A Simplified Application of Ordered Statistics Decoding to Single Parity Check Product Codes

Roberto Garello
Department of Electronics and Telecommunications
Politecnico di Torino
Torino, Italy
garello@polito.it

Giacomo Verardo
Department of Electronics and Telecommunications
Politecnico di Torino
Torino, Italy
giacomo.verardo@polito.it

***Abstract*—In this paper we consider Ordered Statistics Decoding and we discuss some ideas aiming to reduce its complexity. As a case study, we focus on Single Parity Check Product Codes. First, we investigate how to simplify the construction of a reliable basis by exploiting the code structure. Then, we consider the iterative application of Soft-Input Soft-Output Ordered Statistics Decoding with small order to lower-dimensional subcodes. Results show that these simplified algorithms, both stand-alone and iterative, are still able to approach Maximum Likelihood Decoding of Single Parity Check Product Codes.**

***Index Terms*—Single Parity Check Product Codes, Ordered Statistics Decoding, Iterative Decoding**

I. Introduction and Motivations

Ordered Statistics Decoding (OSD) is a soft decoding technique introduced by Fossorier and Lin in 1995 [1] and further elaborated in a number of following papers [2], [3]. Given a binary code $C(n, k)$, OSD consists basically of four steps divided in two phases. Phase A (preparation): (i) ordering of the received n real values by decreasing reliability; (ii) joint construction of the k-bit Most Reliable Basis (MRB) and of a systematic generating matrix G using it as information set. Phase B (evaluation): (iii) generation and distance evaluation of the first candidate codeword obtained by encoding the k MRB bits; (iv) generation and distance evaluation of all the codewords obtained by changing up to I of the k MRB bits, and selection of the best one.

The idea behind OSD is that when we identify the k MRB bits, with high probability they contain few "errors". Then, it is enough to change a small number of them to generate a list of candidate codewords containing the Maximum Likelihood (ML) one, i.e., the codeword at minimum Euclidean distance from the received vector, which is finally chosen as the received codeword.

The parameter I, called the *order* of the OSD algorithm, has clearly a fundamental impact on the algorithm complexity, because the number of codewords to be generated and evaluated is in principal

$$N_c = \sum_{i=0}^{I} \binom{k}{i}. \tag{1}$$

Among the fundamental results derived by Fossorier and Lin in [1], there is the estimation of the minimum order I such that with high probability the ML codeword is inside the list of candidate codewords, given by:

$$I_{\min} = \left\lceil \frac{d_{\min}}{4} \right\rceil, \tag{2}$$

where $d_{\min}$ is the minimum distance of the code. Moreover, it is possibly to develop techniques to reduce the number of candidate codewords to be tested, or to early stop the algorithm. By applying them, a number of results has confirmed that OSD allows to approach the Maximum Likelihood decoding performance of powerful codes with lengths up to some hundreds of bits [1]–[7].

This is extremely important for practical applications. In fact, we know there is currently a lot of interest on short codes. The reason is that 5G and beyond 5G networks are evolving to include ultra-reliable, low-latency wireless communication with short packets [8], [9]. For these use cases, the design of powerful short codes that can be encoded and decoded with small complexity is a key issue [10], [11]. In the short block length regime, Low-Density Parity-Check (LDPC) codes, Turbo Codes and Polar Codes are still good, but not so close to ideal limits as for long codeword lengths. Usually, in this region they can be outperformed by algebraic codes (like Bose–Chaudhuri–Hocquenghem (BCH) or Reed-Solomon codes) [7]. (It is interesting to note that the Polar Code scheme recently selected for 5G control channels [12] exploits an inner CRC code to further improve its performance [11].) Then, the availability of a nearly-optimal, low-complexity soft decoding algorithm could lead to a revamping of "classical" codes, or codes obtained by combining them, for the new applications.

Anyway, the complexity of OSD for powerful codes can still be an obstacle to its real-time implementation. As an example, let's consider the extended BCH code C(128,64) with minimum distance $d_{\min} = 22$, which is guessed to be the most powerful code with these parameters. For this code, the total number of different codewords to be evaluated by an order $I = 4$ OSD algorithm is in principle 679,121. Even if some reduction can certainly be applied (see for example [3] or [7], where 200,000 patterns only were considered), these numbers still look too big for a low complexity implementation.

As another example, CCSDS (Consultative Committee for Space Data Systems), that reunites the most important Space Agencies, has recently updated its Recommendation for Telecommand (Ground to Space) systems to include LDPC codes as new coding options [13]. When applied to the LDPC(128,64) code of [13], simulations show that OSD can achieve a significant gain of more 1.5 dB with respect to usual iterative decoding (sum-product, min-sum, normalized min-sum) [14]. A real-time implementation of OSD (probably one of the first ever) has been realized for this code within the framework of an ESA contract [15]. If on one side the mixed hardware/software implementation described in [15] confirmed the performance gain, on the other side the achieved information bit rate was limited to about 2 kbit/s.

All these elements suggest that any idea to further reduce OSD complexity could ease its application to practical systems. The scope of this conference paper is indeed to discuss some simple ideas for this purpose. As a case study, we focus on Single Parity Check (SPC) codes and its product concatenation.

If we consider the two phases of OSD, most of the attention in the literature has been devoted to the second one, aiming to reduce the number of codewords to be tested or to early stop the algorithm. This is reasonable, because the complexity of this phase is much bigger. Anyway, as [15] pointed out, Phase A complexity is not negligible, especially when many column swaps of the generator matrix are needed to find an information set. Moreover, if a high degree of parallelism can strongly reduce the complexity of Phase B (in principle, any codeword can be generated and tested in parallel), this is not true for Phase A.

For this reason, in this paper we study OSD simplification with two goals: (i) reduce Phase A complexity by simplifying the construction of a reliable basis, and (ii) reduce Phase B complexity by using small OSD orders. The basic ideas contained in this paper are:

- Reliability ordering is obviously simpler when applied to shorter vectors.
- The code structure can be exploited to identify a Reliable Basis, equal or close to the Most Reliable Basis, which is easy to compute and avoids long generator matrix elaboration. (In other words, we look for a simplified procedure to select an information set composed by very reliable bits.)
- If OSD is applied to codes with $d_{\min} \leq 4$, order $I_{\min} = 1$ is enough and the number of codewords to be tested is very small.
- Working on subcodes allows to implement multiple simplified SISO OSD copies working in parallel, iteratively exchanging their soft information.

In the following, simulation results show that (stand-alone or iterative) simplified OSD obtained by applying these ideas can still approach Maximum Likelihood decoding of 2D and 3D SPC product codes.

The paper is organized as follows. In Section II we introduce Single Parity Check Product codes. In Section III we present OSD and its Soft-Input-Soft-Ouput (SISO) version. In Section IV we briefly discuss the application of OSD to one-dimensional Single Parity Check Codes and its link to min-sum iterative decoding. In Section V we focus on two-dimensional SPC product codes and present some techniques to build a Reliable Basis without matrix manipulation. In Section VI we consider three-dimensional SPC product codes and discuss how to build a Reliable Basis with few matrix manipulation. In Section VII we discuss how to decode 3D SPC product codes by the iterative application of simplified OSD working on 2D subcodes. Section VIII concludes.

II. Single Parity Check Product Codes

Single Parity Check codes are $P_k(n = k+1, k)$ binary linear codes with minimum distance $d_{\min} = 2$ and multiplicity (number of codewords with weight $d_{\min}$) $A_{\min} = k + \binom{k}{2}$.

An M-dimensional Single Parity Check Product Code $C_M(n_M, k_M) = \otimes_{m=1}^{M} P_{k_m}(n_m, k_m)$ has parameters

$$n_M = \prod_{m=1}^{M} n_m \quad k_M = \prod_{m=1}^{M} k_m.$$

Its (non-systematic) generator matrix is obtained by the Kronecker product of the generator matrix of the constituent SPC codes [16].

In [16] Caire, Taricco and Battail proved that the distance profile of a multi-dimensional SPC product code approaches that of a random code if the smallest code length in the product grows to infinity. In the same paper, they showed that the best choice consists of taking all constituent SPC codes of equal length, a choice that we also adopt in this paper: $\forall m \;\; k_m = k$.

The $k_M = k^M$ information bits are written into an M-dimensional hypercube with side k, then encoded in all M directions. Examples of codewords for 1D (a single SPC code), 2D and 3D product codes based on the $P_2(3,2)$ code are shown in Fig. 1.

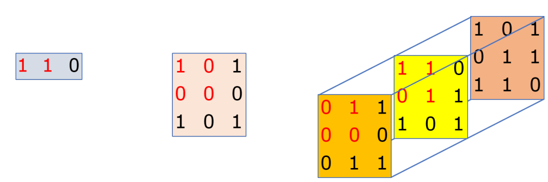

Fig. 1. Example of 1D, 2D and 3D SPC product codewords based on $P_2(3,2)$. Information bits in red, parity bits in black.

It is easy to show that SPC product codes have parameters:

$$d_{\rm min,M} = 2^M \quad A_{\rm min,M} = (A_{\min})^M \,. \tag{3}$$

Moreover, they have a transitive automorphism group, then the multiplicty property holds [17], and the information multiplicity (sum of information weights for all the $A_{\min}$ codewords) can be computed by the formula

$$w_{\rm min,M} = A_{\rm min,M} \;\; d_{\rm min,M} \;\; \frac{k_M}{n_M}. \tag{4}$$

The two multiplicity values $A_{\rm min,M}$ and $w_{\rm min,M}$ allow to compute the truncated union bound involving the minimum distance term only, which represents an excellent estimation of the Maximum Likelihood Codeword Error Rate (CER) and Bit Error Rate (BER) performance at medium/high Signal-to-Noise Ratio (SNR), i.e., for CER $\leq 10^{-3}$ [17]:

$$\text{CER} \simeq \frac{1}{2} A_{\min,\mathrm{M}} \text{erfc} \sqrt{d_{\min,\mathrm{M}} \frac{k_M}{n_M}}, \tag{5}$$

$$\text{BER} \simeq \frac{1}{2} \frac{w_{\min,\mathrm{M}}}{k_M} \text{erfc} \sqrt{d_{min,M} \frac{k_M}{n_M}}. \tag{6}$$

III. Ordered Statistics Decoding and its SISO version

Given a binary code $C(n, k)$ with generator matrix G, let $\underline{r} = (r_1, ..., r_n)$ by the received real vector output by the soft demodulator. The OSD algorithm of order I consists of the following steps [1]:

1) Order the n components for decreasing reliability. (In this paper we consider a 2-PSK constellation transmitted over an Additive White Gaussian Noise (AWGN) channel, then we order for decreasing magnitude.)
2) Start from the vector $\underline{v}^*$ made by the k most reliable components and build a systematic form of the generator matrix G^* where the information bits coincide with $\underline{v}^*$. If $\underline{v}^*$ is not an information set (i.e., the corresponding columns of G are not linearly independent), slightly change $\underline{v}^*$ (by swapping the linear dependent columns with the first unused columns with position from $k + 1$ to n), until $\underline{v}^*$ is an information set and the systematic G^* can be built.
3) Encode $\underline{v}^*$ by G^* to get the first candidate codeword $\underline{c}^*$. Compute its Euclidean distance from $\underline{r}$. Initialize the best codeword and its metric.
4) Consider all k-bit vectors (also called error patterns) $\underline{p}$ with weight less or equal to I. For each of them compute the vector $\underline{v}' = \underline{v}^* + \underline{p}$, encode it by G^* to get $\underline{c}'$ and compute its distance from $\underline{r}$. If smaller than the current value, update the best codeword and its metric. At the end of the process, release the best codeword as the chosen received codeword.

To speed up the algorithm, it is possible to stop when the current best distance (computed on the entire codeword or a portion only) is below a given threshold (see [7] and references therein). Moreover, a substantial reduction of the number of codewords to be tested can be obtained by pattern reordering. In fact, it can be shown that different patterns with the same weight have different reliability [5]. Then, we can generate off-line a list of patterns to be tested with decreasing probability and stop their processing when the most probable ones have been considered. This approach allows to strongly reduce the number of patterns with limited performance penalty (as an example, only 200,000 out of 679,121 were considered in [7] for the OSD decoding of an eBCH(128,64) code).

In [2], the Soft-Input Soft-Output (SISO) version of OSD was presented. The optimal SISO should deliver, for each component $1 \leq j \leq n$ a Logarithmic Likelihood Ratio (LLR) value

$$L_j = \log \frac{\sum_{\underline{c} \in C, c_j = 1} P(\underline{c}|\underline{r})}{\sum_{\underline{c} \in C, c_j = 0} P(\underline{c}|\underline{r})}. \tag{7}$$

As an alternative it is possible to apply the sub-optimal Max-Log-MAP algorithm, which delivers

$$L_j = \log \frac{\max_{\underline{c} \in C, c_j = 1} P(\underline{c}|\underline{r})}{\max_{\underline{c} \in C, c_j = 0} P(c|r)}. \tag{8}$$

To compute this value by OSD, Fossorier and Lin proposed the two-stage order-I reprocessing decoding described in [2]. SPC structure allows a further simplification, as discussed in the next sections.

IV. SISO Ordered Statistics Decoding for 1D Single Parity Check codes and applications

In this section we briefly discuss the application of OSD to one-dimensional SPC. Even if its application to higher dimensions is not the true scope of this work, what follows can be useful to better understand some properties of OSD over SPC codes. For 1D Single Parity Check codes, OSD is very simple due to these properties:

1) Since $k = n - 1$, we do not really need to order the entire received vector $\underline{r}$, but only to identify its least reliable bit.
2) Since any set of k-bits is an information set, no matrix reordering is needed. The least reliable bit can be generated by simply summing the MRB ones.
3) An order $I = 0$ is enough, i.e., no pattern must be tested to determine the minimum distance codeword, because the MRB bits automatically identify the minimum distance codeword. (This property immediately follows from MRB definition. The only bit outside MRB has the least reliability. Changing an MRB bit inverts this bit and the least reliable one. The distance becomes bigger.)

If we now consider the SISO version of OSD [2], we can adapt it to the 1D SPC code by noting that:

- An order $I = 1$ OSD is enough to obtain the SISO LLR values.
- For each bit, the most likely codeword is the output of the order-0 OSD.
- For all MRB bits $j \in \text{MRB}$, the most likely codeword with inverted bit is obtained by changing this bit and the least reliable one. Since all the other bits are the same, for the output LLR we have:

$$L_j = \log \frac{\max_{c_j = 1} P(c|r)}{\max_{c_j = 0} P(c|r)} = L(c_j) \pm L(c_{j'}), \tag{9}$$

  where the last sign is positive if, in the most likely codeword, the last bit was one (or negative if zero), so it depends if the other bits agree or not.
- For the least reliable bit $j' \notin \text{MRB}$, the most likely codeword with inverted bit is obtained by changing this

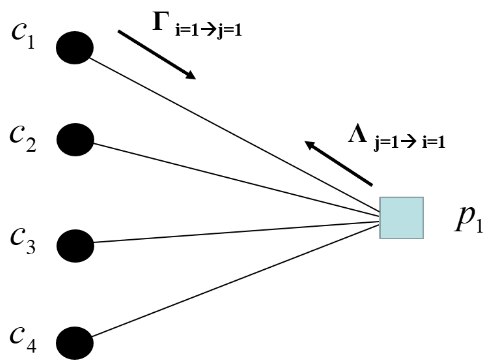

Fig. 2. Bipartite graph representation of 1D SPC (4,3).

bit and the least reliable bit inside MRB. If j'' denotes the position of the second, for the output LLR we have:

$$L_{j'} = \log \frac{\max_{c_{j'}=1} P(c|r)}{\max_{c_{j'}=0} P(c|r)} = L(c_{j'}) \pm L(c_{j''}), \quad (10)$$

where again the last sign depends on the other bits.

Then, the only difference between the input and output LLR values of a given bit is the smallest LLR among the other bits. It is plain to note that these equations are very similar to the check node processing for an LDPC-like bipartite graph representation of the SPC, when the MinSum update rule is applied. Taking as a reference the bipartite graph of an SPC code depicted in Fig. 2, the messages sent from the check node to the variable nodes are given by this equation:

$$\Lambda_{j\rightarrow i} = \left[\prod_{h\in A(j)\backslash i} \operatorname{sgn}\left(\Gamma_{h\rightarrow j}\right) \right] \cdot \left[\min_{h\in A(j)\backslash i} |\Gamma_{h\rightarrow j}| \right] \quad (11)$$

where, as usually, the symbol $A(j)\backslash i$ represents all the variable nodes connected to check node j, but variable node i. By comparing (11) with (9) and (10), we note they are applying the same reprocessing.

V. OSD for 2D SPC Product Codes: Simplified Basis Construction

In this section we discuss how to generalize the ideas presented in the previous section to 2D SPC product codes. Since every 2D SPC product code has distance $d_{\min} = 4$, then with high probability order $I_{\min} = 1$ OSD is sufficient to find the minimum distance codeword. The problem is that, in this case, it is no longer true that every set of k^2 bits is an information set as for the 1D SPC code: matrix reduction is required. If the starting set of most reliable bits is not an information set it is necessary to swap some columns, a simple but not costless operation.

Now, let us suppose that, before applying matrix reduction, we are able to delete a significant number of bits that will certainly not be part of the Most Reliable Basis. This way, the following construction will certainly be simplified because applied to a much smaller set. Now, if we take any parity check equation of the code and we consider its least reliable bit, it will certainly not belong to the MRB. By looking at 2D product code structure, all rows and columns are parity check equations. Then we can analyze them in parallel and cancel their least reliable bit. This suggests the following algorithm.

Algorithm A. Given a 2D SPC codeword:

- There are n rows and n columns, each of them is an n-bit vector. Process these $2n$ vectors in parallel and for each of them delete the least reliable bit (you may have collisions, i.e., some of the deleted bits can be the least reliable for both the two vectors it belongs). Let us denote by $n \leq \alpha \leq 2n-1$ the number of deleted bits.
- Build the Most Reliable Basis working on the survived $n^2-\alpha$ bits.
- Apply the order 1 OSD on this Most Reliable Basis.

This algorithm allows to reduce the complexity of phase A because the MRB is built starting from a quite smaller set. Ordering is faster and for matrix reduction we have a significant decrease of column swap operations. The price to pay is the initial row and column processing, but they can be analyzed in parallel.

As an alternative, we have considered a method to directly obtain an alternative information set, that we call a Simplified Reliable Basis (SRB).

Algorithm B. Given a 2D SPC codeword:

- For each row identify the two least reliable bits, erase the last one.
- Erase the row with least reliable bit.

The process is further explained by the example depicted in Fig. 3.

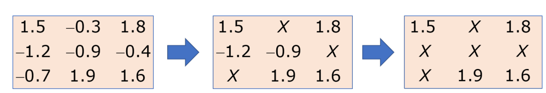

Fig. 3. Identification of the Simplified Reliable Basis for a $(3,2)^2$ code.

The Simplified Reliable Basis may be slightly different from the exact Most Reliable Basis, but has two big advantages:

- We do not need to reorder the entire 2D matrix of n^2 bits, but only to identify the two least reliable bits of each row (an operation that can be done in parallel for each row, which is a short vector of n bits).
- We do not need to swap the columns and reduce the generator matrix: by construction the SRB is automatically an information set because the surviving positions are enough to determine the erased ones.

Even if the SRB is not exactly the best MRB, when we apply the OSD we obtain pretty good performance. In Fig. 4 and Fig. 5 we consider the $(5,4)^2$ and $(7,6)^2$ 2D SPC codes and we show the CER curves for:

- The truncated union bound, computed by equation (5).
- For the first code (2^{16} codewords), the ML exhaustive minimum distance algorithm.
- The original OSD algorithm with order $I = 1$.
- The OSD algorithm with order $I = 1$ applied on the Reliable Basis built with Algorithm A.

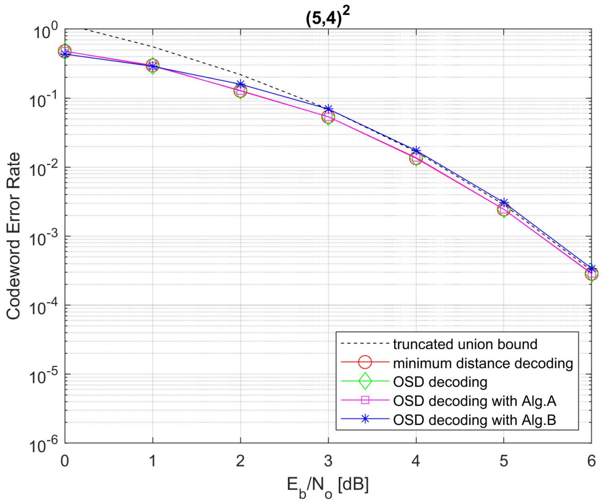

Fig. 4. Single Parity Check Product Code $(5,4)^2$. Truncated union bound, minimum distance decoding, original OSD ($I = 1$), OSD ($I = 1$) with Algorithm A, OSD ($I = 1$) with Algorithm B.

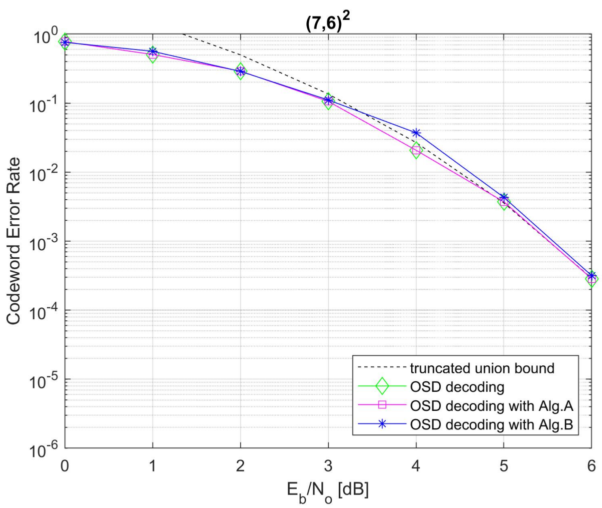

Fig. 5. Single Parity Check Product Code $(7,6)^2$. Truncated union bound, original OSD ($I = 1$), OSD ($I = 1$) with Algorithm A, OSD ($I = 1$) with Algorithm B.

- The OSD algorithm with order $I = 1$ applied on the Reliable Basis built with Algorithm B.

By looking at the figures we can observe that: the original OSD coincides with ML decoding; as expected, the basis generated by Algorithm A is optimal; the basis generated by Algorithm B is slightly sub-optimal but still achieves very good performance, despite the simplicity of its construction.

VI. OSD for 3D SPC product codes: simplified basis construction

If we consider the application of the OSD algorithm to the entire 3D SPC product code, its minimum distance is $d_{\min} = 8$, then we have $I_{\min} = 2$, which still looks reasonable. As done in the previous section, our first goal is to simplify the preparatory Phase A of OSD.

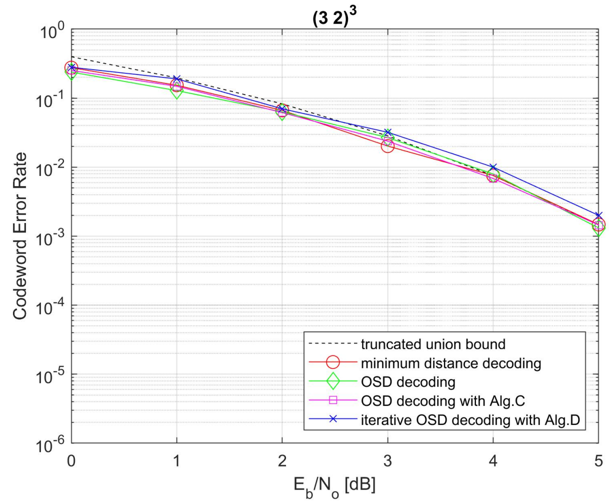

Fig. 6. Single Parity Check Product Code $(3,2)^3$. Truncated union bound, minimum distance decoding, original OSD ($I = 2$), OSD ($I = 2$) with Algorithm C, iterative OSD ($I = 1$) with Algorithm D.

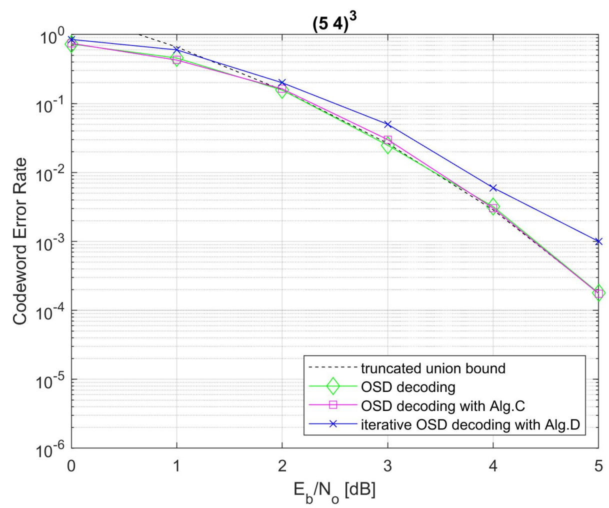

Fig. 7. Single Parity Check Product Code $(5,4)^3$. Truncated union bound, original OSD ($I = 2$), OSD ($I = 2$) with Algorithm C, iterative OSD ($I = 1$) with Algorithm D.

Also in this case we would like to delete a number of bits that will certainly not be part of the Most Reliable Basis. For a 3D product code, all rows, all columns, all vectors along the third dimension are parity check equations. Then we can analyze them in parallel and cancel their least reliable bit. This suggests the following algorithm.

Algorithm C. Given a 3D $(n,k)^3$ code:

- Consider all $3n^2$ n-bit vectors in the three dimensions. Process them in parallel and delete the least reliable bit of each n-bit vector (we may have collisions, i.e., some of the deleted bits can be the least reliable for 2 or 3 vectors they belong). Let us denote by $n \leq \alpha \leq 3n^3 - 3n + 1$ the number of deleted bits.
- Build the Most Reliable Basis working on the survived

$n^3 - \alpha$ bits.

- Apply the order 2 OSD on this Most Reliable Basis.

Results for the application of Algorithm C are shown in Fig. 7, where we compare

- The truncated union bound, computed by equation (5).
- The original OSD algorithm with order $I = 2$ applied on the optimal Most Reliable Basis.
- The OSD algorithm with order $I = 2$ applied on the Reliable Basis obtained with Algorithm C.

(The last curve will be explained in the next section.) By looking at Fig.7 we can see that both the original OSD and the OSD working with Algorithm C achieve nearly-ML performance.

VII. Iterative OSD for 3D SPC product codes

In this section, as a further idea to reduce OSD complexity, we focus on 2D subcodes of 3D product codes. For these subcodes we have $d_{\min} \leq 4$, then order $I_{\min} = 1$ is enough and the number of codewords to be tested is very small. Moreover, we can implement multiple simple SISO OSD copies working in parallel, iteratively exchanging their soft information.

The starting point is to have an efficient, simple SISO OSD algorithm for 2D product codes. To this purpose, we have adapted the original version of [2] by exploiting SPC product code structure and the SRB construction of Algorithm B. This simplified algorithm follows.

Algorithm D. Given a 2D codeword:

- Identify the Simplified Reliable Basis by using Alg. B.
- Apply order-1 OSD by testing $1 + k^2$ candidate codewords.
- For each of the n^2 bits c_j, when the $1 + k^2$ candidate codewords are evaluated, keep memory of the most probable one for the two values $c_j = 1$ and $c_j = 0$.
- Release the LLR for each of the n^2 bits by using (8).

Note that this approach does not require any order I reprocessing: an order-1 OSD application is sufficient.

Now, we can apply Algorithm D to any of the n 2D faces of a 3D product code, and iteratively exchange the LLR information. The results of the iterative application of Algorithm D to $(3, 2)^3$ and $(5, 4)^3$ 3D product codes have been reported in Fig. 6 and Fig. 7. We can see that there is a penalty with respect to ideal ML decoding, with a larger gap for the second code. Even if sub-optimal, the performance are still quite good despite its simplicity (as an example, if we compare the iterative curve of Fig. 7 with that obtained in [11] for the same code by applying successive cancellation, an improvement of about 0.5 dB is achieved). This suggests that this approach merging low-order (iterative 2D sub-code processing) and low-complexity (simplified reliable basis construction) OSD can be considered for other more powerful codes, too.

VIII. Conclusions and Future Work

In this paper we have presented some preliminary results aiming to simplify Ordered Statistics Decoding, based on two ideas: (i) by exploiting the code structure it is possible to simplify the preliminary phase of OSD (that cannot be parallelized), i.e., the construction of reliable basis and a systematic generator matrix for it, and (ii) if we identify subcodes with distance $d_{\min} \leq 4$, OSD order can be limited to $I_{\min} = 1$ with very few patterns and apply iterative decoding. Some simplified algorithms have been presented, working on both 2D and 3D SPC product codes, able to approach ideal Maximum Likelihood decoding. Future work includes the extension of these techniques to more complex codes, obtained by combining more powerful codes.

References

[1] M. P. C. Fossorier and S. Lin, “Soft-decision decoding of linear block codes based on ordered statistics,” IEEE Trans. on Information Theory, vol. 41, pp. 1379–1396, Sep. 1995.

[2] M. P. C. Fossorier and S. Lin, “Soft-Input Soft-Output Decoding of Linear Block Codes Based on Ordered Statistics,” IEEE Globecom 1998, Nov. 1998.

[3] A. Valembois and M. P. C. Fossorier, “Box and match techniques applied to soft-decision decoding,” IEEE Trans. on Information Theory, vol. 50, pp. 796–810, May 2004.

[4] H. Yagi, T. Matsushima, S. Hirasawa, “Soft-decision decoding using ordered recodings on the most reliable basis”, IEEE Trans. on Information Theory, vol. 53, pp.829—836, 2007.

[5] A. Kabat, F. Guilloud, R. Pyndiah, “New approach to order statistics decoding of long linear block codes”, IEEE GLOBECOM 07, Nov. 2007.

[6] M. Baldi, M. Bianchi, F. Chiaraluce, R. Garello, I. Aguilar Sanchez, S. Cioni, “Advanced channel coding for space mission telecommand links”, 2013 IEEE 78th Vehicular Technology Conference, VTC2013-Fall, Sep. 2013.

[7] M. Baldi, F. Chiaraluce, R. Garello, N. Maturo, I. Aguilar Sanchez, S. Cioni, “Analysis and performance evaluation of new coding options for space telecommand links - Part I: AWGN channels,” International Journal of Satellite Communications and Networking, 2015.

[8] F. Boccardi, R. W. Heath, A. Lozano, T. L. Marzetta, P. Popovski, “Five disruptive technology directions for 5G”, IEEE Commun. Mag., vol. 52 pp 74-–80, 2014.

[9] G. Durisi, T. Koch, P. Popovski, “Toward massive, ultrareliable, and low-latency wireless communication with short packets”, Proc. IEEE, vol. 104 , pp. 1711-–1726, 2016.

[10] M. C. Coşkun, G. Durisi, T. Jerkovits, G. Liva, W. Ryan, B. Stein, F. Steiner, “Efficient Error-Correcting Codes in the Short Blocklength Regime”, Elsevier Physical Communication.

[11] G. Liva, L. Gaudio, T. Ninacs, T. Jerkovits, “Code Design for Short Blocks: A Survey,” ArXiv, 2016.

[12] V. Bioglio, C. Condo, I. Land, “Design of polar codes in 5G new radio,” ArXiv, 2018.

[13] Consultative Committee for Space Data Services. “TC Synchronization and Channel Coding”, Blue Book CCSDS 231.0-B 3, September 2017.

[14] M. Baldi et al., “State-of-the-Art Space Mission Telecommand Receivers,” IEEE Aerospace and Electronic Systems Magazine, vol. 32, pp. 4–15, June 2017.

[15] M. Baldi et al., “Theoretical Analysis and Implementation of Effective Receivers for Telecommand Space Links,” Eight ESA International Workshop on Tracking, Telemetry and Command Systems for Space Applications, TTC2019, 2019.

[16] G. Caire, G. Taricco, G. Battail, “Weight distribution and performance of the iterated product of single parity check codes,” Annales des télécommunications, 1995.

[17] F. Chiaraluce and R. Garello, “Extended Hamming Product Codes Analytical Performance Evaluation for Low Error Rate Applications,” IEEE Trans. on Wireless Communications, vol. 3, pp. 2353–2361, Nov. 2004.