

Row-column parallel turbo decoding of product codes

C. Jégo and P. Adde

A row-column parallel architecture of a turbo decoder dedicated to product codes is presented. This architecture enables simultaneous decoding of the row and the column of a block. The performance of the proposed row-column parallel turbo decoder is similar to that of conventional turbo decoder. However, this new architecture reduces the decoding latency by a factor of two. Moreover, the memory necessary for the block reconstruction between row decoding and column decoding is removed.

Introduction: In telecommunications, forward error correction (FEC) is a system of error control that improves digital communication quality. The most recent development in error correction is turbo coding [1]. The turbo code family refers to two classes of codes: Convolutional Turbo Codes (CTC) and Block Turbo Codes (BTC). Indeed, an iterative Soft Input Soft Output (SISO) decoding was successfully applied [2] to product codes, first introduced by Elias [3]. Product codes are a series of concatenated systematic linear block codes. The product code inherits the properties of the elementary codes that it is composed of. Let us consider an example with two identical BCH (n,k,δ) codes. The parameters of the resulting product code are therefore given by: n^2 (code length), k^2 (number of information symbols), δ^2 (minimum Hamming distance) and r^2 (code rate). The iterative decoding algorithm involves

performing the successive decoding of rows and columns (two half-iterations). Note that each iteration provides decoding operations for all rows and all columns of a matrix (product code). Moreover, a reconstruction of the matrix is necessary between each decoding. The block diagram of an elementary SISO decoder is presented in Figure 1, where k stands for the number of half-iterations.

- ✓ R is the initial word received from the channel,
- ✓ W_k is the pattern that contains the extrinsic information, that is, the additional information given by the decoder concerning the reliability of the decoded symbol,
- ✓ R'_k is the result of extrinsic information weighted by a factor of α_k ,
- ✓ D_k is the result of symbol decoding,
- ✓ α_k and β_k are constants that depend on the current half-iteration.

Previous work: In the last few years, many block turbo decoder architectures have been designed [4]. The conventional approach involves decoding all the rows or all the columns of a matrix before the next half-iteration. When an application requires high-speed decoders, an architectural solution is to cascade SISO elementary decoders for each half-iteration as shown in Figure 2. In this case, memory blocks are necessary between each half-iteration to store channel data and extrinsic information. Each memory block is composed of four memories of qn^2 symbols where q is the number of bits to quantify the matrix symbols. Thus, duplicating a SISO elementary decoder results in duplicating the memory block which is very costly in terms of area. The latency can be defined as the number of symbols that was input into the turbo

decoder before one symbol was fully handled. In the case of cascaded decoders, the architecture latency is equal to $2k*(n^2+n)$ where n is the code length and $2k$ is the iteration number. For one half-iteration, n^2 and n are block memory latency and row or column decoding latency, respectively.

Row-column parallel turbo decoder for product codes: In this letter, a row-column parallel turbo decoding of product code is proposed. Since all the rows and all the columns of the matrix are distinct code words, they can be handled independently. Moreover, the symbol decoding for the rows and the columns does not have any particular order. For these reasons, the rows and the columns of a same matrix can be decoding in parallel. The matrix of extrinsic information is then updated during the decoding process. The aim of the row-column parallel approach is to use the more reliable extrinsic information each time. Let us consider S_i and S_j as the positions of the i -th symbol for a row and the j -th symbol for a column respectively. i and j are included between 1 and n . It can be also noted that the codeword length n of BCH codes is always a power of 2. For row-column parallel turbo decoding, one solution is to take j equal to $n-i$ to avoid simultaneous access to the same symbol by the two SISO decoders. Thus, the row decoder begins with the first symbol S_1 of the row codeword and the column decoder begins with the last symbol S_n of the column codeword, as shown in Figure 3. In this approach, the elementary decoders have to handle only one code word at a time. For this reason, a pipeline architecture cannot be used for the elementary SISO decoder. The architecture of our row-column parallel turbo decoder for one iteration is described in Figure 4. It is composed of two identical elementary

SISO decoders and three memories of qn^2 symbols with double accesses. This structure needs only one memory block per iteration. This enables the memory complexity be reduced by a factor of 8/3. Indeed, two memory blocks per iteration are necessary for the conventional cascaded architecture of a block turbo decoder. Moreover, the proposed parallel architecture is also interesting from a latency point of view. In this case, the architecture latency is equal to $k^*(n^2+n)$, which enables the latency to be reduced by a factor of two.

Performance comparison: Bit-error performance of block turbo codes using extended BCH component codes with single error correction power after 8 iterations is given in Figure 4. We considered an AWGN channel and a system employing BPSK modulation. Simulation results are presented for three BCH components codes with the code lengths being 32, 64 or 128 bits. The performance of the proposed row-column parallel turbo decoding is similar to that of conventional turbo decoding.

Conclusion: In this letter, a row-column parallel architecture of a turbo decoder dedicated to product codes has been presented. Unlike the conventional BTC decoding approach, this architecture uses the more reliable extrinsic information at each step. This reduces the decoding latency by a factor of two, and the memory necessary for the block reconstruction between row decoding and column decoding is removed. It is clear that this innovative proposal provides an attractive solution from the complexity and decoding delay points of view. Simulation results revealed that there is no performance degradation for row-column parallel approach.

References

- 1 BERROU C., GLAVIEUX A., THITIMAJSHIMA P.: 'Near Shannon limit error correcting coding and decoding: Turbo Codes', IEEE International Conference on Communication ICC93, vol. 2/3, May 1993.
- 2 PYNDIAH R., GLAVIEUX A., PICART A., JACQ S: 'Near optimum decoding of product codes', GLOBECOM94, November 1994.
- 3 ELIAS P.: 'Error-free coding', IRE Trans. on Inf. Theory, vol. IT-4, pp. 29-37, September 1954.
- 4 CUEVAS J., ADDE P., KEROUEDAN S., PYNDIAH R.: 'New architecture for high data rate turbo decoding of product codes', GLOBECOM 2002, pp. 139-143, November 2002.

Authors' affiliations:

C. Jégo and P. Adde (GET/ENST Bretagne, CNRS TAMCIC UMR 2872, Brest, France)

Figure captions:

Fig. 1 Block diagram of an elementary SISO decoder

Fig. 2 Block diagram of a cascaded block turbo decoder for one iteration

Fig. 3 Principle of parallel row-column decoding dedicated to product codes

Fig. 4 Block diagram of a parallel row-column block turbo decoder for one iteration

Fig. 5 Performance of conventional and parallel decoded BCH BTCs on AWGN channel after 8 iterations

Figure 1

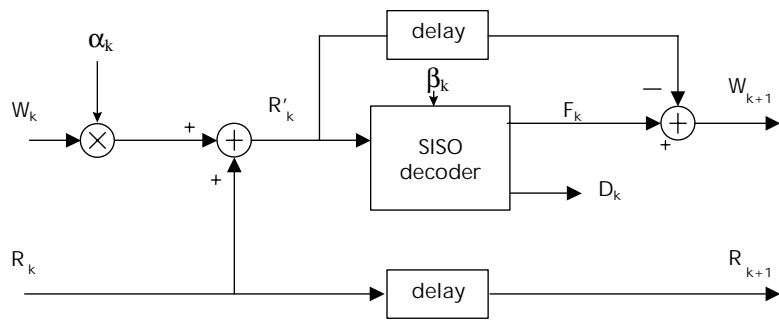


Figure 2

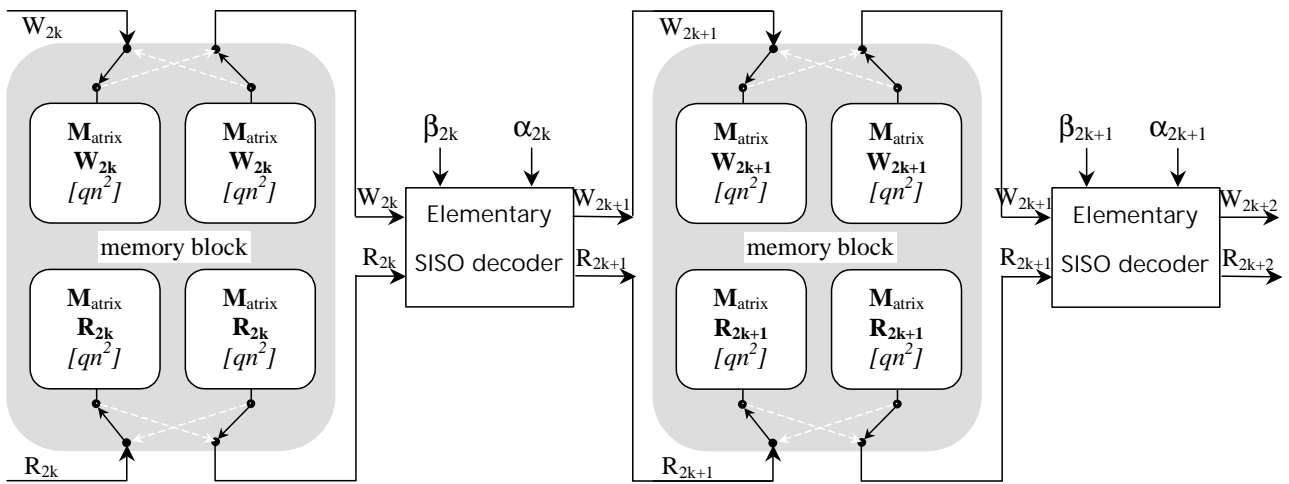


Figure 3

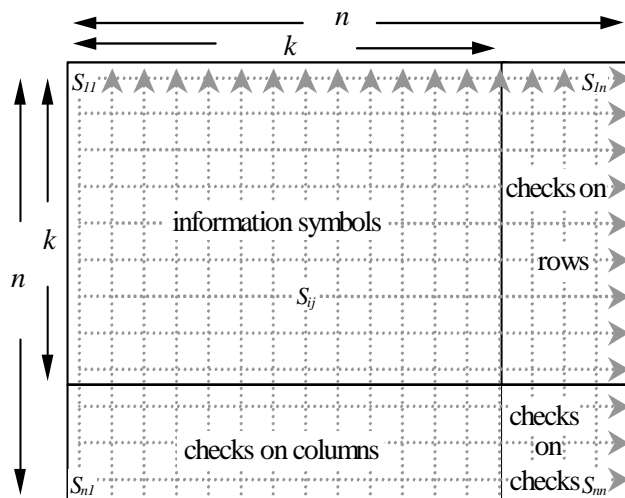


Figure 4

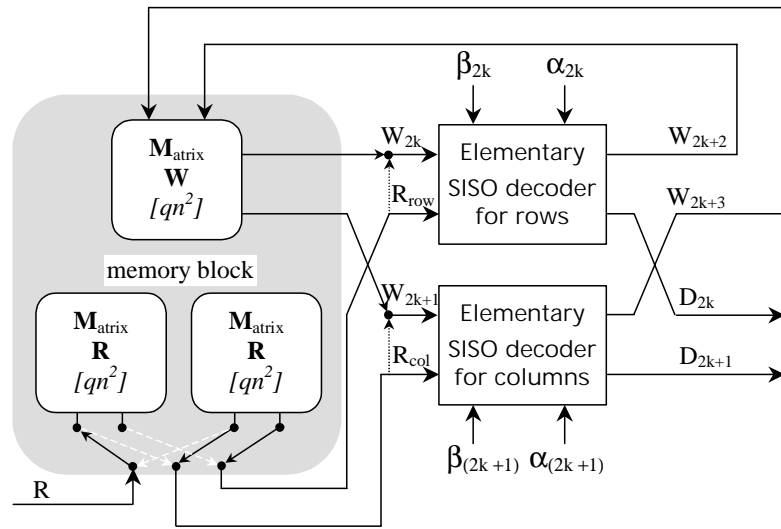


Figure 5

