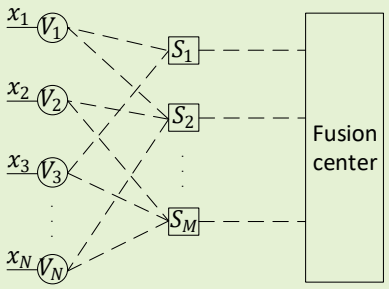
Tree-Search Techniques for Joint iterative Compressive Sensing and LDPC Decoding in Wireless Sensor Networks

Jue Chen, Tsang-Yi Wang, *Member, IEEE*, Jwo-Yuh Wu, *Member, IEEE*, Chih-Peng Li, *Fellow, IEEE*, Soon Xin Ng, *Senior Member, IEEE*, Robert G. Maunder, *Senior Member, IEEE* and Lajos Hanzo, *Life Fellow, IEEE*

***Abstract*— Novel techniques are conceived for joint Compressive Sensing (CS) and Low-Density Parity Check (LDPC) coding in Wireless Sensor Networks (WSNs), namely a Soft-Input Soft-Output (SISO) tree search Sphere Decoding (SD) technique, and a SISO Hamming Distance (HD) based solution. Factor graphs are utilized to describe the connectivity between the signals and sensors, as well as with the LDPC codes. In the Fusion Center (FC), the factor graphs may be used for iterative joint LDPC-CS decoding, in order to recover the signals observed. However, the CS decoder of the FC suffers from high complexity, if the exhaustive Maximum A posteriori (e-MAP) technique is employed, which considers all possible combinations of source signals detected by each of the associated sensors. Hence, in the proposed SD and HD schemes only the more likely combinations of source signals are tested for reducing the CS decoding complexity. More specifically, a tree search technique is used in the first step to find the most likely combination of source signal values. Then, in the second step, the proposed SD continues the tree search to find a set of alternative hypotheses. This facilitates the generation of high quality *extrinsic* information, which may be iteratively exchanged with the LDPC decoder. By contrast, in the HD approach, the second step obtains the alternative hypotheses within a certain HD of the most likely source signal combination. Both our BLock Error Rate (BLER) results and Extrinsic Information Transfer (EXIT) charts show that the proposed SD and HD techniques approach the performance of the full-search e-MAP approach at a significantly reduced complexity. In particular, we show the e-MAP solution is about 56 times more complex than the SD approach and around 210 times more complex than the HD approach. Compared to a Separate Source-Channel Coding (SSCC) hard information benchmarker, the proposed SISO schemes improve the decoding performance by about 1.7 dB. Furthermore, the SISO schemes allow the iterations inside the CS decoding to eliminate the error floors and obtain a further 2.45 dB gain.**



***Index Terms*— Joint source-channel coding, compressive sensing, LDPC coding, sphere decoding, tree search, Hamming distance technique, EXIT charts, factor graph**

L. Hanzo would like to acknowledge the financial support of the Engineering and Physical Sciences Research Council projects EP/P034284/1 and EP/P003990/1 (COALESCE) as well as of the European Research Council's Advanced Fellow Grant QuantCom (Grant No. 789028). Wang's work is supported by MOST of Taiwan under grant 110-2221-E-110-021. J.-Y. Wu's work is supported in part by the Ministry of Science and Technology of Taiwan under grants MOST 108-2221-E-009-025 MY3 and MOST 110-2634-F-009-025, in part by the Higher Education Sprout Project of the National Yang Ming Chiao Tung University and Ministry of Education of Taiwan, and in part by the MOST Joint Research Center for AI Technology and All Vista Healthcare. Prof. Chih-Peng Li would like to acknowledge the financial support from the Ministry of Science and Technology, Taiwan, under grant numbers MOST 108-2218-E-110-014 and MOST 109-2218-E-110-006.

Corresponding Author: L. Hanzo, J. Chen, S. Shao and S.-X. Ng and R.-G. Maunder are with School of Electronics and Computer Science, University of Southampton, SO17 1BJ, UK. (E-mail: lh@ecs.soton.ac.uk; jc7m17@soton.ac.uk; ssecs1994@gmail.com; sxn@ecs.soton.ac.uk; rm@ecs.soton.ac.uk).

T.-Y. Wang is with the Institute of Communications Engineering, National Sun Yat-sen University, Kaohsiung 804, Taiwan (E-mail: tcwang@mail.nsysu.edu.tw).

J.-Y. Wu is with the Institute of Communications Engineering, National Yang Ming Chiao Tung University, Hsinchu 300, Taiwan (E-mail: jywu@cc.nctu.edu.tw).

I. Introduction

Shannon's source-channel separation theorem [1] proves that there is no disadvantage in separating source coding from channel coding [2] under certain idealized circumstances. More specifically, Shannon shows that near-capacity communication does not require the source and channel coding to be performed jointly, provided that the delay and complexity are not limited. However, in finite block length transmission over non-stationary channels, Joint Source-Channel Coding (JSCC) (For convenience, all acronyms used in this paper have been summarized in Table I) has been shown to offer significantly improved coding gain [3] over Separate Source-Channel Coding (SSCC) when relying on limited complexity. More explicitly, JSCC uses source coding to compress a source signal by removing the uncontrollable source redundancy,

C.-P. Li is with the Institute of Communications Engineering, National Sun Yat-sen University, Kaohsiung, Taiwan, and also with the Department of Electrical Engineering, National Sun Yat-sen University, Kaohsiung 80424, Taiwan (E-mail: cpli@faculty.nsysu.edu.tw).

while channel coding is employed to improve the error correction performance by adding carefully controlled redundancy before transmission [4], where the reduction and addition of redundancy is performed jointly between these two processes.

TABLE I: Acronyms employed throughout this paper.

BLER	BLock Error Rate
CNs	Check Nodes
CND	Check Node Decoder
CS	Compressive Sensing
e-MAP	exhaustive Maximum A Posterior
EXIT	EXtrinsic Information Transfer
FC	Fusion Center
HD	Hamming Distance
i.i.d.	Independent and Identically Distributed
JSCC	Joint Source-Channel Coding
LDPC	Low-Density Parity Check
LLRs	Logarithmic-Likelihood Ratios
MI	Mutual Information
QPSK	Quadrature Phase Shift Keying
SD	Sphere Decoding
SISO	Soft-Input Soft-Output
SNs	Sensor Nodes
SND	Sensor Node Decoder
SSCC	Separate Source-Channel Coding
TSs	Time Slots
VN	Variable Nodes
VND	Variable Node Decoder

In particular, JSCC has been demonstrated to play a beneficial role in Wireless Sensor Networks (WSNs) [5] to alleviate the problems of constrained energy [6]–[8], as well as unreliable communication channels [9], [10]. Compressive Sensing (CS) technology has been widely adopted in WSNs to solve energy-constrained problems. In [11] the authors use CS in WSN in the presence of an eavesdropper. They propose a technique to maintain secrecy in the presence of an eavesdropper and achieve an excellent signal detection performance. Mohammadi *et al.* [12] focus their attention on the distributed detection problem of a localized phenomenon of interest resulting in sparse signals at each sensor, where CS and quantization are employed at each sensor. Recently, the combination of JSCC and CS has received particular attention in the literature of WSNs. In [13], Chen *et al.* examine the trade-off between the required transmission energy and the quality of the recovered signals in a WSN that employs a CS coded JSCC scheme. Their results show that the CS technique is robust to both channel fading and noise. In [14], CS was used for source coding and linear network coding for channel coding, in order to recover the signal within a tolerance distortion level. In both [13] and [14], a large amount of traffic is generated by each of the N sensors, which can only be reduced moderately by the compression performed in each sensor [15]. More specifically, a set of N signals may be compressed into a set of M ($M < N$) observations made by a set of M sensors if only K out of N signals are activated at a time, which avoids an excessive amount of network traffic. Furthermore, WSNs typically relying on hostile channels, motivating the application of channel coding to protect the signals during transmission. In particular, Low-Density Parity Check (LDPC) codes are widely used in CS coded JSCC schemes [16], [17], as a benefit of their excellent performance.

Soft-Input Soft-Output (SISO) techniques constitute the core of iterative decoding [18]. In JSCC schemes employing SISO techniques for the concatenated source and channel code [19], iterative joint source-channel decoding [20], [21] may be used for significantly improving the performance of hard decoding [22] for Separate Source-Channel Coding (SSCC) schemes. In the case of a SISO decoder used for CS, an approximate Maximum A Posteriori (MAP) decoder may be employed for discovering the presence/absence of the association between the signals and sensors, as well as for inferring the source statistics [23]. The e-MAP technique is capable of reliably inferring the specific association of N signals with M sensors and it iteratively exchanges its soft values with an LDPC decoder. However, the MAP decoder suffers from high complexity when considering all possible associations between sensors and signals [24], which will be referring to the exhaustive MAP (e-MAP) in this paper. To elaborate further, there may be many connected signals for each sensor with many possible combinations of values, which leads to a potentially excessive complexity upon considering every possible combination in the e-MAP decoder.

Against this background, we propose a pair of novel techniques for reducing the complexity of SISO CS decoding in the iterative JSCC receiver. Sphere Decoding (SD) is typically used for reducing the complexity of signal detection in order to solve minimum Euclidean distance problems in WSNs [25], [26]. In [25], the authors propose a SD algorithm that performs maximum-likelihood decoding based on an extended distance metric. In [26], channel-quality-aware decision fusion rules are conceived for MIMO systems, and a generalized SD is employed for circumventing the exponentially escalating complexity. In this treatise, we also employ the SD technique for reducing the decoding complexity imposed by searching for the most likely candidates. However, in our scheme, we search for the most likely candidates by directly relying on the node metrics calculated using LLRs rather than on the node metrics calculated using Euclidean distance [25], [26]. More specifically, we conceive a SD and a Hamming Distance (HD) based technique to search for the most likely combination of the source signal and associated sensor value, which contains not only the most likely combination but also the most likely alternative hypotheses. Explicitly, these alternative hypotheses are important for generating reliable *extrinsic* information, which may then be iteratively exchanged with the concatenated channel decoder. We commence by using a tree search to find the most likely combination of the source signal and associated sensor value based on the prior knowledge about the sparsity level and presence/absence of association between signals and sensors, as well as the *a priori* information provided by the concatenated channel decoder. Following this, the SD finds

the alternative hypotheses by continuing the tree search, while the HD-based technique finds the alternative hypotheses by considering the HD between the alternative hypotheses and the most likely combination. In this way, we avoid traversing all possible combinations and significantly reduce the complexity of CS decoding.

TABLE II: Contrasting our contribution to the state-of-the-art.

	[13]	[14]	[16]	[17]	[19]	[27]	This work
JSCC	√	√	√	√	√	√	√
WSNs	√	√	√	√			√
CS	√	√	√	√			√
LDPC			√	√	√		√
SD						√	√
HD							√
SISO tree search							√
EXIT chart analysis					√		√

We boldly and explicitly contrast our contributions to the state-of-the-art in Table II and detail them below:

1 We reduce the complexity of SISO CS decoding by using a tree search to find the most likely combination of the source signal and associated sensor values before using the SD to find the set of likely alternative hypotheses, in order to glean high quality *extrinsic* information, which can be iteratively exchanged with the concatenated channel decoder. Unlike the traditional SD routinely used for JSCC or for MIMO detection based on the distance between adjacent points [27]–[29], the proposed scheme performs tree search-based SD relying on the input Logarithmic-Likelihood Ratios (LLRs). Furthermore, our scheme exploits the additional constraints imposed by the sparsity upper bound K' and the presence/absence of connectivity between the source signals and sensors, which make the application of SD a challenge. We demonstrate that the proposed SD reduces the SISO CS decoding complexity by a factor of 56 compared to the e-MAP approach.

2 Furthermore, as a complement to the SD used for finding alternative hypotheses, we also conceive a HD based technique. More specifically, all combinations of source signal and the associated sensor values within a certain HD of the most likely combination are considered as alternative hypotheses. Our results show that the proposed HD technique approaches the error rate of the e-MAP at a complexity, which is about 210 times lower.

3 We provide the Extrinsic Information Transfer (EXIT) chart based characterization of the proposed solutions. In this way, we characterize the iterative exchange of *extrinsic* information between the CS decoder as well as the concatenated channel decoder, and show that both the proposed SD and HD techniques approach the performance of the e-MAP approach.

The rest of this paper is organized as follows. A top-level description of the system model and of the iterative joint LDPC-CS decoder are provided in Section II. Then the SD approach and HD techniques are introduced in Section III. Section IV characterizes the EXIT functions and decoding trajectory of the proposed scheme, before our complexity analysis in Section V. Our simulation results are provided in Section VI before offering our conclusions in Section VII.

Throughout this paper, the following notations are used; $\mathbf{A}$, $\boldsymbol{a}$, A: matrix, vector and scalar; $\mathbf{A}_i$: the i^{th} matrix among all matrices; $\boldsymbol{a}_i$: the i^{th} row of matrix $\mathbf{A}$; $a_{i,j}$: the j^{th} element of vector $\boldsymbol{a}_i$; $\tilde{a}$, $\tilde{a}^a$, $\tilde{a}^e$: LLR, *a priori* LLR, *extrinsic* LLR; $\tilde{a}_{i\to j}$: LLR from the i^{th} node to the j^{th} node; $\hat{a}$: estimated value of a; $\mathbf{A}^T$: transpose.

II. System Model

This section begins by describing the structure of the proposed system model in Section II-A, before the flow of the iterative joint LDPC-CS decoding is discussed in Section II-B.

A. Top level description of the proposed system model

The proposed system model is shown in Fig. 1. The N Variable Nodes (VNs) in Fig. 1 represent N signals, where each VN V_n ($n \in [1, N]$) can be considered to be a random variable, adopting a vector $\boldsymbol{x}_n \in \{0,1\}^{1\times T}$ of T non-equiprobable binary values across T Time Slots (TSs). Here, the binary signal values are sparse, with only $K \ll N$ of the N signals in each TS adopting a value of 1, with all of the other $N-K$ binary values adopting a value of 0. In this treatise, we assume that the sparsity upper bound $K', 1 \leq K \leq K'$ is prior knowledge available at the Fusion Center (FC). In this context two scenarios are considered, the first one is when the true sparsity $K = K'$ is known *a priori* to the FC and the other is when the true sparsity K is unknown to the FC, but the upper bound of the sparsity $K', 1 \leq K \leq K'$ is known to the FC. In a practical CS-based WSN, the residual-based algorithms of [30] may be used either for directly estimating the sparsity or its upper bound during the training phase in the FC [30]. The binary value of the n^{th} signal in the t^{th} TS is represented by the notation $x_{n,t}$, where $n \in [1, N]$ and $t \in [1, T]$, allowing the representation of the n^{th} signal vector as $\boldsymbol{x}_n = [x_{n,1}, \cdots, x_{n,T}]$.

The proposed system model adopts M sensors, which form the WSN to observe the N signals, where the sparsity upper bound K' of the system is exploited to facilitate $M < N$. Each sensor is represented by the notation S_m, where $m \in [1, M]$, and is used for simultaneously observing several signals. In the proposed system model, the association between the signals and sensors are chosen randomly, while are exemplified by the dashed lines in the factor graph of Fig. 1. Here, the edges of the factor graph are characterized by the sensing matrix $\mathbf{\Phi} \in \{0,1\}^{M\times N}$, which is assumed to be prior knowledge available at the FC. Here, the n^{th} column of the sensing matrix represents the association of the n^{th} signal, while the m^{th} row represents the association of the m^{th} sensor. More specifically, a value of 1 in the sensing matrix indicates the association between the n^{th} signal, $n = 1, \ldots, N$ and the m^{th} sensor, $M = 1, \ldots, M$, while a value of 0 indicates no association. In the proposed system model, each sensor may observe both different number of signals and different combinations of the signals, where the degree of each sensor identifies the number of connected signals it observes, corresponding to the number

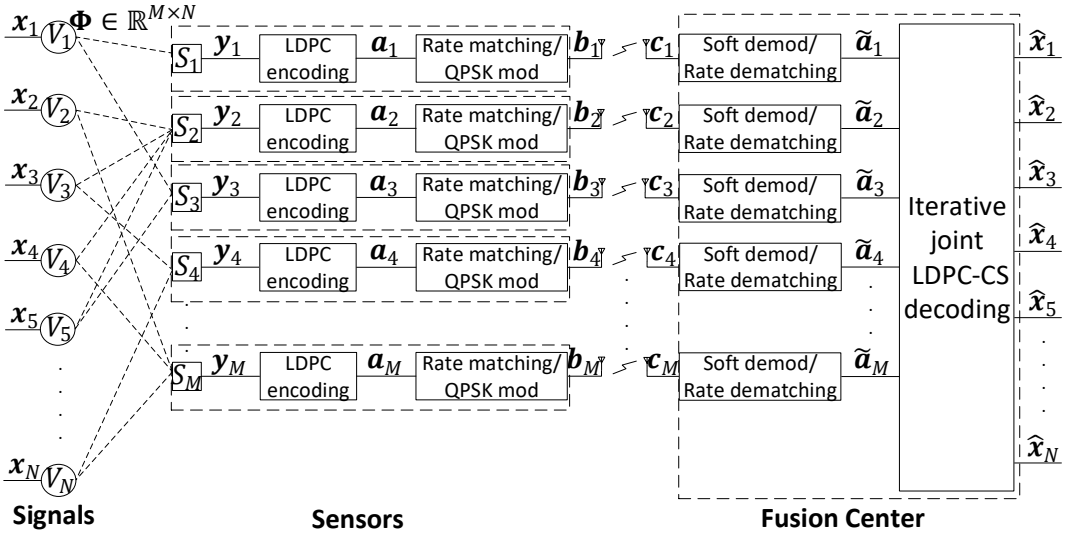


Fig. 1: Block diagram illustrating the relationship between signals, sensors and the Fusion Center (FC).

of 1 in each row of $\mathbf{\Phi}$. Then, the set of degrees may be stored in a vector $\boldsymbol{d} = [d_1, \cdots, d_m, \cdots, d_M]^T$. However, in the proposed system model, the number of sensors used for observing each signal is a fixed value K_s, which means that the number of 1 in each column of $\mathbf{\Phi}$ is fixed to K_s. Furthermore, the randomly selected connectivity between signals and sensors is assumed to be time-invariant across the T TSs. The proposed scheme may be applied in a real-life application of non-invasive wild-life monitoring in a nature habitat. For example, we may consider a case where there are K wild animals living in a habitat, which has been divided into N number of one-square-meter tiles, wherein M microphones have been installed throughout the habitat to detect the actions of wild animals. Each microphone can detect wild animal activity within several neighbouring tiles. In the case where we know that K wild animals live in the habitat, we will know that exactly K tiles will be occupied in any time slot and hence these microphones allow us to track wild animals in a non-invasive manner

In the proposed system model, the sensors perform an OR function to combine the binary values provided by the set of connected signals. Explicitly, a sensor outputs a binary values of 0 if all connected signals provide value of 0 at a particular TS and a sensor output a value of 1 otherwise. In this way, the N binary signals provided in each TS are compressed to M binary observations provided by the M sensors. More specifically, the observation of the m^{th} sensor is a binary vector $\boldsymbol{y}_m = [y_{m,1}, \cdots, y_{m,T}]$ comprising T observations across the T TSs. Following this, each of the M sensors processes its vector of observations $\boldsymbol{y}_m$ using an LDPC encoder [31] having a parity check matrix $\mathbf{H} \in \{0,1\}^{W \times U}$ and coding rate $R = \frac{T}{E}$ in order to generate the encoded bit sequence $\boldsymbol{a}_m = [a_{m,1}, \ldots, a_{m,U}]$ having a length U. Then rate matching adjusts the encoded bit sequence length to E. Afterwards, Quadrature Phase Shift Keying (QPSK) modulation is performed individually for each pair of encoded bits before transmission in order to obtain the complex vector $\boldsymbol{b}_m \in \mathbb{C}^{1 \times \frac{E}{2}}$. Here, the symbol energy E_s is normalized to 1.

The complex vector $\boldsymbol{b}_m$ is transmitted over an AWGN channel in order to obtain the received complex vector $\boldsymbol{c}_m$, while the components of the AWGN are zero-mean, Independent and Identically Distributed (i.i.d.) complex Gaussian random variables with variance N_0. Hence, the Signal-to-Noise Ratio (SNR) is given by $\frac{E_s}{N_0}$. Soft demodulation for QPSK [32] and rate dematching are performed upon the received signal $\boldsymbol{c}_m$, in order to obtain the channel LLRs $\tilde{\boldsymbol{a}}_m = [\tilde{a}_{m,1}, \cdots, \tilde{a}_{m,E}]$, which are forwarded to the FC as inputs of the iterative joint LDPC-CS decoder detailed in Section II-B.

B. Top level description of the proposed iterative joint LDPC-CS decoding scheme

The iterative joint LDPC-CS decoding block of Fig. 1 is comprised of the factor graphs shown in Fig. 2. These factor graphs are comprised of many Check Nodes (CNs), VNs, and Sensor Nodes (SNs). Algorithm 1 describes the operation of these various nodes during the iterative joint LDPC-CS decoding process. The function of the SNs will be introduced in Section III, while the operations of the CNs and VNs is described in [33].

The factor graphs of Fig. 2 include both LDPC decoder and CS decoding. Here, the LDPC decoder is comprised of the Variable Node Decoder I (VNDI) and Check Node Decoder (CND), having W VNs and U CNs in each layer. The number of LDPC decoding layers is equal to the number of sensors M, where each layer represents the CS decoder of the corresponding sensor. More specifically, the CNs and VNs in the same layer work together, while the M layers are operated in parallel but not directly together, as shown in Fig. 2. Similar to LDPC decoding, CS decoding comprises the Sensor Node Decoder (SND) and Variable Node Decoder II (VNDII), where the former has M SNs in each of the T layers, while the latter has N VNs in each layer. To elaborate further, each layer of CS decoding considers one TS, which is in contrast to LDPC decoding, where each layer considers one sensor. This creates a time/sensor grade, which allows the TSs of the CS decoding to indirectly help each other via the

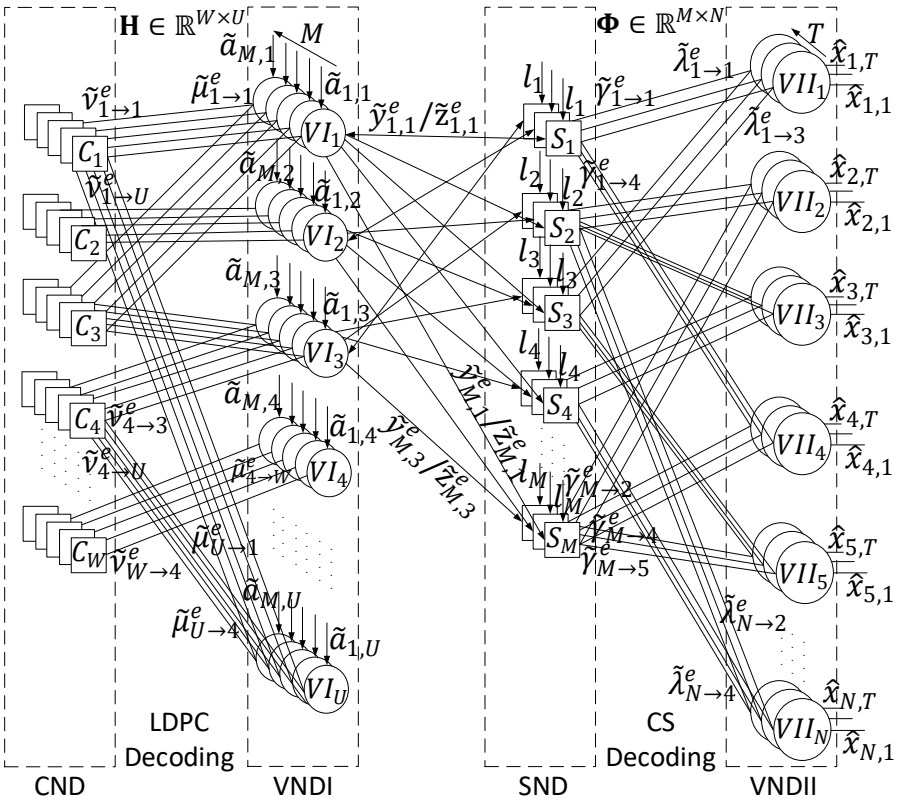


Fig. 2: The factor graph to show the iterative exchange of LLRs between VND I, CN decoder for LDPC decoding, SN and VND II for CS decoding.

sensors of the LDPC decoder and vice versa. More specifically, the SNs and VNs II in the same layer work together directly, whilst those of different layers work in parallel but without direct interaction, as shown in Fig. 2.

The input of the $m^{th}, m \in [1, M]$ layer in the LDPC decoder is the rate-dematched channel LLRs $\tilde{\boldsymbol{a}}_m$, which is generated as discussed in Section II-A and shown in Fig. 1. During iterative joint LDPC-CS decoding, the m^{th} layer can also take input from the CS decoder, which provides the *a priori* LLRs $\tilde{\boldsymbol{y}}^a_m$. Each VN processes the LLRs gleaned from its connected CNs, CS decoding SNs as well as channel LLRs, and then passes the *extrinsic* LLRs $\tilde{\boldsymbol{\mu}}^e$ to its connected CNs. Following this, each CN is operated and *extrinsic* LLRs $\tilde{\boldsymbol{\nu}}^e$ are passed to the connected VNs. Here, we add subscripts to the LLRs within the LDPC decoder and CS decoder to represent the connectivity of the factor graph edge upon which the LLR is exchanged. For example, $\tilde{\mu}^e_{U\to 1}$ represents an *extrinsic* LLR being exchanged from the U^{th} VN I to the first CN. The connectivity of VNs and CNs in each layer depends on the factor graph adopted by the particular LDPC coding standard, with all layers having the same connectivity. In this work, we employ the 3GPP 5G NR LDPC code Base Graph (BG) 2 [31]. Following the completion of i_{LDPC} number of iterations between the VNDI and CND in the LDPC decoder, the *extrinsic* LLRs $\tilde{\mathbf{Y}}^e$ generated by the VNDI comprise all *extrinsic* LLRs gleaned from all VNs I in the M layers, which are forwarded to the connected SNs. The subscripts of the LLRs exchanged between the LDPC and CS decoder represent the layers they belong to. For instance, $\tilde{y}^e_{M\to 3}$ represents an *extrinsic* LLR passed from the M^{th} layer in the LDPC decoder to the 3^{rd} layer of the CS decoder. The connectivity between the CS decoding SNs and VNs I is based on the layers they belong to. More specifically, in the LDPC decoder, the $u^{th}, u \in [1, T]$ VN I in the $m^{th}, m \in [1, M]$ layer connects to the m^{th} sensor in the t^{th} layer of the CS decoder, where we have $u = t$. The number U of VNs in the VNDI is typically higher than T, since the encoded LDPC bits comprise both systematic bits and parity bits. However, the CS decoder can only provide *extrinsic* information pertaining to the T systematic bits and hence it is not connected to the $U - T$ VNs that characterize parity bits.

The input of the $t^{th}, t \in [1, T]$ layer in the CS decoder is constituted by the *a priori* LLRs $\tilde{\boldsymbol{z}}^a_t$ provided by the LDPC decoder for the SNs. Furthermore, CS decoding also benefits from *a priori* knowledge of the probability of each compressed signal $\boldsymbol{y}_m, m = 1, \ldots, M$ having the value 0, where this knowledge is represented by the LLRs $\boldsymbol{l} = [l_1, \ldots, l_M]$, which are added to the *a priori* LLRs $\tilde{\boldsymbol{z}}^a_t$ provided by the LDPC decoder, as shown in Fig. 2. To elaborate further, the LLRs representing the probability that the m^{th} compressed signal $\boldsymbol{y}_m$ adopts the binary value 0 is given by $l_m = \log\left(\frac{P_m}{1-P_m}\right)$, where $P_m = (P_0)^{d_m}$ and P_0 is the probability of occurrence of 0 in each TS. Further still, the $m^{th}, m = 1, \ldots, M$ SN in the t^{th} layer takes the *a priori* LLR inputs $\tilde{\gamma}^a$ from the connected

VNs II of the CS decoder. Following this, the SN processes $\tilde{z}^a_{m\to t}$, $\tilde{\gamma}^a$ in order to generate the *extrinsic* LLRs $\tilde{\gamma}^e$, which are passed to its connected VNs II. In response, each VN II generates the *extrinsic* LLRs $\tilde{\lambda}^e$, which are then passed back to the SNs, where the SNs and the VNs II iterate over i_{CS} iterations, as shown in Algorithm 1. The connectivity between the SNs and VNs II in each layer mirrors the connectivity between the signals and sensors in Fig. 1. Following the completion of these iterations, all SNs in all T layers generate the *extrinsic* LLR $\tilde{\mathbf{Z}}^e$, which are passed back to the LDPC decoder as *a priori* LLRs $\tilde{\mathbf{Y}}^a$. Like the VNs I, the VNs II operate according to [33]. By contrast, the operations of the SNs is detailed in Section III-A.

After completing $i_{LDPC-CS}$ iterations between the LDPC and CS decoder, all VNs II in the CS decoder output the *a posteriori* LLRs $\tilde{\mathbf{\Lambda}}^p$. Final, SSCC hard decisions are made in order to obtain the estimated signals $\hat{\mathbf{X}} = [\hat{\boldsymbol{x}}_1, \ldots, \hat{\boldsymbol{x}}_N]^T$. Here, the *a posteriori* LLR $\tilde{\lambda}^p$ of a VN II is obtained by adding up all *a priori* LLRs $\tilde{\lambda}^a$ arriving from its connected SNs. Then, the final decision is made by exploiting the knowledge that there are K' signals having a binary value of 1 in each of the T TSs. Accordingly, the *a posteriori* LLRs $\tilde{\boldsymbol{\lambda}}^p_t$ in the $t^{th}, t = 1, \ldots, T$ TS are sorted in decreasing order of values and those that have the K' highest values are identified as being the ones most likely to have binary values of 1 in each of the T TSs.

As shown in Algorithm 1, each of the $i_{LDPC-CS}$ outer iterations between the LDPC and CS decoder comprises i_{LDPC} inner iterations and i_{CS} inner iterations for the LDPC and CS decoder, respectively. Algorithm 1 relies on the signal combination matrices $[\mathbf{Q}_1, \cdots, \mathbf{Q}_M]$, which will be detailed in Section III. In the initialization of the LDPC decoder during the outer iterations, the *a priori* LLRs $\tilde{\mathbf{Y}}_a$ provided by the CS decoder are set to zero, before the probability knowledge $\boldsymbol{l} = [l_1, l_2, \cdots, l_M]$ is added. Furthermore, the *a priori* LLRs $\tilde{\mathbf{M}}_a$ provided by the CND for the VNDI are reset to zero during the initialization of the LDPC decoder during each outer iteration. Similarly, the *a priori* LLRs $\tilde{\mathbf{\Gamma}}_a$ forwarded by the VNDII to the SND are reset to zero during the initialization of the CS decoder during each outer iteration.

Here, iterative joint LDPC-CS decoding employs SISO decoding for the iterative exchange of *extrinsic* LLRs. By contrast, the SSCC benchmarker employs hard information decoding components and separate LDPC-CS decoding. In this benchmarker, the m^{th} LDPC decoder converts the channel LLRs $\tilde{\boldsymbol{a}}_m$ into hard information bits, which are forwarded to the SNs of the CS decoder. The SNs then provide votes for their connected VNs in VNDII, where the votes express this opinion concerning the value gleaned from the connected VNs in VNDI and may be used for determining whether the corresponding signals have values of 1 in each of the T TSs. More specifically, a SN forwards a vote of 1 to its connected VNs in VNDII, if it judges that the corresponding signal should adopt the binary value 1, otherwise a value of 0. Then, each VN in VNDII selects a value of 1 for its corresponding signal, if all of the connected SNs vote for this value. To elaborate further, each VN in VNDII counts the number of the votes for a binary value of 1 provided by its connected sensors. If the count is equal to the degree of the VN in VNDII, then it considers the corresponding signal to have a value of 1. By contrast, in the SSCC hard information benchmarker, there is no iterative information exchange either within the CS decoder or between the CS and LDPC decoder. Hence, the complexity of CS decoding using the SSCC hard information approach is quantified by

$$\mathcal{C}^{hard}_{CS} = N(M-1)T. \tag{1}$$

The output of the decoder is a vector of estimated values $\hat{\boldsymbol{x}}_n = [\hat{x}_{n,1}, \cdots, \hat{x}_{n,T}]$ across each signal $n \in [1, N]$ for all T TSs for both the proposed SISO schemes and the SSCC hard information benchmarker.

Algorithm 1: Iterative joint LDPC-CS decoding algorithm

Input: $\mathbf{H}$, K', $\boldsymbol{d} = [d_1, d_2, \ldots, d_M]^T$, i_{LDPC}, i_{CS}, $i_{LDPC-CS}$, N, M, T, $\mathbf{\Phi}$, $\mathbf{Q}_1, \ldots, \mathbf{Q}_M$, $\boldsymbol{l} = [l_1, \ldots, l_M]^T$, channel LLRs: $\tilde{\boldsymbol{a}}_1, \ldots, \tilde{\boldsymbol{a}}_M$
Output: Estimated binary signals $\hat{\boldsymbol{x}}_1, \ldots, \hat{\boldsymbol{x}}_N$

```
for i = 1 to i_LDPC-CS do
    Initialization for LDPC decoding;
    VNDI update in the LDPC decoding;
    for ii = 1 to i_LDPC do
        CND update in the LDPC decoding;
        VNDI update in the LDPC decoding;
    end
    Initialization for CS decoding;
    SND update in the CS decoding;
    for jj = 1 to i_CS do
        VNDII update in the CS decoding;
        SND update in the CS decoding;
    end
end
Perform final decision to obtain the estimated binary signal x̂_1, ..., x̂_N;
```

III. The Proposed SD Approach and HD Approach

In this section, the e-MAP technique of the SN update in the SND of the CS decoder is introduced in Section III-A, while the proposed SD and HD solutions are discussed in Section III-B.

A. e-MAP approach

In the e-MAP approach, each SN in the SND converts the *a priori* LLRs $\tilde{\gamma}^a$ and $\tilde{z}^a$ into the *extrinsic* $\tilde{\gamma}^e$ LLRs by considering all possible combinations of signal values for all connected signals. Let us illustrate this using an example based on the SN S_2 in Fig. 2 during the t^{th} TS. As seen in Fig. 2, S_2 is connected to the second, third, fourth and fifth VNs II, and it also connects to the t^{th} VN I in the second layer of the LDPC decoder. The input LLRs are inferred from VII_2 ($\tilde{\gamma}^a_{2\to 2}$), VII_3 ($\tilde{\gamma}^a_{3\to 2}$), VII_4 ($\tilde{\gamma}^a_{4\to 2}$), VII_5 ($\tilde{\gamma}^a_{5\to 2}$), corresponding to the signals, and VI_2 ($\tilde{z}^a_{t\to 2}$) is provided by the LDPC decoder. The target of the SN S_2 is to calculate the *extrinsic* LLRs ($\tilde{z}^e_{2\to t}$, $\tilde{\gamma}^e_{2\to 2}$, $\tilde{\gamma}^e_{2\to 3}$, $\tilde{\gamma}^e_{2\to 4}$ and $\tilde{\gamma}^e_{2\to 5}$) for the connected VN I and VNs II.

In this example, we assume a sparsity upper bound of $K' = 2$,

so that $K'=2$ of the N signals may have the binary value of 1 in each of the T TSs, with all other signals adopting the value 0 in this particular TS. In this example, we assume that the true sparsity $K=K'$, as well as signals corresponding to the VNs V_3 and V_5, have values of 1. The possible combinations of connected signal values can be listed based on the prior knowledge of the node connectivity $\mathbf{\Phi}$ and the sparsity upper bound K' in the FC. For S_2 of Fig. 2, the matrix of possible combinations can be expressed as

$$\mathbf{Q}_2^{e-MAP} = \begin{matrix} VI_2 \\ VII_2 \\ VII_3 \\ VII_4 \\ VII_5 \end{matrix} \begin{matrix} \begin{matrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{matrix} \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \end{pmatrix} \end{matrix}. \tag{2}$$

Each column in the brackets represents a possible combination of the associated signal values, where the first bit of each combination represents the predicted value of VII_2 in this particular combination, while the second, third and fourth bits correspond to VII_3, VII_4 and VII_5, respectively in this example. The values at the top-line outside the brackets are stored in a row vector denoted by $\boldsymbol{g}_2$, and they correspond to the specific values of VI_2, which are obtained by performing the OR function that describes the operation of the SNs on the corresponding combinations. The first column represents the particular combination, where all four connected VNs II have zero values, and hence the predicted value from VI_2 is 0. However, the combination represented by the second column predicts that VII_5 has a value of 1 and VII_2, VII_3 and VII_4 all have zero values, in which case VI_2 has 1. Because the sparsity upper bound K' is prior knowledge in the FC, no combination has a number of 1 values higher than K'. In this example, the true combination is $[0\ 1\ 0\ 1]^T$, and the true value of VI_2 is 1. Since the connectivity $\mathbf{\Phi}$ of the factor graph does not vary across the T TSs, the combinations matrix $\mathbf{Q}$ corresponding to each SN does not change in this period, but the true combination of each SN may be expected to change in different TSs. While this example is fairly trivial, the number of possible combinations can be large, where the number of possible combinations for the m^{th} SN S_m may be quantified as

$$P_m^{e-MAP} = \sum_{k=0}^{\min\{K,d_m\}} \binom{d_m}{k}. \tag{3}$$

According to its combination matrix $\mathbf{Q}_m$, the m^{th} SN preforms the decoding. First of all, the probabilities in the logarithmic domain of for the set of P_m^{e-MAP} combination are calculated as:

$$\boldsymbol{\eta}_m = \sum_{i=1}^{d_m} \left(|\mathbf{Q}_m^{e-MAP}(i,:) - \mathbf{1}| \odot \tilde{\boldsymbol{\gamma}}_m^a(i) \right) + |\boldsymbol{g}_m - \mathbf{1}| \odot \tilde{z}_{m\to t}^a, \tag{4}$$

where the $\boldsymbol{\eta}_m$ is a probability vector of length of P_m^{e-MAP}, while the $\mathbf{1}$ vectors have the length of P_m^{e-MAP}, with all elements set to 1. The notation $|\cdot|$ represents the element-wise absolute values of a vector. Here, $\tilde{\boldsymbol{\gamma}}_m^a$ is a column vector that contains all *a priori* LLRs provided by the VNs II connected to the SN S_m, while $\tilde{z}_{m\to t}^a = \tilde{y}_{m\to t}^e$ is the *a priori* LLR provided by the VN I connected in the t^{th} TS, and $\odot$ is the dot product.

The probabilities of the P_m^{e-MAP} combinations are then exploited for calculating the *extrinsic* LLRs $\tilde{\boldsymbol{\gamma}}^e$ of the SN S_m. More specifically, the *extrinsic* LLR forwarded from the S_m to its k^{th} ($k \in [1, d_m]$) connected VN II VII_n are calculated as

$$\begin{aligned} \tilde{\gamma}_{m\to n}^e = &\max\left[\boldsymbol{\eta}_m(\mathbf{Q}_m^{e-MAP}(k,:)=0)\right] \\ &- \max\left[\boldsymbol{\eta}_m(\mathbf{Q}_m^{e-MAP}(k,:)=1)\right] - \tilde{\gamma}_{n\to m}^a, \end{aligned} \tag{5}$$

where $\boldsymbol{\eta}_m(\mathbf{Q}_m^{e-MAP}(k,:)=0)$ represents the set of probabilities associated with the corresponding combinations having 0 values in the k^{th} row, and $\boldsymbol{\eta}_m(\mathbf{Q}_m^{e-MAP}(k,:)=1)$ is the set of probabilities associated with the corresponding combinations having 1 values in the k^{th} row. Here, $\tilde{\gamma}_{n\to m}^a$ is the *a priori* LLR provided by the n^{th} VN II for the SN S_m in the t^{th} TS. Similarly, the *extrinsic* LLR provided by S_m for its connected VN I in the t^{th} TS is calculated as

$$\tilde{z}_{t\to m}^e = \eta_m(g_m=0) - \max\{\boldsymbol{\eta}_m(\boldsymbol{g}_m=1)\} - \tilde{z}_{m\to t}^a, \tag{6}$$

where $\tilde{z}_{m\to t}^a$ is the *a priori* LLR provided for S_m by its connected VN I in the t^{th} TS.

The complexity of each SN S_m employing the e-MAP approach is quantified in terms of the number of addition and max calculations formulated as

$$\mathcal{C}_{S_m}^{e-MAP} = 2d_m \times P_m^{e-MAP} - \sum_{k=0}^{\min\{K,d_m\}} k\binom{d_m}{k}. \tag{7}$$

B. The proposed approaches

As shown by the binomial coefficients in Eq. (3) and (7) above, the number of possible combinations escalates rapidly with the sparsity upper bound K' and the degree d_m of the m^{th} sensor, which leads to an excessive the complexity for the e-MAP approach. Hence, this section introduces the proposed SD and HD techniques, which aim for pruning the combination matrix $\mathbf{Q}$ of the e-MAP approach in order to reduce the associated complexity. This is achieved by generating a combination matrix, which contains only the more likely combinations and abandons the less likely ones. More specifically, the pruned combination matrix contains the single most likely combination, which is identified by using a tree search, as well as several alternative hypotheses having high likelihoods, which are necessary for obtaining high-quality *extrinsic* LLRs $\tilde{\mathbf{\Gamma}}^e$ and $\tilde{\mathbf{Z}}^e$. The rest of this section is organised as follows. Section III-B.1 introduces a first step that is common for both the proposed SD and HD approaches, which identifies the most likely combination using a tree search. Section III-B.2 then elaborates further on the SD approach, while Section III-B.3 relies on Section III-B.1 to elaborate further on the HD approach.

1) The tree search: Algorithm 2 shows the flow of search for the most likely combination using a tree search. Fig. 3 illustrates a specific example of tree search in a particular TS based on the example of S_2 in Fig. 2, which has a degree of 4 and it is connected to the 2^{nd}, 3^{rd}, 4^{th} and 5^{th} VN II of Fig. 2, where the sparsity upper bound is $K'=2$. The proposed method of searching for the most likely combination is based only on the *a priori* LLRs arriving from VNs II. The

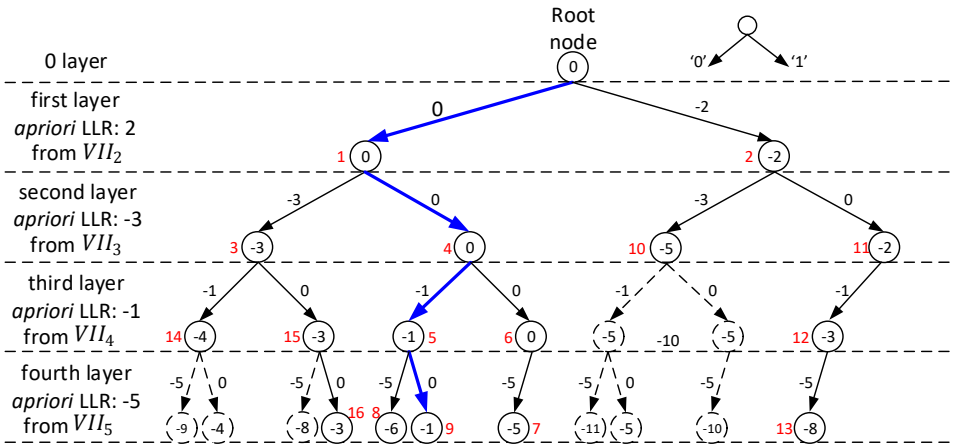


Fig. 3: An example of the proposed SD approach based on S_2 in Fig. 2, where $[0\ 1\ 0\ 1]^T$ is the most likely combination. Here transitions to left imply '0' values, while transitions to right imply '1' values. The values alongside each edge in the tree are the metrics of the paths, while the values in the circles are the metrics of each node, and the numbers next to the circles are the order that each node is traversed in. The emphasised paths correspond to the most likely combination, while the dashed lines represent the nodes and paths that are not traversed through at all.

a priori LLR $\tilde{z}^a$ provided by the VN I is not considered in the tree search, because it only has to be used when considering the all-zero combination during the calculation of the *extrinsic* LLRs $\tilde{\boldsymbol{z}}^e$. Apart from the all-zero combination, all other combinations have binary values 1 for the values $\boldsymbol{g}_m$ predicted from the LDPC decoder. Since the alternative hypothesis of the bits $\boldsymbol{g}_m$ is required, hence the all-zero combination may be considered as a special case and always be included in the pruned combination matrix, as will be detailed in Sections III-B.2 and III-B.3. Owing to this, the bit corresponding to the LDPC decoding does not have to be considered in the tree search and may be removed in order to reduce the associated complexity.

To elaborate, let us consider a particular example where the *a priori* LLRs $\tilde{\gamma}^a$ of S_2 provided by VII_2, VII_3, VII_4 and VII_5 are 2, -3, -1 and -5, respectively. Accordingly, there are four layers in the tree corresponding to the four connected VNs II, as shown in Fig. 3. The tree search starts from the root node in layer 0 with a node metric equal to 0. The path metrics associated with the edges in the tree are dependent on the *a priori* LLRs of the corresponding layer in Fig. 3. In cases where an *a priori* LLR is positive, the path metrics associated with branches to the left in the corresponding layer are set to 0, and the right hand side path metrics are set to the negative of the *a priori* LLRs. By contrast, if the *a priori* LLR is negative, the left path metrics in that layer are set equal to that particular *a priori* LLR, and the right path metrics are set to 0. The *a priori* LLR associated with the first layer is provided by VII_2 in Fig. 3 and has a value of 2. Hence, the left path metric from the root node down to the left child node in the first layer adopts the value of $p_1^l = 0$, while the right path metric adopts the value of $p_1^r = -2$. as shown in Fig. 3. In the first layer of the example, the metrics of the left child node and right child node are $w_1^l = 0$ and $w_2^r = -2$, respectively, which are defined by adding the path metrics of the respective branches to the node metric of the root node. Upon generating two new child nodes, the child node having a more positive node metric will be set as the next parent node, and all other child nodes will be saved in the stack for later traversal. In this example, the left child node in the first layer is set as parent node in the next step, and the right node is saved in the stack. Then, the tree search of Fig. 3 progresses to the second layer, whose *a priori* LLR from VII_3 is -3. In the second layer, the left path metric is set as $p_2^l = -3$ and the right path adopts $p_2^r = 0$ accordingly. Hereafter, the metrics of child node 3 and child node 4 are calculated as $w_3^l = w_1^l + p_2^l = -3$ and $w_4^r = w_1^l + p_2^r = 0$.

Following the traversal rules, the tree search of Fig. 3 extends down to the last layer, whereupon the parent node is node 6 with metric 0 and node 2, 3 and 5 are in the stack. The final *a priori* LLR from VII_5 has a value of -5, which implies a binary value of 1 and a traversal to the right. However, the transition to the right node in this case is not allowed, because the number of right traversals would exceed the sparsity upper bound of $K' = 2$ in this particular example. Hence, the only option is to traverse left to node 7 in the last layer, which corresponds to the combination $[0\ 1\ 1\ 0]^T$ and has the metric of -5. However, the first combination reached is not always the most likely combination, and hence the traversal should be continued further.

Whenever the traversal reaches the last layer, the new parent node for the next step is selected as the entry in the stack with the most positive metric, before removing the new parent node from the stack and extending its child nodes. In cases where more than one nodes have the highest node metric in the stack, the node having the smallest index may be selected as the new parent node. Following this, the traversal continues from the layer where the new parent node is to the last layer. In order to limit the complexity of the tree search when the number of layers is large, a limit ξ may be imposed on the number of combinations that will be identified before terminating the search for the most likely combination. In this simple example, we set the threshold to $\xi = 3$, and the combination having the most positive metric is selected as the

Algorithm 2: Tree search for searching the most likely combination

Input: d_m, K', *a priori* LLRs from the connected VNs II $\tilde{\gamma}_1^a, \ldots, \tilde{\gamma}_{d_m}^a$, threshold ξ
Output: the most likely combination

```
Stack: S = {root node}
for i = 1 to d_m do
    if γ̃_i^a > 0 then
        p_i^l = 0;
        p_i^r = −γ̃_i^a
    else
        p_i^l = γ̃_i^a;
        p_i^r = 0;
    end
end
while the number of combinations < ξ do
    Parent node: the node with the most positive
      metric w^p in the stack S.
    Delete the parent node in the stack S.
    L_p: the layer that the parent node is in.
    for j = L_p + 1 to d_m do
        if the number of traversed right paths is smaller
          than K' then
            Left child node metric = w^p + p_j^l;
            Right child node metric = w^p + p_j^r;
            if j ≠ d_m then
                Set the node having more positive
                  metric as the new parent and save the
                  other one in the stack S.
            else
                Obtain two combinations;
            end
        else if the number of traversed right paths is
          larger than or equal to K' then
            Left child node metric = w^p + p_j^l;
            if j ≠ d_m then
                Always set the left child node as the
                  new child node;
            else
                Obtain one combination;
            end
    end
    Store the combinations in the combination matrix.
end
Choose the combination with most positive metric as
  the most likely combination.
```

most likely combination, when the limit of 3 combinations is reached. Therefore, the traversal continues until we reach the nodes 8 and 9 of Fig. 3, which have the metrics of -6 and -1, respectively. In this example, the combination $[0\ 1\ 0\ 1]^T$ corresponds to node 9 with the highest node metric -1 may be set as the most likely combination. Alternatively, the search may be terminated, when there are no remaining nodes in the stack having a node metric higher than that of the node in the final layer corresponding to the most likely combination found so far.

2) Proposed SD approach: In the proposed SD approach, the tree search traversal will continue beyond finding the most likely combination until alternative hypotheses have been obtained for each layer. In the example of Fig. 3, on the way to reaching the most likely combination $[0\ 1\ 0\ 1]^T$, we also obtained two other combinations $[0\ 1\ 1\ 0]^T$ and $[0\ 1\ 0\ 0]^T$, corresponding to node 7 and node 8, respectively. At this point, node 2, and node 3 of Fig. 3 are in the stack available for further traversal. The combinations $[0\ 1\ 1\ 0]^T$ and $[0\ 1\ 0\ 0]^T$ represent alternative hypotheses for the third and fourth bits of the most likely combination $[0\ 1\ 0\ 1]^T$, but alternative hypotheses are missing for the first and second bits. In cases like this where we do not have alternative hypotheses for all bits, the SD approach continues to traverse, evolving from the node in the stack having the most positive metric, until all alternative hypotheses are obtained. As mentioned in Section III-B.1, the all-zero combination may be treated as a special case. If the all-zero combination is not present in the combination matrix after the tree search, then it may be added to the last column of the combination matrix. After extending the tree search until alternative hypotheses are obtained for all bits in the example above, we obtain the following combination matrix for S_2 using the proposed SD approach.

$$\mathbf{Q}_2^{SD} = \begin{array}{c} \\ VI_2 \\ VII_2 \\ VII_3 \\ VII_4 \\ VII_5 \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & 1 & 1 & 1 & 1 & 0 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{pmatrix} \end{array}, \tag{8}$$

In order to constrain the complexity of the proposed joint iterative LDPC-CS scheme, the tree search seeking alternative hypotheses may be abandoned after the number of combinations considered has reached a limit φ. For example, we may consider the case where an alternative hypothesis that provides a binary value of 1 for the first bit is still missing after obtaining $\varphi = 100$ combinations. In this case, the left path of the first layer and the child nodes beneath that path of Fig. 3 will not be visited, and only the right path of the first layer and the child nodes beneath it will be visited. This helps us obtain at least one alternative hypothesis for each bit as quickly as possible.

The proposed SD approach significantly reduces the decoding complexity compared to the e-MAP approach by reducing the number of combinations that are used for calculating the *extrinsic* LLRs. After we obtain the combination matrix using the proposed SD approach, the combination matrix in Eq. (4) and Eq. (5) may be replaced by $\mathbf{Q}_m^{SD}$, while $\boldsymbol{g}_m$ in Eq. (6) may be replaced by the specific $\boldsymbol{g}_m$ corresponding to the $\mathbf{Q}_m^{SD}$.

The complexity of each SN S_m using the proposed SD approach in the CS decoder may be quantified in terms of by the number of additions and $\max$ computations during the *extrinsic* LLR calculation. Additionally, the complexity of tree search should be included in the proposed SD approach. In contrast to the e-MAP approach, where the combination matrix of S_m does not vary across the T TSs, may vary from TS to TS for the proposed SD approach, depending on the values of the input *a priori* LLRs. The complexity associated with each visit to a SN S_m using the proposed SD approach may be calculated as

$$\mathcal{C}_{S_m}^{SD} = 2d_m \times P_m^{SD} - \sum(\mathbf{Q}_m^{SD}) + num_{nodes} - 2, \tag{9}$$

where $\sum \mathbf{Q}_m^{SD}$ is the total number of elements having a value of 1 among all rows and columns of $\mathbf{Q}_m^{SD}$, which corresponds

to the second term of Eq. (7), $\sum_{k=0}^{\min\{K,d_m\}} k\binom{d_m}{k}$. Here, P_m^{SD} is the number of combinations obtained by the SD approach, which depends on the input LLRs for S_m. Meanwhile, num_{nodes} represents the number of nodes traversed during the tree search. Note that the calculation complexity of SN decoding in the SD approach is mainly influenced by the number of combinations used for the *extrinsic* LLR, since the complexity of the tree search is low compared to the savings attained by pruning the less likely combinations.

3) Proposed HD approach: In contrast to the tree search of the proposed SD approach, our HD technique considers the Hamming distance between combinations for selecting alternative hypotheses. More specifically, the HD approach populates the combination matrix with the most likely combination obtained by the tree search and all other combinations that are within a specified HD of that most likely combination. Returning to the example of Section III-B.1, the most likely combination is $[0\ 1\ 0\ 1]^T$, and the set of all valid combinations within a HD of say 2 are specified in the following combination matrix of S_2

$$\mathbf{Q}_2^{HD} = \begin{matrix} VI_2 \\ VII_2 \\ VII_3 \\ VII_4 \\ VII_5 \end{matrix} \begin{matrix} \begin{matrix} 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \end{matrix} \\ \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \end{pmatrix} \end{matrix}, \tag{10}$$

Having said that we still exclude all combinations that are within the specified HD of the most likely combination, but have more than K' binary value of 1.

The combination matrix $\mathbf{Q}_m^{HD}$ may be used for performing decoding processing of the SN by replacing the combination matrix $\mathbf{Q}_m^{e-MAP}$ in Eq. (4) and Eq. (5), while $\boldsymbol{g}_m$ in Eq. (6) is replaced by the specific $\boldsymbol{g}_m$ corresponding to the combination matrix $\mathbf{Q}_m^{HD}$. As in the SD approach, if the combination matrix does not include the all-zero combination, it may be added to the last column. The number of combinations obtained for the SN S_m by employing the proposed HD approach may be formulated as:

$$P_m^{HD} = \sum_{k=0}^{\min\{HD,K,d_m\}} \binom{d_m}{k}. \tag{11}$$

The complexity of a SN S_m employing the proposed HD approach in the CS decoder may be quantified by the number of additions and max functions evaluations according to

$$\mathcal{C}_{S_m}^{HD} = 2d_m \times P_m^{HD} - \sum(\mathbf{Q}_m^{HD}) + num_{most} - 2, \tag{12}$$

where $\sum(\mathbf{Q}_m^{HD})$ is the sum of all elements in $\mathbf{Q}_m^{HD}$, and num_{most} is the number of nodes that are traversed through, while searching for the most likely combination. As mentioned in Section III-B, the complexity of the decoding in the SNs is mainly influenced by the number of combinations. When the specified HD is small, the number of combinations considered by the HD approach can be significantly reduced compared to the e-MAP approach, yielding a very significant decoding complexity reduction.

IV. EXIT Chart Analysis

In Section IV-A, EXIT functions are introduced for the case of non-equiprobable bits. The LDPC EXIT function and CS EXIT function are detailed in Sections IV-B, Section IV-C, respectively. Following this, Section IV-D portrays our trajectory plots for characterizing the iterative exchange of Mutual Information (MI) between the LDPC and CS decoders.

A. EXIT function for non-equiprobable bit sequence

A constituent decoder in an iterative decoding process may be characterized by its EXIT functions [34]. Typically, a pair of constituent decoders engage in an iterative exchange of *extrinsic* information during iterative decoding, which may be characterized by a decoding trajectory that represents the MI evolution of the iterative exchange of *extrinsic* LLRs. The proposed iterative joint LDPC-CS decoding scheme is divided into LDPC decoding and CS decoding. In our scheme, the observations in the sensors $(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_M)$ may be sparse, according to the sparse signals $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)$ and sparse sensing matrix $\boldsymbol{\Phi}$. Owing to this, the *extrinsic* LLRs that are exchanged between the LDPC and CS decoder in Fig. 2 pertain to non-equiprobable bits. As a results of this, the EXIT chart analysis of the LDPC decoding and CS decoding is different from conventional EXIT chart analysis, where the iteratively exchanged *extrinsic* LLRs pertain to equiprobable bit values. In an EXIT chart, the quality of the exchanged *a priori* and *extrinsic* LLRs is quantified by the MI between the LLR sequence and its corresponding bit sequence [35]. In our software used for generating the EXIT functions, the *a priori* LLRs are generated artificially based on their corresponding bit sequence, in order to characterize each decoder individually. In this section, we adopt the notation $\boldsymbol{\omega}$ to represent a bit sequence having non-equiprobable bit values, and the notation $\tilde{\boldsymbol{\omega}}$ to represent its corresponding LLR sequence. Meanwhile, the quality of the LLR sequence $\tilde{\boldsymbol{\omega}}$ is quantified by the MI $I(\boldsymbol{\omega}; \tilde{\boldsymbol{\omega}})$. The probability that each bit of the bit sequence $\boldsymbol{\omega}$ adopts the binary value 0 is P_ω. We may artificially generate each LLR of $\tilde{\boldsymbol{\omega}}$ using

$$\tilde{\omega}_i = \frac{\sigma_\omega^2}{2}(1 - 2\omega_i) + N_\omega + \log\left(\frac{P_\omega}{1 - P_\omega}\right), \tag{13}$$

where ω_i is a bit of the bit sequence $\boldsymbol{\omega}$. Here, N_ω is an independent Gaussian random variable with zero mean and variance σ_ω^2 [34], [36].

The MI $I(\boldsymbol{\omega}; \tilde{\boldsymbol{\omega}})$ between the sequence of non-equiprobable bits $\boldsymbol{\omega}$ and the LLRs $\tilde{\boldsymbol{\omega}}$ may be calculated using the so-called histogram-based method of [37]

$$I(\boldsymbol{\omega}; \tilde{\boldsymbol{\omega}}) = \sum_{\omega_i=0,1} \int_{-\infty}^{+\infty} f_{\tilde{\boldsymbol{\omega}}|\boldsymbol{\omega}}(\tilde{\omega}_i|\omega_i) f_{\boldsymbol{\omega}}(\omega_i) \log_2\left(\frac{f_{\tilde{\boldsymbol{\omega}}|\boldsymbol{\omega}}(\tilde{\omega}_i|\omega_i)}{f_{\tilde{\boldsymbol{\omega}}}(\tilde{\omega}_i)}\right) d\tilde{\omega}_i. \tag{14}$$

B. LDPC EXIT function

Fig. 4 shows a block diagram illustrating our software used for characterizing an LDPC EXIT function. As discussed in Section II-B, M number of LDPC decoders are operated in parallel, and each layer works in parallel across the M layers of the factor graph of Fig. 2. Hence, Fig. 4 illustrates the generation of the LDPC EXIT function for one of these layers having an index m. Once the EXIT function of all M layers has been obtained, an overall LDPC EXIT function may be

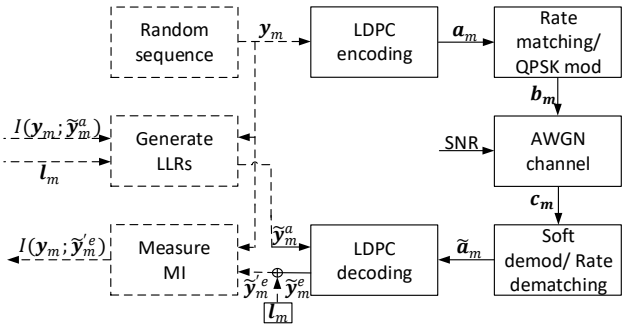


Fig. 4: Adapted schematic used to depict the generation of the EXIT function for LDPC decoding, where solid lines are used for components of the real system and dash lines are used for components that are particular to the EXIT charts software components.

obtained by taking an average over the M layers.

As an inner decoding process, LDPC decoding has two input *a priori* LLR sequences, namely $\tilde{\boldsymbol{a}}_m$ provided by the channel and $\tilde{\boldsymbol{y}}^a_m$ by the CS decoder, respectively. The channel LLRs $\tilde{\boldsymbol{a}}_m$ may be generated as discussed in Section II-A and shown in Fig. 4, wherein a random bit sequence $\boldsymbol{y}_m$ is processed by LDPC encoding, rate matching, QPSK modulation, before transmission over an AWGN channel. At the receiver, QPSK soft demodulation and rate dematching are used for obtaining the channel LLRs $\tilde{\boldsymbol{a}}_m$. By contrast, the other *a priori* LLRs $\tilde{\boldsymbol{y}}^a_m$ provided by CS decoding may be generated artificially according to Eq. (13), allowing the EXIT function of LDPC decoding to be characterized independently from the EXIT function of CS decoding.

As discussed in Section II-B, the processing of LDPC decoding comprises initialization, CN update and VN update. Upon completion of these processes, the *extrinsic* LLRs $\tilde{\boldsymbol{y}}^e_m$ are output by LDPC decoding, and each element in the vector $\tilde{\boldsymbol{y}}^e_m$ is added with an LLR quantifying the probability of the corresponding bit in $\boldsymbol{y}_m$ adopting a value of 0. More specifically, these LLRs are given by $\boldsymbol{l}_m = \left[\log\left(\frac{P_m}{1-P_m}\right),\ldots,\log\left(\frac{P_m}{1-P_m}\right)\right] \in \mathbb{R}^{1\times T}$, and the resultant superimposed *extrinsic* LLRs $\tilde{\boldsymbol{y}}'^e_m$ may be used for quantifying the MI using Eq. (14).

Fig. 5 characterizes the iterative exchange of *extrinsic* LLRs $\tilde{\boldsymbol{y}}'^e_m$ and $\tilde{\boldsymbol{z}}'^e_t$ between the LDPC and CS decoder, where the corresponding bit sequences of $\boldsymbol{y}_m$ and $\boldsymbol{z}_t$ have non-equiprobable binary values. Here, we adopt $N = 500$ signals, $M = 150$ sensors, a degree for each signal of $K_s = 5$, true sparsity of $K = K' = 5$, coding rate of $R = \frac{1}{3}$ and $T = 500$ TSs, using the 3GPP NR BG 2 to generate the parity check matrix used by LDPC coding. Unless otherwise specified, these fundamental parameters are adopted for all simulation results throughout this paper, including our EXIT charts, complexity analysis, and BLock Error Rate (BLER) plots. The axes of the EXIT chart represent the average MI $I(\boldsymbol{y};\tilde{\boldsymbol{y}}'^e)$ of the *extrinsic* LLRs $\tilde{\boldsymbol{y}}'^e$ and $\tilde{\boldsymbol{z}}'^e$, each of which ranges from 0 to the average entropy $\bar{H}$ of all M bit sequences $[\boldsymbol{y}_1;\ldots,\boldsymbol{y}_m;\ldots;\boldsymbol{y}_M]$. Here, an MI of $I(\boldsymbol{y};\tilde{\boldsymbol{y}}'^e) = 0$ indicates that there is no information in the LLR sequences $\tilde{\boldsymbol{y}}'^e$ about the corresponding bit sequences $\boldsymbol{y}$, while an MI equal to the

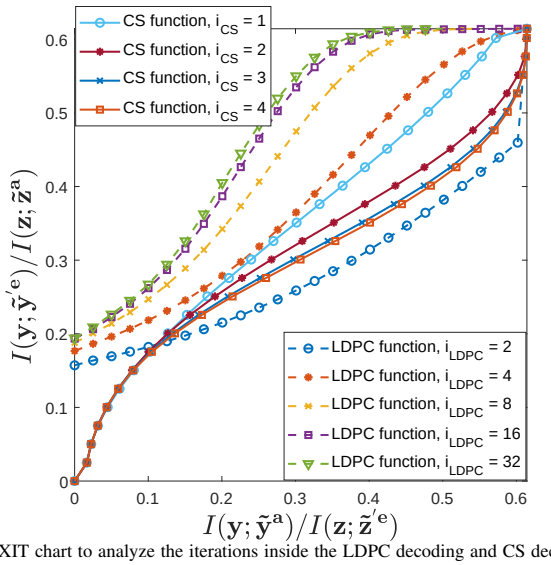


Fig. 5: EXIT chart to analyze the iterations inside the LDPC decoding and CS decoding for the case of $N = 500$ signals, $M = 150$ sensors, a degree for each signal of $K_s = 5$, true sparsity $K = K' = 5$, LDPC BG of 2, coding rate $R = \frac{1}{3}$, $T = 500$ TSs, SNR = -4.4 dB, threshold of finding the most likely combination $\xi = 10$ and threshold of finding alternative hypotheses $\varphi = 100$, using the e-MAP approach for CS decoding with QPSK modulation for communication over an AWGN channel.

average entropy $\bar{H}$ indicates that the LLR sequences $\tilde{\boldsymbol{y}}'^e$ have perfect information about the corresponding bit sequences $\boldsymbol{y}$. Recall that, the M non-equiprobable bit sequences in the proposed system are provided by the M different sensors, where each potentially has a different entropy $H(y_{m,t})$ owing to their different degrees $d_m, m \in [1, M]$. Owing to this, the bits in the different bit sequences $\boldsymbol{y}_m$ have different probabilities of adopting a value of 0. In order to characterize the system as a whole, the EXIT chart shows the mean over all entropies $H(y_{m,t})$, where $m = 1,\ldots,M$ and the final average MI $\bar{H}$ is the average of MI values that correspond to M layers in the LDPC decoder. Here, the entropy of each bit in the bit sequence $\boldsymbol{y}_m$ with probability P_m of adopting a value of 0 is calculated as

$$H(y_{m,t}) = -P_m \times \log_2(P_m) - (1-P_m) \times \log_2(1-P_m). \quad (15)$$

The LDPC EXIT functions of Fig. 5 show that when the LDPC decoder is presented with *a priori* LLRs $\tilde{\boldsymbol{y}}^a_m$ having an MI of 0, it responds by providing *extrinsic* LLRs $\tilde{\boldsymbol{y}}'^e_m$ having a non-zero MI. This may be explained by the information provided by the channel LLRs $\tilde{\boldsymbol{a}}_m$, which contain information about the content of the bit sequences $\boldsymbol{y}_m$. Furthermore, when the *a priori* LLRs $\tilde{\boldsymbol{y}}^a_m$ provided by the CS decoder contain perfect information, the LDPC decoder responds by providing *extrinsic* LLRs having perfect information, hence allowing the LDPC decoder to converge to the top right-hand corner of the EXIT chart. Fig. 5 shows that increasing the number of iterations i_{LDPC} performed within the LDPC decoder, moves the LDPC EXIT functions upwards. More specifically, as the number of iteration is increased, the value of the *extrinsic* MI $I(\boldsymbol{y}_m;\tilde{\boldsymbol{y}}'^e_m)$ increases for a given value of the *a priori* MI $I(\boldsymbol{y}_m;\tilde{\boldsymbol{y}}^a_m)$, leading to improved decoding performance. However, when the number of iterations i_{LDPC} performed by the LDPC decoder is higher than 16, Fig. 5 shows that only limited further gain may be obtained for channel SNRs above -4.4 dB. The capacity of the LDPC-CS scheme is expressed as the discrete-input continuous-output memoryless channel capacity when the transmission is over a QPSK modulated

AWGN channel. The capacity bound is given by the specific SNR, when the channel capacity equals the throughput of the LDPC-CS scheme. Furthermore, the throughput is calculated as $\eta = H(x_{n,t})\frac{N}{M}R\log_2\delta$, where $H(x_{n,t})$ is the entropy of each signal in the $\boldsymbol{x}_n$, and δ is the modulation order. In our scheme, when $N = 500$, $M = 150$, $K = K' = 5$ and $R = 1/3$ using QPSK modulation, the capacity bound is about -8.75 dB. Owing to this, we recommend striking an attractive performance vs. complexity trade-off by setting the number of iterations performed in the LDPC decoder to $i_{LDPC} = 16$ throughout this paper.

C. CS EXIT function

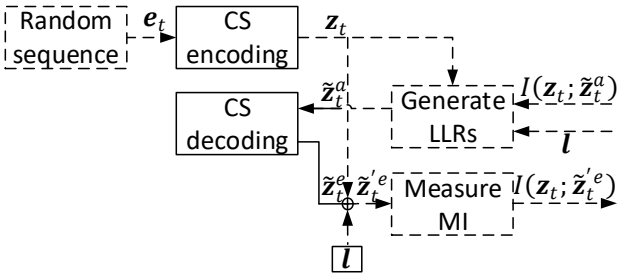


Fig. 6: Adapted schematic used to depict the generation of EXIT function for CS decoding, solid lines are used for components, which appear in the iterative decoding system, while dashed lines are used for EXIT chart software components.

Fig. 6 illustrates the block diagram for characterizing CS EXIT functions. As discussed in Section II-B, there are T layers in the CS decoder, corresponding to the T TSs. Here, the block diagram of Fig. 6 corresponds to a single layer of CS decoding having an index t. More specifically, the randomly generated bit sequence $\boldsymbol{e}_t$ represents the t^{th} column of $\mathbf{X}$ and contains all N signal values in the t^{th} TS. As shown in Fig. 6, the bit sequence $\boldsymbol{e}_t$ is processed by CS encoding based on the factor graph of Fig. 1. Rather than being provided by the LDPC decoder, the *a priori* LLRs $\tilde{\boldsymbol{z}}_t^a$ are generated artificially using Eq. (13), which includes the prior knowledge of the probability that each bit in the compressed bit sequence $\boldsymbol{z}_t$ adopts the binary value 0. In contrast to LDPC decoding, where each bit of the sequence $\boldsymbol{y}_m$ has the same probability of adopting a binary value of 0, the probability of each bit in $\boldsymbol{z}_t$ adopting a binary value 0 is different. This may be attributed to the different degrees of the M sensors, which results in the different probabilities for the M bits obtained in the t^{th} TS. Hence, the prior knowledge of the bit value probabilities may be represented by the vector of LLRs $\boldsymbol{l} = [l_1, \cdots, l_M]$, which is included in the artificially generated *a priori* LLRs $\tilde{\boldsymbol{z}}_t^a$. The CS decoder accepts the inputs $\tilde{\boldsymbol{z}}_t^a$ and generates the *extrinsic* LLRs $\tilde{\boldsymbol{z}}_t^e$ as discussed in Section II-B and shown in Fig. 2. Following this, the *extrinsic* LLRs $\tilde{\boldsymbol{z}}_t^e$ of the CS decoder are added to the vector $\boldsymbol{l}$ representing the MI, and the resultant LLRs are measured using Eq. (14). The final MI of the system is obtained by averaging across all T MI values corresponding to T layers in the CS decoder.

Fig. 5 shows that the MI $I(\boldsymbol{z}_t;\tilde{\boldsymbol{z}}_t'^e)$ of the *extrinsic* LLRs $\tilde{\boldsymbol{z}}_t'^e$ and the MI $I(\boldsymbol{z}_t;\tilde{\boldsymbol{z}}_t^a)$ of the *a priori* LLRs $\tilde{\boldsymbol{z}}^a$ of the CS EXIT functions are plotted as the swapped axes relative to the LDPC EXIT functions. This swapping is adopted because the *extrinsic* LLRs $\tilde{\boldsymbol{z}}_t'^e$ produced by the LDPC decoder become the *a priori* LLRs $\tilde{\boldsymbol{z}}_t^a$ of the CS decoder, and vice versa. Hence, inverting the CS decoder EXIT functions represents the corresponding MIs on the same axis. In contrast to the LDPC EXIT functions, Fig. 5 shows that the CS EXIT functions start from the bottom left corner of the EXIT chart, where $I(\boldsymbol{z}_t;\tilde{\boldsymbol{z}}_t'^e)$ equals to 0, when $I(\boldsymbol{z}_t;\tilde{\boldsymbol{z}}_t^a)$ is equal to 0. This is because the CS decoder has a single source of information and when the *a priori* LLR vector $\tilde{\boldsymbol{z}}_t^a$ contains no information, then the CS decoder is unable to generate any information for the *extrinsic* LLR vector $\tilde{\boldsymbol{z}}_t'^e$. Fig. 5 shows that when the number of iterations i_{CS} performed within the CS decoder is increased, the CS EXIT functions move downwards. More specifically, the *extrinsic* MI $I(\boldsymbol{z}_t;\tilde{\boldsymbol{z}}_t'^e)$ increases with the number of iterations i_{CS} performed within the CS decoder for a given value of *a priori* MI $I(\boldsymbol{z}_t;\tilde{\boldsymbol{z}}_t^a)$, implying superior decoding performance. However, as shown in Fig. 5, there is diminishing returns beyond $i_{CS} = 3$ iterations, hence we adopt $i_{CS} = 3$ throughout the rest of this paper.

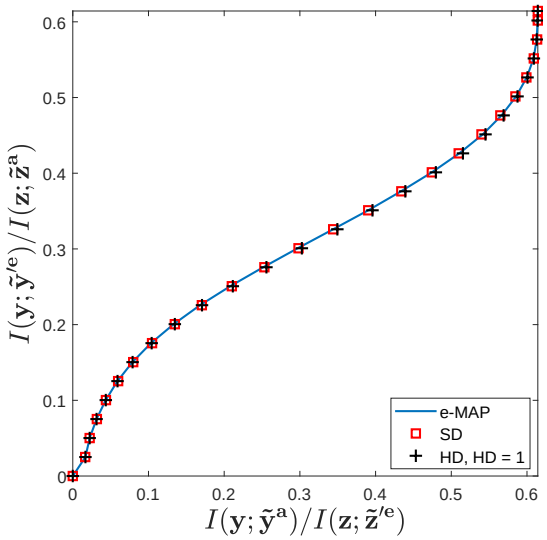


Fig. 7: CS functions for e-MAP, SD and HD approaches using $N = 500$ signals, $M = 150$ sensors, a degree for each signal of $K_s = 5$, true sparsity $K = K' = 5$, LDPC BG of 2, coding rate $R = \frac{1}{3}$, $T = 500$ TSs, threshold of finding most likely combination $\xi = 10$ and threshold of finding counter-hypotheses $\varphi = 100$, $i_{LDPC} = 16$, $i_{CS} = 3$, for the case of using QPSK modulation for communication over an AWGN channel.

Fig. 7 compares the CS EXIT functions for the proposed SD and HD approaches, and adopts the e-MAP approach as a benchmarker. The CS EXIT functions in Fig. 7 show that the proposed SD and HD approaches result in very similar CS EXIT functions as the e-MAP approach, implying that the CS decoder can obtain the same decoding performance by employing the proposed techniques, with the benefit of substantially reduced complexity.

D. Decoding trajectory

In an EXIT chart, a stair-case-shaped decoding trajectory may be used to illustrate the iterative exchange of *extrinsic*

information between a pair of decoders, namely the LDPC and CS decoders in the proposed scheme. This stair-case-shaped the decoding trajectory characterizes the MIs measured during the iterative exchange of *extrinsic* LLRs in the proposed scheme. The calculation of MI for the decoding trajectory is performed using the same Eq. (14) as used for drawing the EXIT functions.

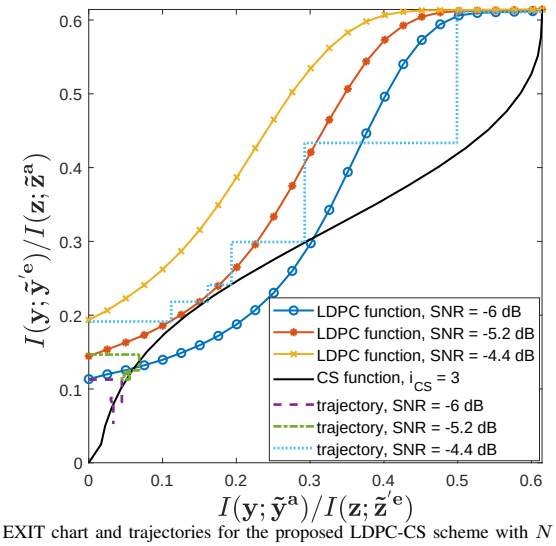


Fig. 8: EXIT chart and trajectories for the proposed LDPC-CS scheme with $N = 500$ signals, $M = 150$ sensors, a degree for each signal of $K_s = 5$, true sparsity $K = K' = 5$, LDPC BG of 2, coding rate $R = \frac{1}{3}$, $T = 500$ TSs, $i_{LDPC} = 16$, $i_{CS} = 3$, threshold of finding most likely combination $\xi = 10$ and threshold of finding counter-hypotheses $\varphi = 100$, when using QPSK modulation for communication over an AWGN channel having various SNRs.

Fig. 8 illustrates the LDPC and CS EXIT functions together with the corresponding decoding trajectories. Here, each horizontal transition in the staircase characterizes the improvement in MI obtained by the CS decoder, while each vertical transition corresponds to the LDPC decoder. As shown in Fig. 8, for the case of SNR of -6 dB, when the LDPC and CS EXIT functions intersect, the EXIT chart tunnel may be deemed closed. In this case, the iterative decoding trajectory becomes curtailed at the intersection point of the LDPC and CS EXIT functions, and will become unable to pass beyond it towards the top right-hand corner of the EXIT chart, where a low BLER is achieved. Hence, a closed EXIT tunnel implies having a high BLER. By contrast, an open EXIT tunnel allows the iterative decoding trajectory to evolve towards the top right-hand corner of the EXIT chart, as exemplified for the case SNR $= -4.4$ dB in Fig. 8. It may be observed that as the SNR is increased, the open EXIT chart tunnel becomes wider, and fewer steps are required to get through the tunnel, indicating that a low BLER can be achieved using fewer iterations. Increasing the number of iterations i_{LDPC} and i_{CS} performed within the LDPC and CS decoder has the effect of widening the EXIT chart tunnel, as described in Sections IV-B and IV-C.

Fig. 8 shows that at SNR $= -4.4$ dB, 6 steps/iterations are required for reaching the top right-hand corner of the EXIT chart, where a low BLER is attained. This implies that $i_{LDPC-CS} = 6$ iterations are recommended between the LDPC and CS decoder. Note that, there is a slight mismatch between the corner points of the decoding trajectory and the EXIT functions. This may be explained by the random independent generation of the LLRs in the *a priori* LLR vector $\tilde{y}_m^a$ and $\tilde{z}_t^a$ when drawing the LDPC and CS EXIT functions. However, when iterating between the LDPC and CS decoder, the factor graph of Fig. 2 has many short cycles of length 4 in the connectivity between the LDPC and CS decoder, which creates correlation between the LLRs, and this is not modeled by the EXIT charts analysis. However, Fig. 8 shows the general agreement between the trajectories and EXIT functions, which validates the proposed scheme.

V. Complexity Analysis

The complexity of each operation in each SN for each of the e-MAP, SD and HD approaches was quantified in Section III. Furthermore, the complexity calculation for each VN I and VN II may be quantified by characterizing the forwards and backwards propagation algorithm [38] as follows

$$\mathcal{C}_{VN_v} = 3(d_{V_v} - 1), \tag{16}$$

where d_{V_v} is the degree of the VN v.
Likewise, the complexity of each operation of a CN c may be quantified by [39]

$$\mathcal{C}_{CN_c} = 3(d_{C_c} - 2), \tag{17}$$

where d_{C_c} is the degree of the CN c.
The complexity of the proposed iterative joint LDPC-CS scheme may be compared to that of a benchmarker, which uses soft-input hard-output LDPC decoding and no iterations within the CS decoder or between the LDPC and CS decoders. The complexity of this hard information benchmarker may be calculated using Eq. (18) from Table III.

The overall complexity of the proposed iterative joint LDPC-CS decoding scheme takes into account each operation of all VNs, CNs and SNs, and may be calculated according to Eq. (19) of Table III. Explicitly, the term of $\mathcal{C}_{S_m}$ in Eq. (19) represents the complexity of the SNs, which may be replaced by the superscripted versions provided in Section III III for each of the e-MAP, SD and HD approaches. Note that the complexity associated with a tree search depends on the specific *a priori* LLRs $\tilde{\gamma}^a$ provided for the SN, hence we rely on Monte Carlo simulations for quantifying the complexity of SD and HD approaches.

Fig. 9 characterizes the average number of additions and max evaluations performed by the proposed iterative joint LDPC-CS decoding scheme as a function of the sparsity upper bound K', when employing the same parameters as in the EXIT charts of Section IV with the number of iterations performed within the LDPC and CS decoding of $i_{LDPC} = 16$, $i_{CS} = 3$, for each of the different approaches. As shown in Fig. 9, the complexity of the proposed HD approach is independent of the sparsity upper bound K'. This is because the complexity is dominated by the number of combinations considered by the respective approaches, and in the case of the HD approach using HD $= 1$, the number of combinations considered using Eq. (11) depends only on the degree of sensors, but not on the sparsity upper bound K'. By contrast, the complexity of the proposed SD approach increases with the sparsity upper bound K'. This may be explained by the

TABLE III: Complexity equations for different approaches.

hard information	$\mathcal{C}_{LDPC-CS} = i_{LDPC} \times \left(\sum_{v=1}^{V} \mathcal{C}_{VN_v} + \sum_{c=1}^{C} \mathcal{C}_{CN_c}\right) + \mathcal{C}_{CS}^{hard}$. (18)	
e-MAP	$\mathcal{C}_{S_m}^{e-MAP}$ (7)	$\mathcal{C}_{LDPC-CS} = i_{LDPC-CS} \times \left(i_{LDPC} \times \left(\sum_{v=1}^{V} \mathcal{C}_{VN_v} + \sum_{c=1}^{C} \mathcal{C}_{CN_c}\right) + (i_{CS}+1) \times \sum_{m=1}^{M} \mathcal{C}_{S_m} + i_{CS} \times \sum_{n=1}^{N} \mathcal{C}_{V_{VN_n}}\right)$. (19)
SD	$\mathcal{C}_{S_m}^{SD}$ (9)	
HD	$\mathcal{C}_{S_m}^{HD}$ (12)	

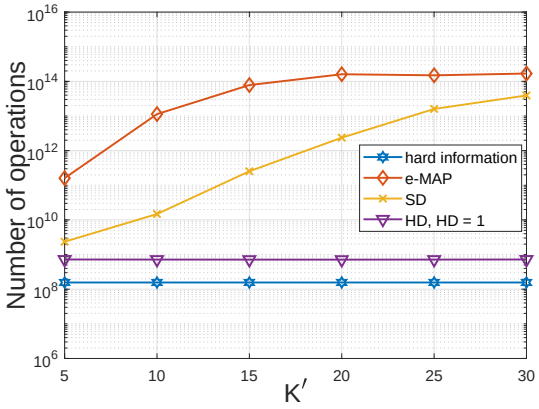


Fig. 9: Complexity of different approaches for the case of $N = 500$ signals, $M = 150$ sensors, a degree for each signal of $K_s = 5$, sparsity upper bound $K' = 5$, LDPC BG of 2, coding rate $R = \frac{1}{3}$, $T = 500$ TSs, $i_{LDPC} = 16$, $i_{CS} = 3$, threshold of finding most likely combination $\xi = 10$ and threshold of finding counter-hypotheses $\varphi = 100$, using QPSK modulation for communication over an AWGN channel.

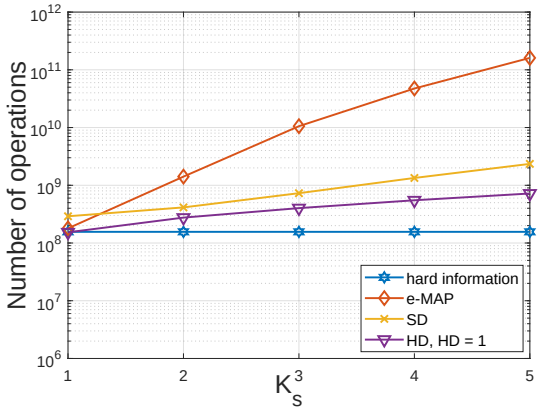


Fig. 10: Complexity of different approaches for the case of $M/N = 3/10$, a degree for each signal of $K_s = 5$, sparsity upper bound $K' = 5$, LDPC BG of 2, coding rate $R = \frac{1}{3}$, $T = 500$ TSs, threshold of finding most likely combination $\xi = 10$ and threshold of finding counter-hypotheses $\varphi = 100$, $i_{LDPC} = 16$, $i_{CS} = 3$, using QPSK modulation over an AWGN channel.

higher complexity of the tree that results from using a higher sparsity upper bound K', typically leading to more nodes and more combinations in the SD approach, hence increasing the complexity. For the e-MAP approach, the complexity initially grows rapidly as the sparsity upper bound K' is increased before such a rating. As shown in Eq. (3), the number of combinations considered by the e-MAP approach is influenced by the sparsity upper bound K' when K' is lower than the SN degree d_m, but when K' is increased above d_m, then the complexity only depends on the value of d_m. The complexity of the SSCC hard information benchmarker only depends on the number of signals N, sensors M and number of TSs T, as shown in Eq. (18). Owing to this, the complexity of the SSCC hard information approach is not influenced by the sparsity upper bound K', as shown in Fig. 9. As expected, Fig. 9 shows that the SSCC hard information approach has the lowest complexity. But it may be observed that the specific choice of the scheme significantly inferences the joint source-channel coding complexity relative to the e-MAP approach.

Fig. 10 characterizes the complexity of the proposed iterative joint LDPC-CS as a function of the degree of the signals K_s for each of the different approaches, when employing the same fundamental parameters as in the EXIT charts of Section IV and the number of iterations inside within the LDPC decoder is $i_{LDPC} = 16$ and within the CS decoder is $i_{CS} = 3$. As the degree of signals K_s is increased, the average degree of sensors is also increased, and hence the number of combinations considered by the SNs of the proposed SD approach, HD approach and e-MAP approach escalates accordingly. Fig. 10 shows that the rate of complexity increase as the signal degree K_s is significantly higher for the e-MAP approach than for the proposed SD and HD techniques. But the complexity of the SSCC hard information approach does not depend on the degree K_s of signals since no iterative decoding is performed within the CS decoder or between the LDPC and CS decoder.

VI. Block Error Rate Results

This section characterizes the BLER performance of the proposed joint LDPC-CS decoding scheme, which is compared to benchmarkers. The fundamental parameters adopted in the following investigations are the same as those mentioned above in Section IV, where we adopt $N = 500$ signals, $M = 150$ sensors, each signal has a degree of $K_s = 5$, an LDPC coding rate of $R = \frac{1}{3}$, a true sparsity of $K = K' = 5$, an LDPC block length that comprises $T = 500$ TSs, 3GPP BG 2 and QPSK modulation for communication over an AWGN channel. More specifically, we compare the BLER performance of the proposed SD and HD approaches against the SSCC hard information approach as well as e-MAP approach and investigate the impact of using different numbers of iterations in the various schemes. Our results show that the proposed SD and HD approaches can maintain the same performance as the e-MAP approach, while considerably reducing the complexity. Moreover, the BLER results of the joint LDPC-CS decoding schemes show that the use of soft information allows iterative decoding within the CS decoder to eliminate the limitation associated with CS decoder in the SSCC hard information benchmarker. Joint LDPC-CS decoding allows iterations between the LDPC and CS decoder, which can further improve the decoding performance.

Fig. 11 displays the BLER performance for different JSCC schemes when employing different combinations of

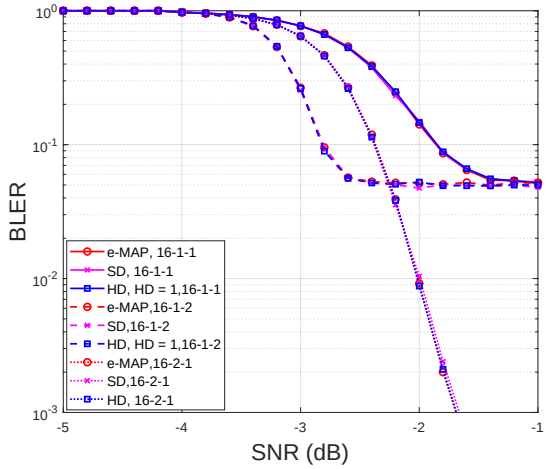


Fig. 11: BLER results for the e-MAP approach, proposed SD approach and proposed HD approach for the case of $N = 500$ signals, $M = 150$ sensors, a degree for each signal of $K_s = 5$, true sparsity $K = K' = 5$, LDPC BG of 2, coding rate $R = \frac{1}{3}$, $T = 500$ TSs, threshold of finding most likely combination $\xi = 10$ and threshold of finding alternative hypotheses $\varphi = 100$, using QPSK modulation for communication over an AWGN channel. The iteration combinations are shown as $i_{LDPC} - i_{CS} - i_{LDPC-CS}$ in the legend.

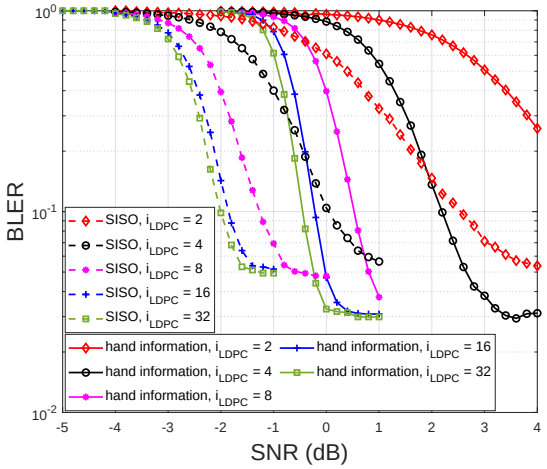


Fig. 12: BLER results for SSCC hard information approach and JSCC HD approach for the case of HD $= 1$ with $N = 500$ signals, $M = 150$ sensors, a degree for each signal of $K_s = 5$, true sparsity $K = K' = 5$, LDPC BG of 2, coding rate $R = \frac{1}{3}$, $T = 500$ TSs, threshold for finding most likely combination $\xi = 10$ and threshold for finding counter-hypotheses $\varphi = 100$, $i_{CS} = 1$, using QPSK modulation for communication over an AWGN channel.

i_{LDPC}, i_{CS} and $i_{LDPC-CS}$ iterations. Fig. 11 shows that the proposed SD and HD approaches always achieve very similar performance as the e-MAP approach, across all combinations of iterations. This confirms the results of Fig. 7, which showed that CS decoding employing the proposed SD and HD approaches can maintain the same decoding performance as the e-MAP approach. According to Eq. (7), the CS decoding complexity relying on the e-MAP approach is around 4.02×10^{10} additions and max operations, which is about 57 times the 7.17×10^{8} additions and max operations of the SD approach based on Eq. (9). Moreover, the complexity of CS decoding employing the e-MAP approach is around 211 times the 1.91×10^{8} additions and max operations of the proposed HD approach, based on Eq. (12). These numerical results demonstrate that the proposed SD approach significantly reduces the complexity compared to the e-MAP approach, and that the HD approach even further reduces it. Since the proposed HD approach achieves the same decoding performance as the e-MAP approach while having the lowest complexity, we adopt the HD approach for the joint LDPC-CS decoding scheme throughout the rest of this section.

Fig. 12 characterizes the BLER of the benchmarker using the hard information, and the SISO scheme using the proposed HD approach. Observe that the BLER performance increases significantly when increasing the number of LDPC iterations towards $i_{LDPC} = 16$, but then offers diminishing returns beyond this point for both the proposed HD approach and the hard information approach in agreement with the results of Fig. 5. Upon considering the performance vs. complexity trdae-off, i_{LDPC} is an attractive choice. The proposed HD approach achieves about 1.7 dB gain at BLER $= 10^{-1}$ compared to the hard information benchmarker, when $i_{LDPC} = 16$, $i_{CS} = 1$. However, Fig. 12 shows that both the proposed HD approach and hard information benchmarker suffer from error floors when $i_{CS} = 1$, which is caused by the limitation of the CS decoding performance. Fortunately, the soft information used in the proposed HD approach allows iterative decoding within the CS decoder and between the LDPC and CS decoder, which eliminates this error floor when $i_{CS} > 1$.

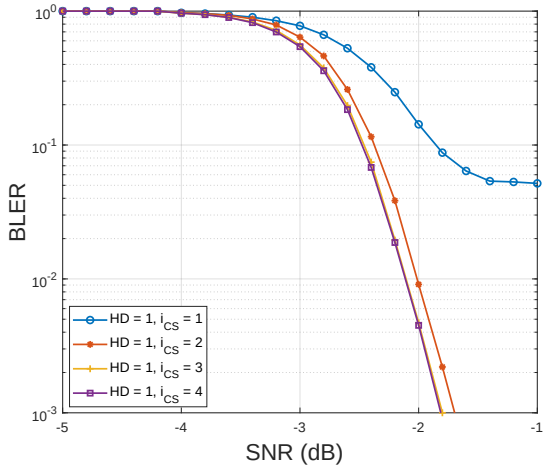


Fig. 13: BLER results for HD approach with various numbers of CS iterations for the case of $N = 500$ signals, $M = 150$ sensors, a degree for each signal of $K_s = 5$, true sparsity $K = K' = 5$, LDPC BG of 2, coding rate $R = \frac{1}{3}$, $T = 500$ TSs, $i_{LDPC} = 16$ in LDPC decoding, threshold for finding most likely combination $\xi = 10$ and threshold for finding counter-hypotheses $\varphi = 100$, $i_{CS} = 1$, using QPSK modulation for communication over an AWGN channel.

Fig. 13 characterizes the BLER performance of the proposed HD approach as a function of SNR for different numbers of CS decoding iterations i_{CS} in the JSCC scheme. It may be observed that an error floor emerges, when adopting only $i_{CS} = 1$ iteration for CS decoding, but this error floor may be removed by increasing the number of iterations to $i_{CS} = 2$. A further BLER improvement is attained by opting for $i_{CS} = 3$, but we observe no further gain beyond $i_{CS} = 3$, which is consistent with the results of Fig. 5. In order to achieve good performance without wasting resources, $i_{CS} = 3$ is the best choice for the number of inner iterations in the CS decoding, which was determined by testing a range of different configurations. Hence, Fig. 13 demonstrates that iterative CS decoding is necessary for avoiding error floors and hence the complexity reduction in this paper is strongly motivated.

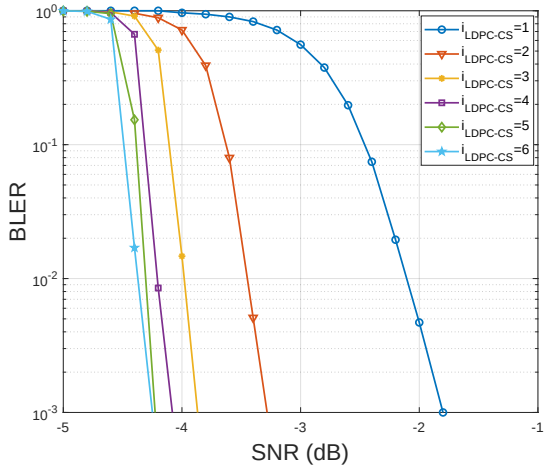


Fig. 14: BLER results for HD approach for various number of iterations between LDPC and CS decoding $i_{LDPC-CS}$ for the case of $N = 500$ signals, $M = 150$ sensors, a degree for each signal of $K_s = 5$, true sparsity $K = K' = 5$, LDPC BG of 2, coding rate $R = \frac{1}{3}$, $T = 500$ TSs, $i_{LDPC} = 16$ in LDPC decoding, $i_{CS} = 3$ in CS decoding, threshold for finding most likely combination $\xi = 10$ and threshold for finding counter-hypotheses $\varphi = 100$, using QPSK modulation for communication over an AWGN channel.

Fig. 14 characterizes the BLER performance of the proposed HD approach for various numbers of iterations $i_{LDPC-CS}$ between LDPC and CS decoder in the JSCC scheme. It may be observed that significant BLER gain is obtained by increasing the number of iterations between the LDPC and CS decoder, although diminishing returns are offered beyond $i_{LDPC-CS} = 6$, which is consistent with the result suggested by the trajectory of Fig. 8. In particular, the adoption of $i_{LDPC-CS} = 6$ offers about 2.45 dB improvement at a BLER 10^{-3} compared to using $i_{LDPC-CS} = 1$ but naturally at the cost of increasing the complexity by a factor of about 6 times.

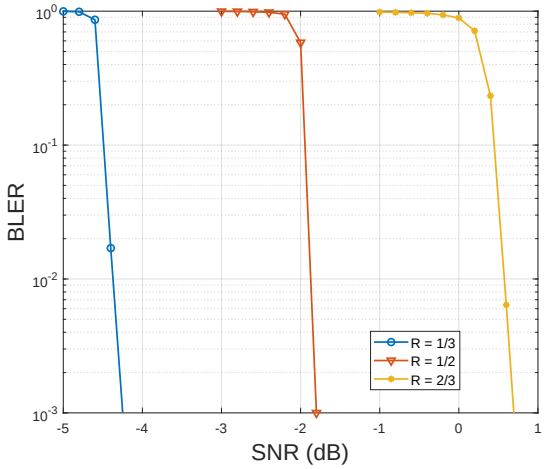


Fig. 15: BLER results of the HD approach when adopting various coding rates R, for the case of $N = 500$ signals, $M = 150$ sensors, a degree for each signal of $K_s = 5$, true sparsity $K = K' = 5$, LDPC BG of 2, $T = 500$ TSs, $i_{LDPC} = 16$ LDPC decoding iterations, $i_{CS} = 3$ CS decoding iterations, $i_{LDPC-CS} = 6$ iterations between LDPC decoding and CS decoding, a threshold for finding the most likely combination of $\xi = 10$ and a threshold for finding counter-hypotheses of $\varphi = 100$, for the case of using QPSK modulation for communication over an AWGN channel.

Fig. 15 characterizes the BLER performance of the proposed HD approach for various LDPC coding rates R. As may be expected, when the coding rate increases, the BLER performance is significantly degraded, but this capability demonstrates the flexibility of the proposed WSN scheme.

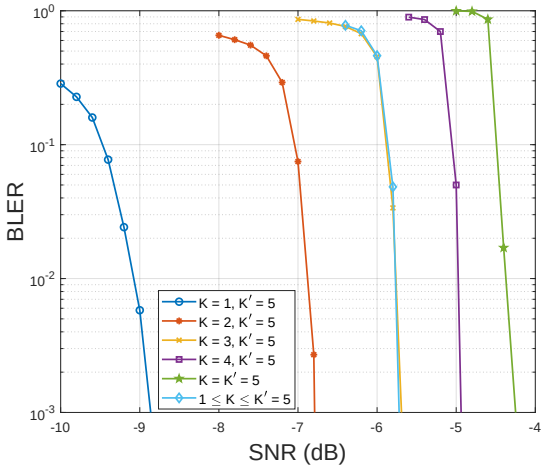


Fig. 16: BLER results of the proposed HD approach when adopting various values for the true sparsity K, for the case of $N = 500$ signals and $M = 150$ sensors. Furthermore, the degree of each signal is $K_s = 5$, the sparsity upper bound $K' = 5$, the LDPC BG is 2, the coding rate is $R = 1/3$, $T = 500$ TSs, and we use $i_{LDPC} = 16$ LDPC decoding iterations, $i_{CS} = 3$ CS decoding iterations, $i_{LDPC-CS} = 6$ iterations between LDPC decoding and CS decoding. We employ a threshold of $\xi = 10$ for finding the most likely combination and a threshold of $\varphi = 100$ for finding counter-hypotheses, for the case of using QPSK modulation for communication over an AWGN channel.

Fig. 16 shows another scenario in that the sparsity upper bound K' is known to the FC, and the true sparsity K is not always equal to the K'. In this case, we define the BLER as the particular fraction of blocks in which not all non-zero values of the signal are identified. As shown in the figure, when the true sparsity K is smaller than the upper bound K', the decoding BLER performance becomes better, and the lower the true sparsity is, the better the decoding performance becomes. The explanation for this is that the lower the value of K, the fewer non-zero entries are defined and the easier it becomes to decode successfully when the FC has the prior knowledge of the sparsity upper bound K'.

VII. CONCLUSIONS

In this paper, we have proposed a pair of novel techniques for reducing the complexity of a joint iterative LDPC-CS decoding scheme, namely the proposed SD and HD approaches. In particular, these dramatically reduce the complexity required to generate high quality *extrinsic* LLRs from the SN. Both techniques employ a tree search to explore the various combinations of signal values and identify the most likely combination. Following this first processing step, the SD approach continues the tree search to obtain alternative hypotheses, which are necessary for generating high-quality *extrinsic* LLRs. By contrast, the HD approach obtains the alternative hypotheses by selecting all candidates within a certain HD of the most likely combination. Our EXIT charts and BLER results demonstrate that the proposed SD and HD techniques achieve the same decoding performance as the full-search based e-MAP approach, which considers all possible combinations of signal values during CS decoding. Our numerical results demonstrate that the e-MAP approach has about 56 times higher complexity than that of the proposed SD approach, and about 210 times higher complexity than that of the proposed HD approach. Furthermore, we compare the proposed HD approach designed for the JSCC scheme to a benchmarker in the SSCC scheme, which performs CS

decoding using hard information. Our results show that the use of soft information in the proposed HD approach improves the decoding performance by about 1.7 dB in the case of QPSK modulation over an AWGN channel. Furthermore, we demonstrate that invoking iterations both within the CS decoder and between the LDPC and CS decoder eliminates the error floor, which allows a further 2.45 dB of gain to be obtained. Our results are validated by EXIT chart analysis, which is consistent with the BLER results.

The proposed FC requires M number of LDPC decoders and hence our future work will consider hardware efficient implementation of this. Among the state-of-the-art techniques for channel coding, LDPC decoding is widely considered to be the most hardware-efficient [40], [41]. In the proposed WSN scheme, all sensors transmit messages comprising the same number T of bits, which presents an opportunity for parallel LDPC decoding to be used. More specifically, the main complexity contribution of an LDPC decoder is typically that of the data handling routine and memory access, rather than the computation. In the case where all M LDPC decoders process messages of the same length, the routine may be considered to be common among them. Hence, an efficient hardware architecture, which shares the routine across all of the M decoders may be conceived, which has M parallel computation units. Furthermore, even if the different sensors are using different coding rates, we may harness the rate matching architecture of the 3GPP LDPC code to overcome this issue in our future research. More specifically, upon reception of short blocks from some sensor nodes employing high coding rates, zero padding can be applied to increase the length of the received vectors of LLRs to match the longest received signal vector of LLRs. In this way, all M LDPC codes may be decoded using the same lowest coding rate, while sharing the associated routines and the memory access.

REFERENCES

[1] S. Vembu, S. Verdu, and Y. Steinberg, "The source-channel separation theorem revisited," IEEE Transactions on Information Theory, vol. 41, no. 1, pp. 44–54, 1995.

[2] C. E. Shannon, "A mathematical theory of communication," The Bell System Technical Journal, vol. 27, no. 3, pp. 379–423, 1948.

[3] A. Zribi, R. Pyndiah, S. Zaibi, F. Guilloud, and A. Bouallegue, "Low-complexity soft decoding of Huffman codes and iterative joint source channel decoding," IEEE Transactions on Communications, vol. 60, no. 6, pp. 1669–1679, 2012.

[4] M. Fresia, F. Peréz-Cruz, H. V. Poor, and S. Verdú, "Joint source and channel coding," IEEE Signal Processing Magazine, vol. 27, no. 6, pp. 104–113, 2010.

[5] J. Yan, J. Huang, and C. Huang, "Deep learning aided joint source-channel coding for wireless networks," in 2021 IEEE/CIC International Conference on Communications in China (ICCC), 2021, pp. 805–810.

[6] C. Karakus, A. C. Gurbuz, and B. Tavli, "Analysis of energy efficiency of compressive sensing in wireless sensor networks," IEEE Sensors Journal, vol. 13, no. 5, pp. 1999–2008, 2013.

[7] E. A. Hodgson, G. Brante, R. D. Souza, J. Garcia-Frias, and J. L. Rebelatto, "Compensating spectral efficiency loss of wireless rf energy transfer with analog joint source channel coding compression," IEEE Sensors Journal, vol. 16, no. 16, pp. 6458–6469, 2016.

[8] J. Zheng, H. Zhang, Y. Cai, R. Li, and A. Anpalagan, "Game-theoretic multi-channel multi-access in energy harvesting wireless sensor networks," IEEE Sensors Journal, vol. 16, no. 11, pp. 4587–4594, 2016.

[9] Q. Chen and L. Wang, "Design and analysis of joint source channel coding schemes over non-standard coding channels," IEEE Transactions on Vehicular Technology, vol. 69, no. 5, pp. 5369–5380, 2020.

[10] M. A. M. Izhar, A. J. Aljohani, S. X. Ng, and L. Hanzo, "Distributed joint source coding and trellis coded modulation for symbol-based Markov sources," IEEE Transactions on Vehicular Technology, vol. 67, no. 5, pp. 4031–4041, 2018.

[11] C. Li, G. Li, B. Kailkhura, and P. K. Varshney, "Secure distributed detection of sparse signals via falsification of local compressive measurements," IEEE Transactions on Signal Processing, vol. 67, no. 18, pp. 4696–4706, 2019.

[12] A. Mohammadi, D. Ciuonzo, A. Khazaee, and P. S. Rossi, "Generalized locally most powerful tests for distributed sparse signal detection," IEEE Transactions on Signal and Information Processing over Networks, vol. 8, pp. 528–542, 2022.

[13] F. Chen, F. Lim, O. Abari, A. Chandrakasan, and V. Stojanovic, "Energy-aware design of compressed sensing systems for wireless sensors under performance and reliability constraints," IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 60, no. 3, pp. 650–661, 2013.

[14] S. Feizi and M. Medard, "A power efficient sensing/communication scheme: Joint source-channel-network coding by using compressive sensing," in 2011 49th Annual Allerton Conference on Communication, Control, and Computing (Allerton), 2011, pp. 1048–1054.

[15] S. Padalkar, A. Korlekar, and U. Pacharaney, "Data gathering in wireless sensor network for energy efficiency with and without compressive sensing at sensor node," in 2016 International Conference on Communication and Signal Processing (ICCSP), 2016, pp. 1356–1359.

[16] L. Zheng, T. Zhu, and X. Ma, "Block compressed sensing-based joint source-channel coding for wireless image transmission," in 2020 International Conference on Wireless Communications and Signal Processing (WCSP), 2020, pp. 13–18.

[17] R. Ma, Q. Hao, X. Hu, and C. Wang, "Space coding schemes for multiple human localization with fiber-optic sensors," IEEE Sensors Journal, vol. 18, no. 11, pp. 4643–4653, 2018.

[18] L. Hanzo and T. Liew, Turbo Coding, Turbo Equalisation and Space-Time Coding for Transmission over Fading Channels. John Wiley & Sons, Ltd, 10 2002.

[19] A. Golmohammadi and D. G. M. Mitchell, "Concatenated spatially coupled LDPC codes with sliding window decoding for joint source-channel coding," IEEE Transactions on Communications, vol. 70, no. 2, pp. 851–864, 2022.

[20] C. Studer and H. Bölcskei, "Soft-input soft-output single tree-search sphere decoding," IEEE Transactions on Information Theory, vol. 56, no. 10, pp. 4827–4842, 2010.

[21] L. Hanzo and P. Cherriman, Video Compression and Communications : From Basics to H.261, H.263, H.264, MPEG4 for DVB and HSDPA-Style Adaptive Turbo-Transceivers. Wiley-IEEE Press, 11 2007.

[22] B. Hochwald and S. ten Brink, "Achieving near-capacity on a multiple-antenna channel," IEEE Transactions on Communications, vol. 51, no. 3, pp. 389–399, 2003.

[23] I. E. Bocharova, A. Guillén i Fàbregas, B. D. Kudryashov, A. Martinez, A. Tauste Campo, and G. Vazquez-Vilar, "Multi-class source-channel coding," IEEE Transactions on Information Theory, vol. 62, no. 9, pp. 5093–5104, 2016.

[24] X. Wang and X. Wu, "Index assignment optimization for joint source-channel MAP decoding," IEEE Transactions on Communications, vol. 58, no. 3, pp. 901–910, 2010.

[25] B. Knoop, T. Wiegand, and S. Paul, "Low-complexity and approximative sphere decoding of sparse signals," in 2012 Conference Record of the Forty Sixth Asilomar Conference on Signals, Systems and Computers (ASILOMAR), 2012, pp. 1241–1244.

[26] D. Ciuonzo, G. Romano, and P. S. Rossi, "Channel-aware decision fusion in distributed mimo wireless sensor networks: Decode-and-fuse vs. decode-then-fuse," IEEE Transactions on Wireless Communications, vol. 11, no. 8, pp. 2976–2985, 2012.

[27] P. Suarez-Casal, O. Fresnedo, L. Castedo, and J. Garcia-Frias, "Analog transmission of correlated sources over fading SIMO multiple access channels," IEEE Transactions on Communications, vol. 65, no. 7, pp. 2999–3011, 2017.

[28] M. Mohammadkarimi, M. Mehrabi, M. Ardakani, and Y. Jing, "Deep learning-based sphere decoding," IEEE Transactions on Wireless Communications, vol. 18, no. 9, pp. 4368–4378, 2019.

[29] C. Xu, S. X. Ng, and L. Hanzo, "Multiple-symbol differential sphere detection and decision-feedback differential detection conceived for differential QAM," IEEE Transactions on Vehicular Technology, vol. 65, no. 10, pp. 8345–8360, 2016.

[30] J. W. Choi, B. Shim, Y. Ding, B. Rao, and D. I. Kim, "Compressed sensing for wireless communications: Useful tips and tricks," IEEE

Communications Surveys & Tutorials, vol. 19, no. 3, pp. 1527–1550, 2017.

[31] "Physical layer procedures for data (Release 15)," 3GPP Standard TS, November 2017. [Online]. Available: https://portal.3gpp.org/ngppapp/CreateTdoc.aspx?mode=view&contributionId=840513

[32] Y. Rosmansyah and U. of Surrey, "Soft-demodulation of QPSK and 16-QAM for turbo coded WCDMA mobile communication systems." University of Surrey, 2003, pp. 54–55. [Online]. Available: https://books.google.co.uk/books?id=tZarnQEACAAJ

[33] X.-Y. Hu, E. Eleftheriou, D.-M. Arnold, and A. Dholakia, "Efficient implementations of the sum-product algorithm for decoding LDPC codes," in GLOBECOM'01. IEEE Global Telecommunications Conference (Cat. No.01CH37270), vol. 2, 2001, pp. 1036–1036E vol.2.

[34] A. Movahed, M. C. Reed, N. Aboutorab, and S. E. Tajbakhsh, "EXIT chart analysis of turbo compressed sensing using message passing dequantization," IEEE Transactions on Signal Processing, vol. 64, no. 24, pp. 6600–6612, 2016.

[35] X. Zuo, R. G. Maunder, and L. Hanzo, "Design of fixed-point processing based LDPC codes using EXIT charts," in 2011 IEEE Vehicular Technology Conference (VTC Fall), 2011, pp. 1–5.

[36] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," IEEE Transactions on Communications, vol. 49, no. 10, pp. 1727–1737, 2001.

[37] S. X. Ng, J. Wang, M. Tao, L.-l. Yang, and L. Hanzo, "Iteratively decoded variable length space-time coded modulation: Code construction and convergence analysis," IEEE Transactions on Wireless Communications, vol. 6, no. 5, pp. 1953–1963, 2007.

[38] M. Fossorier, M. Mihaljevic, and H. Imai, "Reduced complexity iterative decoding of low-density parity check codes based on belief propagation," IEEE Transactions on Communications, vol. 47, no. 5, pp. 673–680, 1999.

[39] M. Baldi, G. Cancellieri, A. Carassai, and F. Chiaraluce, "LDPC codes based on serially concatenated multiple parity-check codes," IEEE Communications Letters, vol. 13, no. 2, pp. 142–144, 2009.

[40] P. Hailes, L. Xu, R. G. Maunder, B. M. Al-Hashimi, and L. Hanzo, "Hardware-efficient node processing unit architectures for flexible ldpc decoder implementations," IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 65, no. 12, pp. 1919–1923, 2018.

[41] S. Shao, P. Hailes, T.-Y. Wang, J.-Y. Wu, R. G. Maunder, B. M. Al-Hashimi, and L. Hanzo, "Survey of turbo, ldpc, and polar decoder asic implementations," IEEE Communications Surveys & Tutorials, vol. 21, no. 3, pp. 2309–2333, 2019.