57)TPC in comparison to its equivalent iVTC for BER values up to 10^{-5}. The results show that the (63, 57)TPC rate 0.82 code performs slightly better than the (26, 20)iVTC rate 0.77 code by 0.25dB at a BER of 10^{-5} in a BPSK scheme. In the case of higher order modulation schemes, e.g., QPSK and 16QAM, the coding gain at a BER of 10^{-5} for the (26, 20)iVTC increases to 0.05dB and 0.64dB, respectively, over the rate 0.82 TPC. The largest recorded coding gain over the TPC is the (26, 20)IVTC for a 64QAM modulation scheme, with a significant 1.37dB coding gain at a BER of 10^{-5} as shown in Fig. 4. Early error floors in the rate (26, 20) iVTC are observed in the BER plots of Fig. 4. These early error floors do not occur in larger block length iVTCs as illustrated in Figs. 5 and 6 which show the BER performance of the (127, 120) and (255, 247)TPCs in comparison to its equivalent iVTCs.

For these larger block sizes, the rate 0.85 and 0.88 iVTCs have higher coding gains than the rate 0.89 TPC for all four modulation schemes investigated. We observed that, the iVTCs for the 64QAM modulation schemes recorded the highest coding gain over its corresponding TPC for most of the codes. Similarly, the iVTCs for the 16QAM modulation scheme recorded a higher coding gain over the TPC for the QPSK and BPSK modulation schemes. This underpins the strength of an irregular code with its error

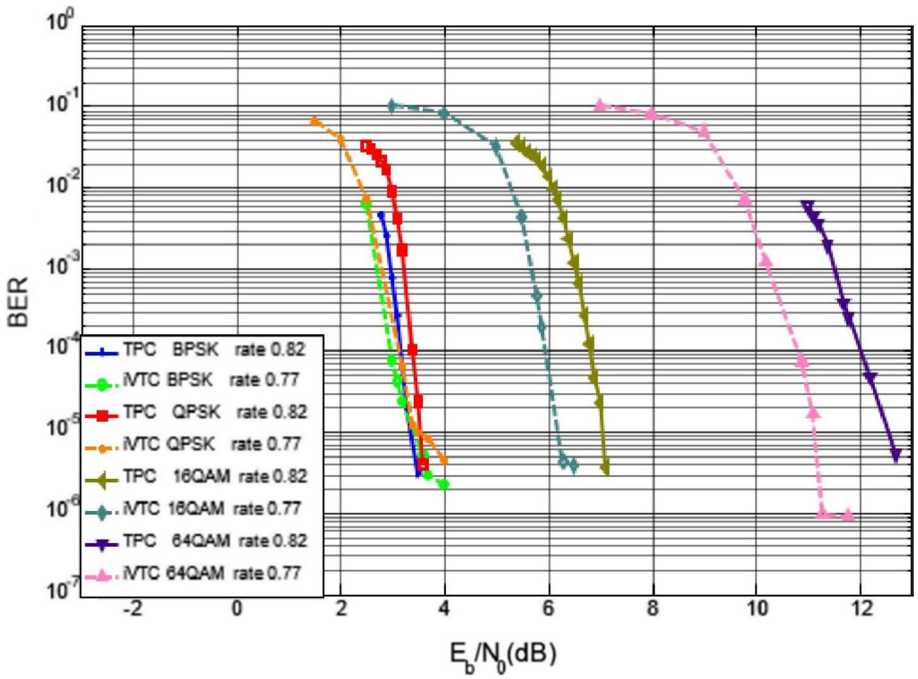

Fig. 4 (63, 57)TPC and equivalent iVTC BER versus E_b/N_0 for BPSK, QPSK, 16QAM and 64QAM

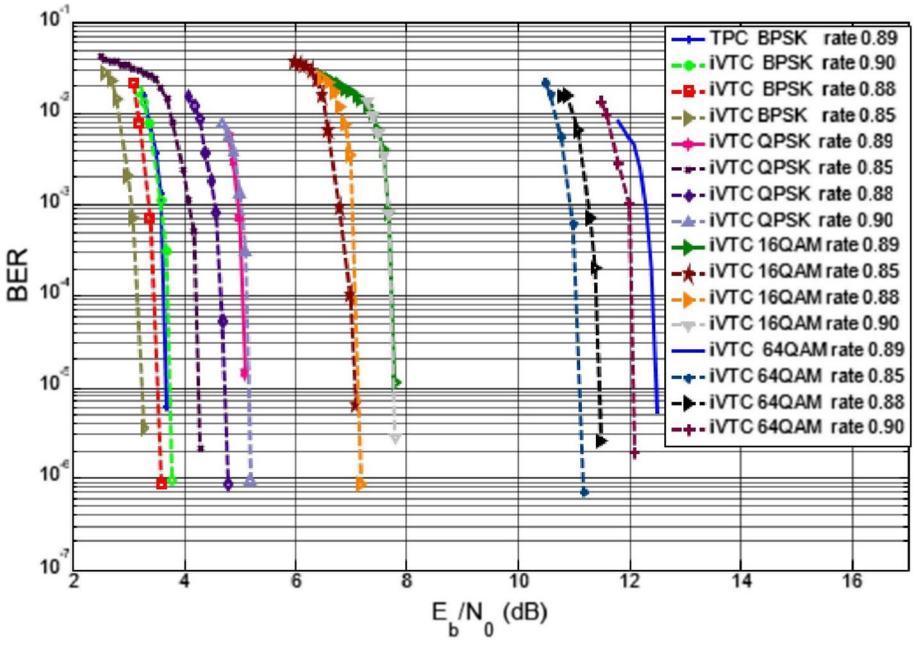

Fig. 5 (127, 120)TPC and equivalent iVTC BER versus E_b/N_0 for BPSK, QPSK, 16QAM and 64QAM

correcting capabilities in higher order modulation schemes. The performance of the iVTC improves as the modulation depth increases due to the higher probability of error in the higher order modulation schemes, and an increased number of protection bits per

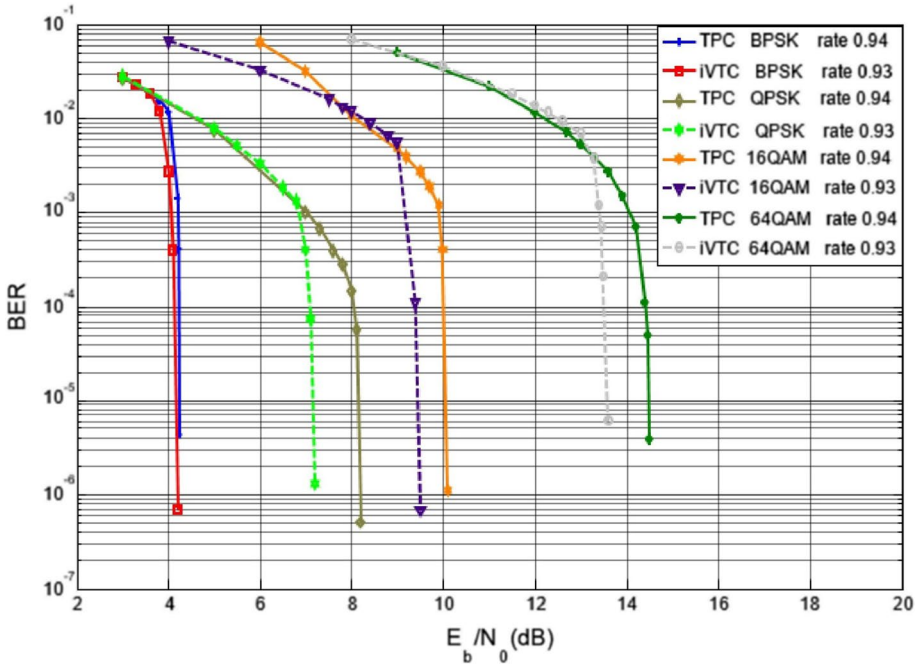

Fig. 6 (255, 247)TPC and equivalent iVTC BER versus E_b/N_0 for BPSK, QPSK, 16QAM and 64QAM

symbol when compared to the lower order modulation schemes when iterated. From Eq. (4), higher order modulation schemes will in general have more a priori information per symbol passed into the Extrinsic Computational Block (ECB), hence gleaning out stronger (in terms of likelihood) extrinsic information for the next iteration. The irregular code due to its unequal protection of information bits, recovers most of its original (generated transmitted) information bits at the receiver earlier than other information bits during iterative decoding, thereby enhancing the correct decoding of the remaining information bits. This means that for higher order modulation schemes such as the 64QAM which has a high noise level, the iVTC is able to recover the originally transmitted bits better than the 16QAM with a lower noise level. The degree 2 iVTC with a code rate of 0.90 (67, 60)iVTC corresponding to the (127, 120)TPC has a very similar performance in comparison to the rate 0.89(127, 120)TPC with respect to distance to capacity as shown in Table 3. In the case of the (108, 100)iVTC (corresponding to the (255, 247) TPC), BER performance shown in Fig. 6, the iVTC with code rate 0.93 records as much as 1dB coding gain over its equivalent rate 0.94 TPC in a QPSK modulation scheme.

The BER versus E_b/N_0 sensitivity to interleaver depth for the iVTC is examined, with the view of comparing the BER performance of the iVTCs for different interleaver depths. The comparison is made in the QPSK modulation scheme. The investigation was performed on the current interleaver depth of (57 by 20), (120 by 40) and (247 by 100) corresponding to the (26, 20), (47, 40) and the (108, 100)iVTC. This was done by increasing their interleaver depths by 25% and then decreasing them by 25% and 50%. The results show that an increase in the interleaver depth for the (47,40)iVTC from (120 by 40) to (150 by 40) (i.e., 25% increment) resulted to a coding gain slightly larger than the (120 by 40) interleaver depth. On the other hand, a reduction in the interleaver depth

Table 3 Distances to capacity for TPCs and iVTCs

Code	Modulation	D(TPC) (dB)	D(iVTC) (dB)	iVTC code rate
(63, 57)	BPSK	1.30	1.58	0.77
	QPSK	1.34	1.42	0.77
	16QAM	1.71	1.99	0.77
	64QAM	3.64	3.66	0.77
(127,120)	BPSK	0.61	0.49	0.90
			0.60	0.88
			1.46	0.85
	QPSK	2.02	2.00	0.90
			1.93	0.88
			1.74	0.85
	16QAM	1.35	1.41	0.90
			1.05	0.88
			1.30	0.85
	64QAM	2.25	1.82	0.90
			1.58	0.88
			1.78	0.85
(255, 247)	BPSK	0.28	0.53	0.93
	QPSK	4.26	3.53	0.93
	16QAM	2.70	2.41	0.93
	64QAM	3.47	2.64	0.93

[a] The (63, 57), (127, 120) and (255, 247)TPCs have fixed code rates of 0.82, 0.89 and 0.94, respectively. D(TPC), D(iVTC) in the table represent, the distance to capacity for the TPC and the iVTC in dB, respectively

from (120 by 40) to (90 by 40) and (60 by 40) (i.e., 25% and 50% reduction) resulted to an appreciable coding loss with respect to the (120 by 40) interleaver depth. The same analysis can be said for the (108, 100) and the (26, 20)iVTCs where the larger interleaver depth resulted in a slightly larger coding gain with the smaller interleaver depth resulting in an appreciable coding loss with respect to their original interleaver depth. This relatively makes the original interleaver depth a better option in terms of BER performance versus complexity since complexity is also a function of the interleaver depth.

The throughput per channel use is given by $S = R \times \left(\log_2 M\right)(1 - BLER)$, where R is the rate of the code, $BLER$ the block error rate and M the M-ary order of the modulation scheme. A block in the TPC is the size of the original information bits ($k \times k$), while a block in the iVTC consists of h rows of k information bits, i.e., $h \times k$.

The behavioral patterns of the throughput curves in Figs. 7, 8 and 9 for the iVTC and the TPC in an AWGN channel show a continuous coding gain of the iVTCs over their corresponding TPCs in all four modulation schemes investigated. This is so because the iVTCs have converged to a very low probability of error before the TPCs can achieve any appreciable throughput value. An exception to this is the degree 2 iVTC with the same throughput performance as the TPC as shown in Fig. 8. The Shannon capacity curve shown in Figs. 7, 8 and 9 was calculated by (8).

$$\frac{C}{B} = \log_2\left(1 + \frac{S}{N}\right) \text{bit/s/Hz}. \tag{8}$$

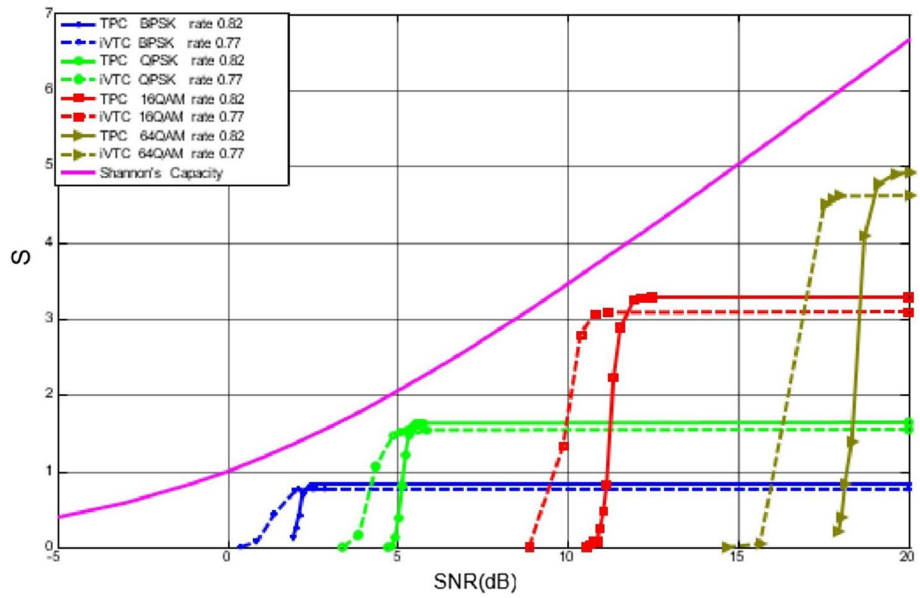

Fig. 7 (63, 57)TPC and equivalent iVTC throughput per channel use versus SNR. ***S*** represents bits/channel use

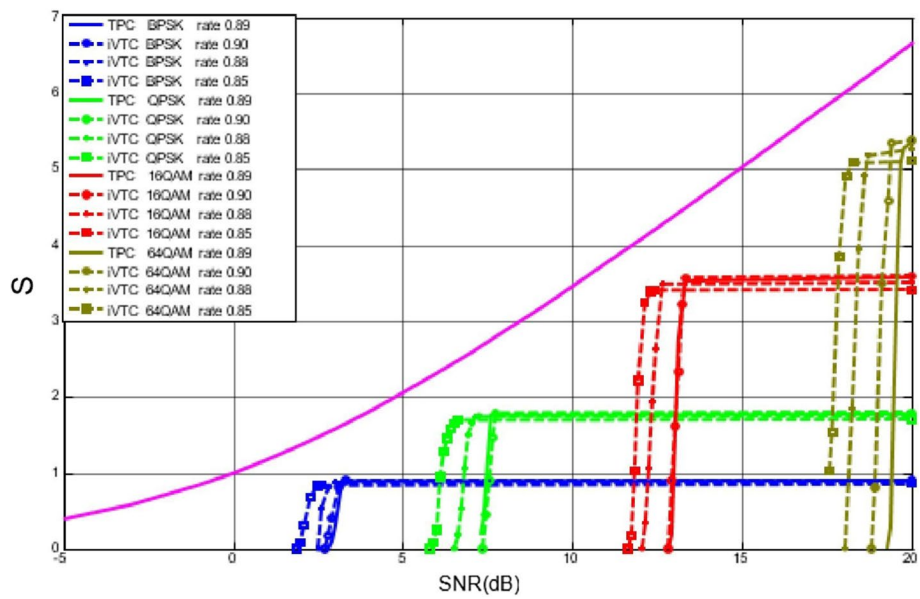

Fig. 8 (127,120)TPC and equivalent iVTC throughput per channel use versus SNR. ***S*** represents bits/channel use

In Sect. 4, (6) was used to compute the attainable capacity C_A of the channel for a TPC where R_{in} is the rate of the inner code, which is also the code rate seen by the Log-MAP decoder at the receiver section, i.e., (n, k) (where n is the block length and k, the information bit length, making the code rate seen by the Log-MAP decoder $\frac{k}{n}$). In the case of the iVTC, there is no inner code as such, hence the rate R_{in} used in (6) for the case of the iVTC is the same as the code rate seen by the Log-MAP decoder for the iVTC, i.e., $\frac{k}{n}$.

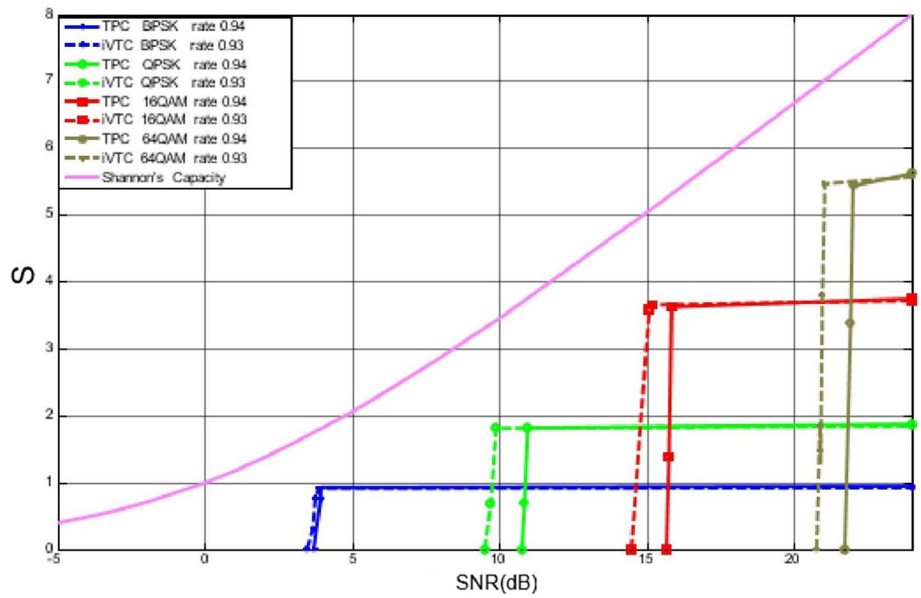

Fig. 9 (255,247)TPC and equivalent iVTC throughput per channel use versus SNR. ***S*** represents bits/channel use

Table 4 Distance to capacity in bits per channel use for TPC and iVTC

Code	Modulation	D(TPC)	D(iVTC)
(63, 57)	BPSK	0.05	0.10
	QPSK	0.10	0.20
	16QAM	0.22	0.31
	64QAM	0.44	0.54
(127, 120)	BPSK	0.04	0.03
			0.04
			0.05
	QPSK	0.10	0.09
			0.10
			0.13
	16QAM	0.16	0.12
			0.16
			0.20
	64QAM	0.24	0.15
			0.27
			0.20
(255,247)	BPSK	0.04	0.10
	QPSK	0.91	0.40
	16QAM	1.15	0.22
	64QAM	0.96	0.47

[a] D(TPC), D(iVTC) in the table represent, the distance to attainable capacity for the TPC and the iVTC in bits per channel use, respectively

Table 4 shows the distance in bits per channel use between the various codes (TPC and iVTC) and their corresponding attainable capacities computed from their EXIT functions. From Table 4, the iVTC with larger block sizes are closer to the attainable channel

capacity in comparison to their corresponding TPCs (an exception is the (108, 100)iVTC in a BPSK modulation scheme). The (47, 40), (57, 50), (67, 60) and the (108, 100)iVTCs give good performance due to their large block sizes and the extra protection offered to certain information bits by the repetition code. In comparison to the best known LDPC codes (regular and irregular), the iVTC requires a far smaller number of iterations to converge to a low BER as reported in [5, 10, 15]. Also the iVTC has a higher rate in comparison to well known LDPC codes. In [15] Divsalar et al. reported a well known (8176, 7156) LDPC code for a BPSK modulation scheme and code rate 0.875 that converged at 3.78dB at a BER of 10^{-5}. This result was originally reported in [5]. In comparison to an equivalent iVTC code rate 0.88, the iVTC requires an E_b/N_0 value of 3.6dB, resulting in a coding gain of 0.18dB. The iVTC also requires 10 iterations in comparison to the reported 50 iterations required for the LDPC code rate 0.875. In terms of code length, the iVTC's are about half the length of the LDPC codes reported by the authors in [5, 10, 15]. Also in [6], irregular LDPC code of rates 0.75, 0.8, 0.83 and 0.857 with 50 iterations requires an E_b/N_0 of 4.3dB, 4.4dB, 4.45dB and 4.6dB, respectively, to converge to a low bit error rate of 10^{-5} for a BPSK modulation scheme. In comparison to the iVTC, the iVTC of rates 0.77, 0.85, 0.88, 0.90 and 0.93 requires an E_b/N_0 of 3.6dB, 3.4dB, 3.6dB, 3.8dB and 4.16dB, respectively, as well as requiring only 6, 6, 10, 17 and 10 iterations, respectively, in order to converge to a low bit error rate of 10^{-5} for a BPSK modulation scheme.

The new iVTC previously termed an irregular Block Turbo Code [1], like the TPC, is a high rate code, thereby saving bandwidth during transmission and ensuring efficient bandwidth utilization. This is so because, for every channel use, the parity bits present in high rate codes are low in comparison to the information bits, thereby ensuring maximum utilization of the channel for information purposes. Potential applications of the iVTCs are in digital communication channels with Gaussian noise such as fixed Broadband (point to point and point to multi-point), VSAT modems (data and voice), optical fiber communication systems and other microwave point to point links.

6 Conclusion

In this paper a novel, irregular Vector Turbo Code (iVTC) for Gaussian noise systems, is presented for the first time. The new channel codec is a high speed channel codec. The iVTC is also a flexible channel codec in terms of construction. The iVTC is closer to Shannon's capacity in comparison to the existing Turbo Product Code and capable of converging to a very low probability of error (10^{-5}) at a significant coding gain of 1.28dB for the (127, 120)iVTC over its corresponding TPC for 64QAM modulation. This ensures a large energy savings over a long period of time with the use of the iVTC over its corresponding TPC. Equally important is the lower complexity of the iVTC over the equivalent TPC, as the iVTC in some cases requires almost 46% fewer operations to achieve a low BER. In general, the proposed iVTC due to its extra protection of information bits performs no worse and frequently much better than the existing TPC. The authors are currently investigating the BER performance of the new iVTC in a multipath wireless channel.

Abbreviations
TPC: Turbo Product Code; BTC: Block Turbo Code; BCJR: Bahl, Cocke, Jelinek and Raviv algorithm; iVTC: Irregular Vector Turbo Code; AWGN: Additive White Gaussian Noise; EXIT: Extrinsic Information; SISO: Single-input single-output; BER: Bit error rate; LDPC: Low-density parity-check.

Authors' contributions
A.S (corresponding author) carried out the research studies, conducted the experiments which includes the channel codec design, analysis, simulation, result evaluation, paper writing and editing. T.O supervised the research studies, providing guidelines, supervision of the research which includes defining research goal, providing structure to the paper and result evaluation. T.O also edited the paper. All authors read and approved the final manuscript.

Funding
This research was funded by the Overseas Research Scholarship (ORS) in the U.K. The contribution of the ORS was a part tuition fees contribution toward a Ph.D degree.

Availability of data and materials
Data sharing not applicable to this article as no datasets were generated or analyzed during the current study.

Declarations

Competing interests
There are no financial and non-financial competing interest in regards to this paper.

Author details
[1]Swansea University, Swansea, West Glamorgan, UK. [2]Department of Electronic and Electrical Engineering, University of Sheffield, Sheffield, UK.

Received: 7 November 2021 Accepted: 16 March 2022
Published online: 04 April 2022

References

1. A.O. Sholiyi, Irregular Block Turbo Codes for Communication Systems. Ph.D. dissertation, College of Eng., Swansea University, Swansea (2011)
2. H.E. Sawaya, J.J. Boutros, Irregular Turbo Codes with symbol-based iterative decoding, in *3rd Inter. Symp. on Turbo Codes and Related Topics* (2003)
3. B. Frey, D. Mackay, Irregular Turbo Codes, in *Proc. 37th Allerton Conf., Illinois* (1999)
4. X. Jaspar, L. Vandendorpe, Joint source-channel codes based on irregular turbo codes and variable length codes. IEEE Trans. Commun. **56**(11), 1824–1835 (2008)
5. O. Book, Low density parity check codes for use in near-earth and deep space. Consulative Committee for Space Data Systems (CCSDS) **131**(2) (2007)
6. S.J. Johnson, S.R. Weller, A family of irregular LDPC codes with low encoding complexity. IEEE Commun. Lett. **7**(2), 79–81 (2003)
7. R.M. Pyndiah, Near-optimum decoding of product codes: block turbo codes. IEEE Trans. Commun. **46**(8), 1003–1010 (1998)
8. C. Berrou, A. Glavieux, P. Thitimajshima, Near Shannon limit error-correcting coding and decoding: Turbo-Codes, in *Proc. ICC*, pp. 1064–1070 (1993)
9. R.G. Maunder, L. Hanzo, Iterative decoding convergence and termination of serially concatenated codes. IEEE Trans. Veh. Technol. **59**(1), 216–224 (2010)
10. J. Boutros, G. Caire, E. Viterbo, H. Sawaya, S. Vialle, Turbo Code at 0.03dB from capacity limit codes, in *Proc. IEEE International Symposium on Information Theory* (2002)
11. J. Hagenauer, E. Offer, L. Papke, Iterative decoding of binary block and convolutional codes. IEEE Trans. Inf. Theory **42**(2), 429–445 (1996)
12. G.M. Kraidy, V. Savin, Irregular Turbo Codes design for binary erasure channel, in *5th Inter. Symp. on Turbo Codes and Related Topics* (2008)
13. S. ten Brink, Convergence behavior of iteratively decoded parallel concatenated codes. IEEE Trans. Commun. **49**(10), 1727–1737 (2001)
14. 3GPP: Technical specification Group Radio Access Network; Multiplexing and channel coding (TDD), Release 8, version 8.4.0 March (2009)
15. K.S. Andrews, D. Divsalar, S. Dolinar, J. Hamkins, C.R. Jones, F. Pollara, The development of turbo and LDPC codes for deep-space applications. Proc. IEEE **95**(11), 2142–2156 (2007)
16. A. Ashikhmin, G. Kramer, S.T. Brink, Code rate and the area under extrinsic information transfer curves, in *Proc. IEEE Inter Symp. on Inf. Theory*, pp. 115 (2002)
17. L.R. Bahl, J. Cocke, F. Jelinek, J. Raviv, Optimal decoding of linear codes for minimizing symbol error rate. IEEE Trans. Inf. Theory **20**(2), 284–287 (1974)

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.