Haddad *et al. EURASIP Journal on Advances in Signal Processing* 2012, **2012**:131
http://asp.eurasipjournals.com/content/2012/1/131

EURASIP Journal on Advances in Signal Processing
a SpringerOpen Journal

RESEARCH Open Access

Complexity adaptive iterative receiver performing TBICM-ID-SSD

Salim Haddad*, Amer Baghdadi and Michel Jezequel

Abstract

Flexible and iterative baseband receivers with advanced channel codes like turbo codes are widely adopted nowadays, ensuring promising error rate performances. Extension of this principle with an additional iterative feedback loop to the demapping function has proven to provide substantial error performance gain at the cost of increased complexity. However, this complexity overhead constitutes commonly an obstacle for its consideration in real implementations. This article illustrates the opposite of what is commonly assumed and proposes a complexity adaptive iterative receiver performing iterative demapping with turbo decoding (TBICM-ID-SSD). Targeting identical error rate, the article shows that for certain system configurations TBICM-ID-SSD presents lower complexity than TBICM-SSD (without iterative demapping). This original result is obtained when considering the equivalent number of iterations through detailed analysis of the corresponding computational and memory access complexity. The analysis is conducted for different parameters in terms of modulation orders and code rates and independently from the architecture for a fair comparison. Considering the proposed adaptive receiver which is able to perform both TBICM-ID-SSD and TBICM-SSD modes, results demonstrate a reduced complexity with TBICM-SSD for high modulation orders. However, for low modulation orders as for QPSK, results show a reduction in arithmetic operations and read access memory up to 45.9% and 47%, respectively for using the TBICM-ID-SSD mode rather than TBICM-SSD performing six turbo decoding iterations over Rayleigh fading channel with erasures.

Introduction

Advanced wireless communication standards impose the use of modern techniques to improve spectral efficiency and reliability. Among these techniques, bit-interleaved coded modulation (BICM) [1] with different modulation orders and Turbo Codes with various code rates are frequently adopted.

The BICM principle currently represents the state-of-the-art in coded modulations over fading channels. The BICM with iterative demapping (BICM-ID) scheme proposed in [2] is based on BICM with additional soft feedback from the soft-input soft-output (SISO) convolutional decoder to the constellation demapper. In [3], the convolutional code classically used in BICM-ID schemes was replaced by a turbo code. Only a small gain of 0.1 dB was observed. This result makes BICM-ID with turbo-like coding solutions (TBICM-ID) unsatisfactory with respect to the added decoding complexity.

On the other hand, signal space diversity (SSD) technique, which consists of a rotation of the constellation followed by a signal space component interleaving, has been recently proposed [4,5]. It increases the diversity order of a communication system without using extra bandwidth.

Combining SSD technique with TBICM-ID at the receiver side has shown excellent error rate performance results particularly in severe channel conditions (erasure, multi-path, real fading models) [6,7]. These results were behind the adoption of this system in DVB-T2 standard (using LDPC channel code). These results will also lead for further adoption discussions in the upcoming standards using turbo codes [6]. The TBICM and TBICM-ID modes applying the SSD technique are denoted by TBICM-SSD and TBICM-ID-SSD.

In fact, almost all related works using these techniques have focused only on error rate performance without considering the implementation perspective. This is due mainly to the commonly assumed impact in terms of complexity overhead. In this article, we demonstrate the

*Correspondence: salim.haddad@telecom-bretagne.eu
[1] Institut Mines-Telecom; Telecom Bretagne; UMR CNRS 3192 Lab-STICC Electronics Department, Technopôle Brest Iroise CS 83818, 29238 Brest, France

effectiveness of the iterative demapping even in terms of complexity for certain system configurations (modulation orders and code rates). In this context, a novel complexity adaptive iterative receiver, performing either in TBICM-ID-SSD mode or in TBICM-SSD mode, is proposed. This original proposal is based on a thorough analysis of the corresponding computational and memory access complexity.

It is worth to note that the article does not provide a comparison in terms of area, as one iterative receiver is considered to perform both modes (TBICM-ID-SSD and TBICM-SSD).

The rest of the article is organized as follows. Section System model and algorithms presents the system model with the associated parameters and gives a brief description of the underlined algorithms for iterative demapping and turbo decoding. Section Complexity evaluation and normalization presents an evaluation of the receiver complexity in terms of number and type of arithmetic operations and memory access. Section Number of iterations analysis for identical complexity analyzes the number of TBCIM-SSD and TBCIM-ID-SSD iterations for identical complexity. Section Complexity analysis for identical performance shows a complexity analysis for identical TBCIM-SSD and TBCIM-ID-SSD error rate performances. Finally, Section Conclusion concludes the article.

System model and algorithms

This section describes the system model and the considered parameters of the transmitter, channel, and receiver of Figure 1. In addition, it gives a brief presentation of the underlined algorithms for the iterative demapping and decoding.

System model

On the transmitter side, information bits U which are called systematic bits are regrouped into symbols u_i consisting of q bits, and encoded with an q-binary turbo encoder. It consists of a parallel concatenation of two identical convolutional codes (PCCC). The output codeword C is then punctured to reach a desired coding rate R_c. We consider in this work the 8-state double binary ($q = 2$) [8] turbo code adopted in the WiMax standard.

In order to gain resilience against error bursts, the resulting sequence is interleaved using an S-random interleaver Π_2 with $S = \sqrt{N/4}$. Punctured and interleaved bits denoted by v_i are then gray mapped to complex channel symbols s_q chosen from a 2^M-ary constellation X, where M is the number of bits per modulated symbol.

Applying the SSD consists first of the rotation of the mapped symbols s_q. The resulting rotated symbols are denoted as $s_{r,q}$. The performance gain obtained when using a rotated constellation X_r depends on the choice of the rotation angle. The optimum rotation angle depends

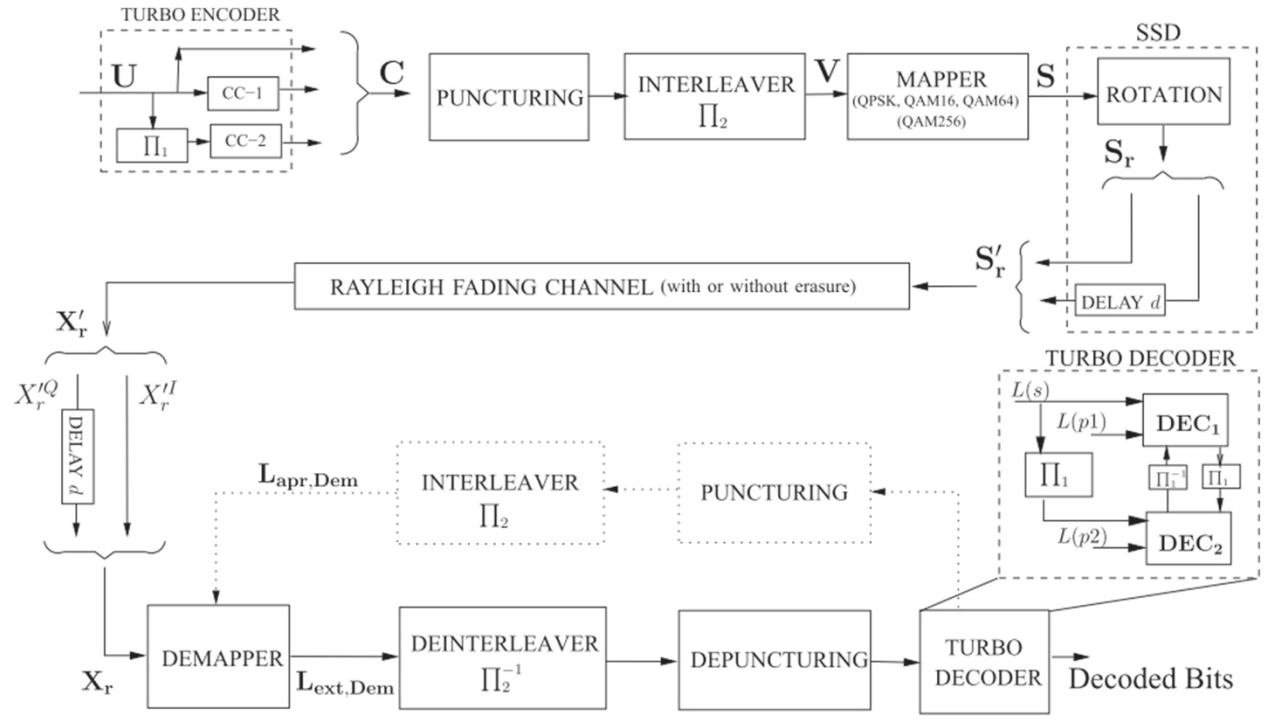

Figure 1 The system model of the transmitter, channel, and TBICM-ID-SSD receiver.

on the chosen modulation and channel type. In this regard, a thorough analysis has been done for the 2nd-generation terrestrial transmission system developed by the DVB Project (DVB-T2) which adopted the rotated constellation technique. A single rotation angle [7] has been chosen for each constellation size independently of the channel type. These angle values are presented in Table 1 and are adopted in this work.

The second step when applying SSD at the transmitter consists of signal space component interleaving. A simple delay is introduced between the transmission of I and Q components. Mapped and shifted symbols $s'_{r,q}$ are then transmitted over a noisy and Rayleigh fast fading channel with or without erasure. The erasure channel model has been used in the case of the DVB-T2 standard to model the destructive interferences caused by the existence of a single-frequency network (SFN). Each received symbol $x'_{r,q}$ is affected by a different fading coefficient, an erasure coefficient, and an additive Gaussian noise.

The channel model considered is a frequency non-selective memoryless channel with erasure probability. The received discrete time baseband complex signal can be written as:

$$\begin{aligned} x'_{r,q} &= h_q.\rho_q.s'_{r,q} + n_q \\ &= h'_q.s'_{r,q} + n_q \end{aligned} \tag{1}$$

where h_q is the Rayleigh fast fading coefficient, ρ_q is the erasure coefficient probability taking value 0 with a probability P_ρ and value 1 with a probability of $1 - P_\rho$. n_q is a complex white Gaussian noise with spectral density $N_0/2$ in each component axes, and h'_q is the channel attenuation.

Max-log-map demapping algorithm

At the receiver side, the complex received symbols $x'_{r,q}$ have their Q-components re-shifted resulting in $x_{r,q}$. An extrinsic log-likelihood ratio $L_{\text{ext,Dem}}(c_{k,q}/x_{r,q})$ is calculated for each bit $c_{k,q}$ corresponding to the kth bit of the received rotated and modulated symbol $x_{r,q}$. After de-interleaving, de-puncturing and turbo decoding, extrinsic information from the turbo decoder $L_{\text{ext,Dec}}(c_{k,q})$ is passed through the interleaver, punctured and fed back as *a priori* information $L_{\text{apr,Dem}}(c_{k,q})$ to the demapper in a turbo demapping scheme. The extrinsic information $L_{\text{ext,Dem}}(c_{k,q}/x_{r,q})$ is the difference between the soft output *a posteriori* $L_{\text{Dem}}(c_{k,q}/x_{r,q})$ and the soft input *a priori* $L_{\text{apr,Dem}}(c_{k,q})$ at the demapper side. It was originally computed in [9] and given by the expression below:

$$\begin{aligned} L_{\text{ext,Dem}}(c_{k,q}/x_{r,q}) &= L_{\text{Dem}}(c_{k,q}/x_{r,q}) - L_{\text{apr,Dem}}(c_{k,q}) \\ &= \log\left(\frac{Z_1}{Z_2}\right) \end{aligned} \tag{2}$$

$Z_{l(l=0,1)}$ can be expressed as:

$$Z_{l(l=0,1)} = \sum_{s_{r,j} \in X^k_{r,l}} e^{-A_q}. \prod_{i=0, i\neq k}^{M-1} P(c_{i,q}) \tag{3}$$

where $X^k_{r,l}$, with $l \in \{0,1\}$, are the symbol sets of the constellation for which symbols have their kth bit equal to l. $P(c_{i,q})$ is the probability of the ith bit of constellation symbol $s_{r,q}$ computed through *a priori* information $L_{\text{apr,Dem}}(c_{k,q})$. Reducing the complexity of the expressions above can be performed by applying the max-log approximation. Thus, Equation (2) can be written as [9]:

$$L_{\text{ext,Dem}}(c_{k,q}/x_{r,q}) = \min_{s_{r,j} \in X^k_{r,0}} (A_q - B_{k,q}) - \min_{s_{r,j} \in X^k_{r,1}} (A_q - B_{k,q}) \tag{4}$$

where

$$A_q = \frac{h'^2_q}{\sigma^2} |x^I_{r,q} - s^I_{r,j}|^2 + \frac{h'^2_{q-1}}{\sigma^2} |x^Q_{r,q} - s^Q_{r,j}|^2 \tag{5}$$

and

$$B_{k,q} = \left(\sum_{i=0, c_{i,q}=1}^{M-1} L_{\text{apr,Dem}}(c_{i,q}) \right) - L_{\text{apr,Dem}}(c_{k,q}) \tag{6}$$

All demapping equations are valid for erasure and no erasure channel. In fact, the channel coefficient h'_q will take into consideration the erasure coefficient given by the channel detector. These simplified expressions exhibit three main computation steps: (a) Euclidean distance computation referred by A_q, (b) *a priori* adder referred by $B_{k,q}$, and (c) minimum finder referred by the *min* operation of Equation (4).

Max-log-MAP decoding algorithm

Following the demapping function at the receiver side, the turbo decoding is applied. The max-log-MAP algorithm [10] is considered for the SISO convolutional decoders. Using input symbols and *a priori* decoding information, each SISO decoder computes extrinsic information. The SISO decoder computes first the branch metrics γ_k. Then it computes the forward α_k and backward β_k metrics between two trellis states s and s'. Max-log-MAP decoding

Table 1 Rotation angle values in DVB-T2, adopted in this work

Modulation	Rotation angle (degrees)
QPSK	29
16-QAM	16.8
64-QAM	8.6
256-QAM	3.6

equations, originally proposed in [10], are expressed as follows:

$$\alpha_k(s) = \max_{(s',s)}(\alpha_{k-1}(s') + \gamma_k(s',s)) \quad (7)$$

$$\beta_k(s) = \max_{(s',s)}(\beta_{k+1}(s') + \gamma_{k+1}(s',s)) \quad (8)$$

where

$$\gamma_k(s',s) = \gamma_k^{\text{Sys}}(s',s) + \gamma_k^{\text{Parity}}(s',s) + \gamma_k^{Ext}(s',s) \quad (9)$$

Finally the soft output $so(d_k = i)$ and extrinsic information $z(d_k = i)$ of the kth coded symbol are computed [10]:

$$\text{so}(d_k = i) = \max_{(s',s)/d(s',s)=i}\left(\alpha_{k-1}(s') + \gamma_k(s',s) + \beta_k(s)\right) \quad (10)$$

$$z(d_k = i) = \text{SF}.\max_{(s',s)/d(s',s)=i}\left(\alpha_{k-1}(s') + \gamma_k^{Ext}(s',s) + \beta_k(s)\right) \quad (11)$$

where SF is the constant scale factor for the Max-Log-MAP decoding algorithm.

In case of iterative demapping and only by one SISO decoder, the bit-level extrinsic information of systematic symbols $d_k = c_p c_{p+1}$ are computed using (12) and (13). Similar computations are needed for parity symbols $c_{p+2}c_{p+3}$.

$$L_{\text{apr,Dem}}(c_p) = \max[z(d_k = 11), z(d_k = 10)] - \max[z(d_k = 01), z(d_k = 00)] \quad (12)$$

$$L_{\text{apr,Dem}}(c_{p+1}) = \max[z(d_k = 11), z(d_k = 01)] - \max[z(d_k = 10), z(d_k = 00)] \quad (13)$$

These expressions exhibit three main computation steps: (a) branch metrics computation referred by γ_k, (b) state metrics computation referred by (α_k and β_k), and (c) extrinsic information computation referred by $L_{\text{apr,Dem}}$ and z.

Complexity evaluation and normalization

In order to appreciate the complexity of the two iterative modes, an accurate evaluation of the complexity in terms of number and type of operations and memory accesses is required. In this section, we consider the two main blocks of the TBICM-SSD and TBICM-ID-SSD system configurations which are the SISO demapper and the SISO decoder. The proposed evaluation considers the low complexity algorithms presented in Section System model and algorithms.

This evaluation will be presented independently from the architecture (serial, or shuffle, or parallel architecture). It will be based on counting operations without quoting prior SISO demapper and decoder implementation results. A typical fixed-point representation of channel inputs and various metrics is considered. Table 2 summarizes the total number of required quantization bits for each parameter.

Table 2 Total number of required quantization bits for each parameter of the Max-Log-MAP algorithms

Parameter	Number of bits
SISO demapper	
Received complex input $(x^I_{r,q}, x^Q_{r,q})$	(10,10)
Coeff. fading & variance $(h'_q)/(\sigma)$	8
Constellation complex symbol $(s^I_{r,j}, s^Q_{r,j})$	(8,8)
Euclidean distance A_q	19
SISO decoder	
Received 4 LLRs	4 × 5
Branch metric γ_k	10
State metric α_k, β_k	10
Extrinsic information z	10

Complexity evaluation of SISO demapper

The complexity of SISO demapping depends on the modulation order (in the context of the above fixed parameters). In fact, for each received modulated symbol $x_{r,q}$ composed of M coded bits, 2^M Euclidean distances are computed. With iterative demapping, *a priori* information coming from the decoder should be added to the associated Euclidean distance. The minimum distance finder is then applied to search for the closest symbol between the 2^M constellation symbols. Thus, the SISO demapper complexity is composed of three principal units: Euclidean distance, *a priori* adder, and minimum finder functions. For each of these functions we will now consider the equations of Section Max-log-map demapping algorithm (1) the required number and type of arithmetic computations and (2) the required number of read memory access (*load*) and write memory access (*store*). The result of this evaluation is summarized in Table 3 and explained below. We use the following notation *operation*(NbOfBitsOfOperand1,NbOfBitsOfOperand2) for arithmetic operations, and *load*(NbOfBits)/*store*(NbOfBits) for read/write memory operations. Thus, *add*(8,10) indicates an addition operation of two operands; one quantized on 8 bits and the second on 10 bits. Similarly, *load*(8) indicates a read access memory of 8-bit word length.

(1) **Euclidean distance computation**

For each modulated symbol (input of the demapper):

- One load(8) to access the fading channel coefficient normalized by the channel variance $\frac{h'_q}{\sigma}$
- Two load(10) to access the channel symbols $x^I_{r,q}$ and $x^Q_{r,q}$.

Table 3 Complexity evaluation of the SISO demapper and SISO decoder in terms of number and type of arithmetic computations and memory access

SISO rotated demapper with a priori input	
Computation units	Number and type of operations per modulated symbol per demapping iteration
Euclidean distance	$2^M\text{Add}(18,18) + 2^{M+1}\text{Sub}(8,10) + 2^{M+1}\text{Mul}(18,18) + 2^{M+1}\text{Mul}(8,10) + 2\text{load}(10) + (1+2^{M+1})\text{load}(8)$
A priori adder	$(2^M - 2)\{E[\frac{M-1}{2}]\,\text{Add}(8,8) + E[\frac{M-1}{4}]\,\text{Add}(9,9) + E[\frac{M-1}{8}]\,\text{Add}(10,10) + M\text{Sub}(8,11) + M\text{Sub}(11,19)\} + M\text{load}(8) + (2^M-2)\text{load}(M)$
	For QPSK $M(2^M-2)\text{Sub}(11,19) + M\text{load}(8) + (2^M-2)\text{load}(M)$
Minimum finder	$M\text{Sub}(8,8) + M.2^M\text{Sub}(19,19) + M\text{store}(8)$
SISO double binary turbo decoder	
Computation units	Number and type of operations per coded symbol per turbo decoding iteration
Branch metric	$4\text{Add}(5,5) + 38\text{Add}(5,10) + 4\text{Sub}(5,5) + 8\text{load}(5) + 6\text{load}(10)$
State metric	$64\text{Add}(10,10) + 48\text{Sub}(9,9) + 8\text{store}(10)$
Extrinsic information	$32\text{Add}(10,10) + 32\text{Sub}(9,9) + 9\text{Sub}(10,10) + 3\text{Mul}(4,10) + 8\text{load}(10) + 5\text{store}(10)$

- For each one of the 2^M symbols of the constellation $(s^I_{r,j}, s^Q_{r,j})$:
  - Two load(8) to access the constellation symbols $s^I_{r,j}$ and $s^Q_{r,j}$
  - Two Sub(8,10) to compute $(x^I_{r,q} - s^I_{r,j})$ and $(x^Q_{r,q} - s^Q_{r,j})$
  - Two Mul(8,10) to multiply with the channel coefficients $\frac{h'_q}{\sigma}$ and $\frac{h'_{q-1}}{\sigma}$
  - Two Mul(18,18) to compute the square of the results above
  - One Add(18,18) to realize the sum of the two Euclidean distance terms

(2) ***A priori* adder**

For each modulated symbol (input of the demapper):

- M load(8) to access the *a priori* information $L_{\text{apr,Dem}}(c_{i,q})$
- For each one of the 2^M symbols of the constellation $(s^I_{r,j}, s^Q_{r,j})$, except two symbols corresponding to all zeros and all ones:
  - One load(M) to access constellation symbol bits $c_{i,q}$. $i = 0, 1, \ldots, M-1$
  - One addition of M *a priori* information to compute $\sum_{i=0,c_{i,q}=1}^{M-1} L_{\text{apr,Dem}}(c_{i,q})$ of Equation (6). $L_{\text{apr,Dem}}(c_{i,q})$ are quantized on 8 bits as shown in Table 2. This addition of M operands is equivalent to the sum of the following 2-input addition operations:
    - $E[\frac{M-1}{2}]$ Add(8,8) to realize the sum of the couples of $L_{\text{apr,Dem}}(c_{i,q})$. Results are quantized on 9 bits.
    - $E[\frac{M-1}{4}]$ Add(9,9) to realize the sum of the couples of the results above. Results are quantized on 10 bits.
    - $E[\frac{M-1}{8}]$ Add(10,10) to realize the final 2-input addition of the results above. Note that $E[\frac{M-1}{8}]$ equals 0 except for QAM64 and QAM256 where it is equal to 1. The result is quantized on 11 bits. $E[x]$ represents here the ordinary rounding of the positive number x to the nearest integer. Taking the example of QAM16 ($M = 4$), $E[\frac{M-1}{2}]$=2, $E[\frac{M-1}{4}]$=1, $E[\frac{M-1}{8}]$=0.
  - M Sub(8,11) to subtract the LLR of the specific kth bit and thus obtain $B_{k,q}$
  - M Sub(11,19) to realize $A_q - B_{k,q}$

However, for the simple QPSK modulation the above operations can be simplified as only two LLRs exist for one modulated symbol. In fact, in Equation (6) there is no need to execute an addition followed by a subtraction of the same LLR. Thus, the total number of required arithmetic operations in this case is 4 Sub(11,19).

(3) **Minimum finder**

For each one of the M bits per modulated symbol:

- 2^M Sub(19,19) to realize the two min operations of Equation (4)
- One Sub(8,8) to subtract the above found two minimum values
- One store(8) to store the extrinsic information value

Complexity evaluation of SISO decoder

The SISO decoder complexity is composed of three principal units: branch metric, state metric, and extrinsic information functions. As for the SISO demapper, the result of the complexity evaluation is summarized in Table 3 and explained below. As stated before, the considered turbo code is an 8-state double binary one. At the turbo decoder side, each double binary symbol should be decoded to take a decision over the four possible values $(00, 01, 10, 11)$.

(1) **Branch metrics (γ)**
For each coded symbol (input of the decoder):

- 4 load(5) to access systematic and parity LLRs
- 3 load(10) to access demapper normalized extrinsic informations
- 2 Add(5,5) and 2 Sub(5,5) to compute systematic and parity branch metrics γ_{11}^{Sys}, γ_{10}^{Sys}, $\gamma_{11}^{\text{Parity}}$ and $\gamma_{10}^{\text{Parity}}$
- 19 Add(5,10) to compute branch metrics γ_k and $\gamma_k^{\text{Sys}} + \gamma_k^{\text{Parity}}$

Operations above should be multiplied by 2 to generate forward and backward branch metrics.

(2) **State metrics (α,β)**
For each coded symbol (input of the decoder):

- 32 Add(10,10) to compute $\alpha_{k-1}(s') + \gamma_k(s', s)$ for the 32 trellis transitions (8-state double binary trellis)
- 24 Sub(9,9) to realize the 8 max (4-input) operations of Equation (7). In fact, finding the maximum of N values can be implemented as N-1 max (2-input) operations
- 8 store(10) to store computed state metrics only for left butterfly algorithm

Operations above should be multiplied by 2 to generate forward α and backward β state metrics.

(3) **Extrinsic information (z)**
For each coded symbol (input of the decoder):

- 8 load(10) to access state metric values
- 32 Add(10,10) to compute the second required addition operation in Equation (10) for the 32 trellis transitions
- 28 Sub(9,9) to realize the 4 max (8-input) operations of Equation (10)
- 4 Sub(10,10) to subtract symbol-level intrinsic information from the computed soft value (generating symbol-level extrinsic information)
- 8 Sub(9,9) and 4 Sub(10,10) to realize the 8 max (2-input) operations and compute 4 bit-level (systematic and parity) extrinsic information as demapper *a priori* information (Equations (12) and (13)). This computation is done only for one of the two SISO decoders
- 4 store(10) to store the computed bit-level (systematic and parity) extrinsic information
- 3 Sub(10,10) to normalize symbol-level extrinsic information by subtracting the one related to decision 00
- 3 Mul(4,10) to multiply the symbol-level extrinsic information by a scaling factor SF
- 3 store(10) to store the computed $\texttt{DEC}_1$ symbol-level extrinsic information as $\texttt{DEC}_2$ *a priori* symbol-level information

Complexity normalization

A fair comparison between the two modes (TBICM-SSD and TBICM-ID-SSD) requires arithmetic and memory access operations normalization. For arithmetic operations, normalization has been done in terms of 2-input one bit full adders (Add(1,1)). Each one of the adders, subtractors, and multipliers can be converted into equivalent number of Add(1,1). For adders and subtractors, bit-to-bit half and full adders are used and generalized for operand sizes n_1 and n_2. Obtained formulas are summarized in Table 4 with simple, yet accurate, analysis of all corner cases. Similarly, multiplication operations are normalized using successive addition operations. Memory access operation of m word of size n are normalized to one memory access operation of $m \times n$ bits.

Applying the proposed complexity normalization approach to Table 3 leads to the results shown in Table 5. This table summarizes the number of normalized operations required to process one modulated and one coded symbol per iteration for all the functional units of the SISO demapper and SISO decoder. Using this table, it becomes possible to compare the complexity of these heterogeneous components. As an example, the processing of one symbol by the SISO decoder incurs a complexity equivalent to 2104 Add(1,1), load(180) and store(130) operations per iteration. On the other hand, and considering a QPSK configuration ($M = 2$), the complexity of the SISO demapper per modulated symbol per

Table 4 Normalization of basic arithmetic operations in terms of *Add*(1,1) when $n_2 > n_1$

Arithmetic operations	Normalized arithmetic operations
1 Add(n_1, n_2)	$0.5 \times (n_1 + n_2 - 1)$ Add$(1, 1)$
1 Sub(n_1, n_2)	$0.5 \times (n_1 + n_2)$ Add$(1, 1)$
1 Mul(n_1, n_2)	$[(n_1 - 1)(n_2 - 1) + 1 - 0.5 \times n_1]Add(1, 1)$

Table 5 Complexity evaluation of the SISO demapper and SISO decoder in terms of number and type of arithmetic computations and memory access after normalization

SISO rotated demapper with *a priori* input	
Computation units	Number and type of operations per modulated symbol per demapping iteration
Euclidean distance	$358.75 \times 2^{M+1}\text{Add}(1,1) + \text{load}(28 + 2^{M+4})$
A priori adder	$(2^M - 2)\{7.5E[\frac{M-1}{2}] + 8.5E[\frac{M-1}{4}] + 9.5E[\frac{M-1}{8}] + 24.5.M\}\text{Add}(1,1) + \text{load}(8M) + \text{load}(M(2^M - 2))$
	For QPSK $15M(2^M - 2)\text{Add}(1,1) + \text{load}(8.M + M(2^M - 2))$
Minimum finder	$(8 + 19.2^M)M\text{Add}(1,1) + \text{store}(8.M)$
SISO double binary turbo decoder	
Computation units	Number and type of operations per coded symbol per turbo decoding iteration
Branch metric	$304\text{Add}(1,1) + \text{load}(100)$
State metric	$1040\text{Add}(1,1) + \text{store}(80)$
Extrinsic information	$760\text{Add}(1,1) + \text{load}(80) + \text{store}(50)$

iteration is equivalent to 1470 Add(1,1), load(116) and store(16) operations. This table will thus enable us in the following sections to compute and to compare the overall complexity of the TBICM-SSD and TBICM-ID-SSD systems. It is worth noting from this table how the complexity of SISO demapping depends on the modulation order M while that of SISO decoding is independent from the system configuration.

Number of iterations analysis for identical complexity

This section discusses and analyzes the complexity of the two iterative modes at different modulation orders and code rates. The first subsection defines the complexity of each mode, while the second subsection analyzes the required number of iterations assuming identical complexity.

TBICM-SSD and TBICM-ID-SSD complexity definition

If the TBICM-SSD mode requires x iterations to process a frame composed of N_{MSymb} modulated symbols (equivalent to N_{CSymb} coded symbol), the complexity C_1 for TBICM-SSD can be calculated as the sum of the complexity of one demapping process and x decoding processes.

$$C_1 = C^{-}_{\text{dem}}(M).N_{\text{MSymb}} + xC_{\text{dec}}.N_{\text{CSymb}} \tag{14}$$

where $C^{-}_{\text{dem}}(M)$ designates the complexity of processing one modulated symbol, which depends on the constellation size, without taking into consideration the *a priori* computation, and C_{dec} designates the complexity of processing one coded symbol.

Regarding the complexity of TBICM-ID-SSD, we consider the work of [11] which proposes an original iteration scheduling by reducing two demapping iterations with reasonable performance loss of less than 0.15 dB for all configurations. The authors have also shown that omitting only one demapping iteration will keep the error rate performance almost identical for number of iterations $y > 3$. This latter scheme is adopted in this work and we denote the required number of iterations by *yIDem_zEIDec*, where z designates the extra decoding iterations.

Thus, the complexity C_2 for TBICM-ID-SSD can be calculated as the sum of the complexity of y demapping processes and $(y + z)$ decoding processes.

$$C_2 = C^{-}_{\text{dem}}(M).N_{\text{MSymb}} + (y-1) \times C^{+}_{\text{dem}}(M).N_{\text{MSymb}} + (y+z)C_{\text{dec}}.N_{\text{CSymb}} \tag{15}$$

where $C^{+}_{\text{dem}}(M)$ designates the complexity of processing one modulated symbol taking into consideration the *a priori* computation.

For the complexity evaluation of C_{dec} and $C_{\text{dem}}(M)$, the low complexity algorithms presented in Section System model and algorithms were thoroughly analyzed.

Considering the code rate R_c and the number of bits per modulated symbol M, the relation between the number of double binary coded symbols (N_{CSymb}) and the corresponding number of modulated symbols (N_{MSymb}) can be written as follows.

$$N_{\text{MSymb}} = \frac{2.N_{\text{CSymb}}}{M.R_c} \tag{16}$$

In addition to the modulation order and the code rate, a third parameter should be considered regarding the iterative demapping implementation choice. In this regard, two configurations should be analyzed. In the first configuration, denoted CASE 1, the Euclidean distances are re-calculated at each demapping iteration. While in the second configuration, denoted CASE 2, the computation of the Euclidean distances are done only once, at the first iteration, then stored and reused in later demapping iterations. Thus, CASE 1 implies higher arithmetic computations, however less memory access, than CASE 2.

Number of iterations for identical complexity

The final objective of this work is to illustrate for which system configuration it is more interesting to use TBICM-ID-SSD rather than TBICM-SSD. This means for which system configuration the complexity of TBICM-ID-SSD becomes lower than TBICM-SSD. Towards this objective, we analyze in this subsection the corresponding number of iterations if both modes have identical complexity. Identical complexity can be expressed as $C_1 = C_2$. Using this equality and replacing C_1 and C_2 by their expressions from equations (14) and (15) lead to the following equation:

$$xC_{\text{dec}}.N_{\text{CSymb}} = (y-1)C^{+}_{\text{dem}}(M).N_{\text{MSymb}} + (y+z)C_{\text{dec}}.N_{\text{CSymb}} \quad (17)$$

This last equation allows to obtain the number of TBICM-ID-SSD iterations $y = y_{\text{Lim}}$ corresponding to identical complexity for both modes. In fact, by replacing N_{MSymb} with equivalent number of N_{CSymb} (as expressed in Equation (16)) and by simplifying, Equation (17) becomes:

$$y_{\text{Lim}} = \frac{(x-z)C_{\text{dec}} + \frac{2}{M.R_c}C^{+}_{\text{dem}}(M)}{C_{\text{dec}} + \frac{2}{M.R_c}C^{+}_{\text{dem}}(M)} \quad (18)$$

This equation can be used to compute individually y_{Lim} for identical arithmetic, identical read memory access or identical write memory access operations.

If we consider $x = 6$, and for different modulation orders and code rates, Table 6 shows the required number of iterations y_{Lim} with no extra decoding iteration ($z = 0$).

y_{Lim} can have positive values as well as negative values. Negative values mean that for the chosen configuration, TBICM-ID-SSD has always a higher complexity than TBICM-SSD. The positive values represent the limits for which performing less demapping iterations will lead to a lower complexity than TBCIM-SSD, and the inverse is true. Hence, it might be possible to perform less y iterations ($y < y_{\text{Lim}}$) with less complexity while having the same error correction capability than TBICM-SSD. In fact, Table 6 shows that this last situation can potentially happen for QPSK and QAM16 configurations where y_{Lim} varies in a higher range (between 2.9 and 5.8) than QAM64 and QAM256 configurations (most y_{Lim} values are around 2 corresponding to identical arithmetic operations). This analysis will be extended in the next section taking into consideration error rate performance simulations.

Complexity analysis for identical performance

The main motivation behind this analysis is to improve the receiver implementation quality by choosing the mode with the less complexity depending on each system configuration. In order to appreciate this study, an accurate evaluation of the complexity in terms of number and type of operations and memory access has been done in Section Number of iterations analysis for identical complexity.

TBICM-ID-SSD iterations, *xIDec* and *yIDem* respectively, iterative processing at the demapper side is shown to provide additional error correction [6]. Thus, for a considered number of x iterations, identical error rate performance results can be reached by using y iterations with $y < x$.

Complexity analysis for a chosen x

Figure 2 shows a BER comparison between the two iterative modes TBICM-SSD and TBICM-ID-SSD for two configurations: (1) QPSK, code rate $\frac{4}{5}$, erasure probability 0.15 and (2) QAM64, code rate $\frac{2}{3}$, non erasure. These parameters are chosen to represent clearly the two sets of curves in the same figure, the same behavior is seen for other configurations. The BER for $x = 6$ iterations and for different configurations, can be seen as the result of $y = 3$ and $y = 4$ iterations for erasure and non erasure channel respectively. However, using results in [11], the complexity of *4IDem* could be reduced to 3*IDem_zEIDec* with $z = 1$.

On the other hand, Table 6 shows that for QPSK modulation, the minimum number of required TBICM-ID-SSD iterations y_{Lim} for all code rates and for identical required

Table 6 Required number of demapping iterations y_{Lim} for different modulation schemes and code rates to achieve identical complexity (arithmetic operations, or read, or write access memory) as for $x = 6$ and $z = 0$

Modulation scheme	CASE1 (with recomputed Euclidean distances)						CASE2 (with stored Euclidean distances)					
	$R_c = 1/2$			$R_c = 6/7$			$R_c = 1/2$			$R_c = 6/7$		
	y_{Lim}			y_{Lim}			y_{Lim}			y_{Lim}		
	Arith	Load	Store	Arith	Load	Store	Arith	Load	Store	Arith	Load	Store
QPSK	4.2	4.4	5.5	4.8	4.9	5.7	5.6	4.2	4.9	5.8	4.8	5.3
QAM16	2.9	3.9	5.5	3.6	4.6	5.7	4.1	3.4	4.4	4.7	4	5
QAM64	1.8	2.8	5.5	2.2	3.4	5.7	2.4	2.2	2.7	2.9	2.8	4
QAM256	1.3	1.7	5.5	1.4	2.1	5.7	1.4	1.5	−2.9	1.7	1.8	0.6

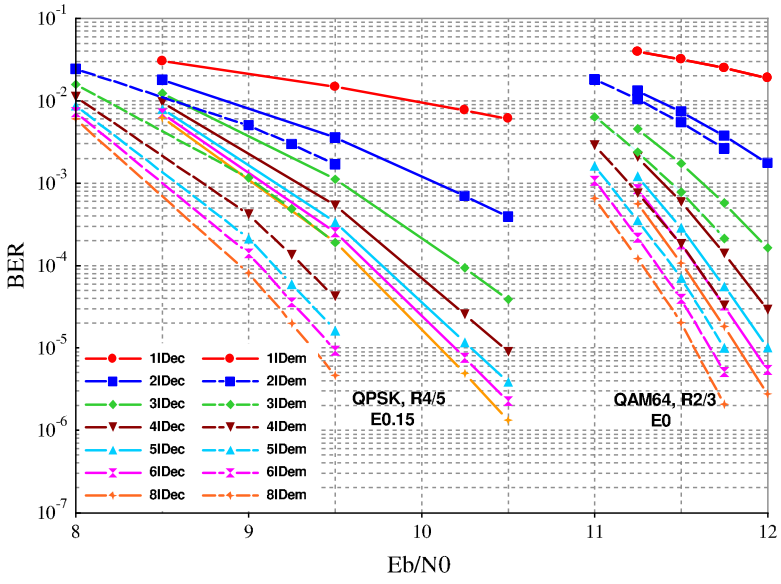

Figure 2 BER performance comparison between TBICM-SSD and TBICM-ID-SSD for the transmission of 1536 information bits frame over Rayleigh fading channel with and without erasure. Different modulation schemes and code rates are considered, for 1 to 8 iterations.

arithmetic operations as *6IDec* is $y_{\text{Lim}} = 4.2$ with $z = 0$. So using $y = 3 < 4.2$ iterations will lead to less arithmetic complexity, meanwhile it has the same error correction capacity as illustrated in Figure 2 for (1).

Complexity improvements have been computed and summarized in Tables 7 and 8. These tables resume the achieved improvements comparing *6IDec* to *3IDem_1EIDec* and *3IDem_0EIDec* respectively for all configurations. In the following we will explain first how these values are computed and then discuss the obtained results.

The complexity reduction ratio (G) is defined as the ratio of the difference in complexity between the two iterative modes to the complexity of TBICM-SSD. It corresponds to the gain ratio of using TBICM-ID-SSD rather than TBICM-SSD. G can be expressed as follows:

$$G = \frac{C_1 - C_2}{C_1} \tag{19}$$

Using this equation and replacing C_1 and C_2 by their expressions from Equations (14) and (15) lead to the following equation:

$$G = \frac{(x - y - z)C_{\text{dec}}.N_{\text{CSymb}} - (y-1)C^{+}_{\text{dem}}(M).N_{\text{MSymb}}}{xC_{\text{dec}}.N_{\text{CSymb}} + C^{-}_{\text{dem}}(M).N_{\text{MSymb}}} \tag{20}$$

By replacing N_{MSymb} with equivalent number of N_{CSymb} (as expressed in Equation (16)) and by simplifying, Equation (20) becomes:

$$G = \frac{(x - y - z)C_{\text{dec}} - \frac{2}{M.R_c}(y-1)C^{+}_{\text{dem}}(M)}{xC_{\text{dec}} + \frac{2}{M.R_c}C^{-}_{\text{dem}}(M)} \tag{21}$$

This last equation has been used to obtain individually the complexity reduction ratios of Tables 7 and 8

Table 7 Achieved reduction values in terms of number of operations, read/write access memory for considering "*3IDem_1EIDec*" rather than "*6IDec*" for different modulation schemes, code rates and no erasure events

Modulation scheme	CASE1 (with recomputed Euclidean distances)						CASE2 (with stored Euclidean distances)					
	$R_c = 1/2$			$R_c = 6/7$			$R_c = 1/2$			$R_c = 6/7$		
	Complexity reduction			Complexity reduction			Complexity reduction			Complexity reduction		
	Arith	Load	Store	Arith	Load	Store	Arith	Load	Store	Arith	Load	Store
QPSK	13.5%	16.8%	28.6%	21.4%	23.5%	30.6%	28%	14%	19.1%	30.1%	21.8%	25%
QAM16	−15.6%	9.3%	28.6%	2.7%	18.9%	30.6%	10.7%	−3.5%	9.5%	19.2%	11.2%	19.3%
QAM64	−89.5%	−22.9%	28.6%	−50.9%	−1.5%	30.6%	−35.8%	−58.6%	−22.3%	−14%	−23.7%	0.6%
QAM256	−207.4%	−108.4%	28.6%	−158.5%	−62.3%	30.6%	−117.9%	−195.6%	−124.1%	−87.2%	−121.1%	−59.3%

Table 8 Achieved reduction values in terms of number of operations, read/write access memory for considering "*3IDem_0EIDec*" rather than "*6IDec*" for different modulation schemes, code rates and erasure events

Modulation scheme	CASE1 (with recomputed Euclidean distances)						CASE2 (with stored Euclidean distances)					
	$R_c = 1/2$			$R_c = 6/7$			$R_c = 1/2$			$R_c = 6/7$		
	Complexity reduction			Complexity reduction			Complexity reduction			Complexity reduction		
	Arith	Load	Store	Arith	Load	Store	Arith	Load	Store	Arith	Load	Store
QPSK	28.9%	32.6%	45%	37.2%	39.6%	47%	43.3%	29.8%	35.4%	45.9%	38%	41.4%
QAM16	−1.6%	24.9%	45%	17.7%	34.9%	47%	24.7%	12.1%	25.9%	34.2%	27.2%	35.8%
QAM64	−78.8%	−8.6%	45%	−38.3%	13.7%	47%	−25.1%	−44.3%	−5.9%	−1.5%	−8.5%	17.1%
QAM256	−201.4%	−97.2%	45%	−150.4%	−49.3%	47%	−111.9%	−184.4%	−107.8%	−79%	−108.1%	−42.8%

in terms of arithmetic, read memory access and write memory access operations. Positive values correspond to a decreasing in complexity, meanwhile negative values correspond to a an increasing in complexity.

In the following, we analyze the values of Table 7 which correspond to a no erasure channel. Similar behavior is seen in Table 8 for erasure channel.

For CASE 1, results show improvements in terms of number of arithmetic operations (up to 21.4%) and read access memory (up to 23.5%) for QPSK scheme. Higher modulation orders require the demapper to fetch symbols from higher constellation memory sizes, which lead to more complexity computations and memory accesses. An increasing in complexity is shown for QAM256 in terms of number of arithmetic operations (−207%) and read access memory (−108%). Moreover, Equation (21) shows that higher the code rate is, higher the benefits are. On the other hand, the improvements in write memory access (28.6% for $R_c = 1/2$ and 30.6% for $R_c = 6/7$) are positive for all modulations orders.

In fact, in the SISO demapper, write memory access is required only to store the extrinsic information which is composed of $M \times 8$ bits. This term is required per modulated symbol and when converted to the equivalent number per coded symbol (Equation (16)) for a fixed code rate, a constant value independent from M is obtained.

Similar behavior is shown for CASE 2, except for two points. The first one concerns the improvements in arithmetic operations and read memory access. In fact, compared to CASE 1, this configuration implies less arithmetic and more memory access operations in the SISO demapper which lead to more benefits for the former operations and less benefits for the latter (Equation (21)). The second point concerns the benefits in write memory access. In fact, besides the term $M \times 8$ bits, a

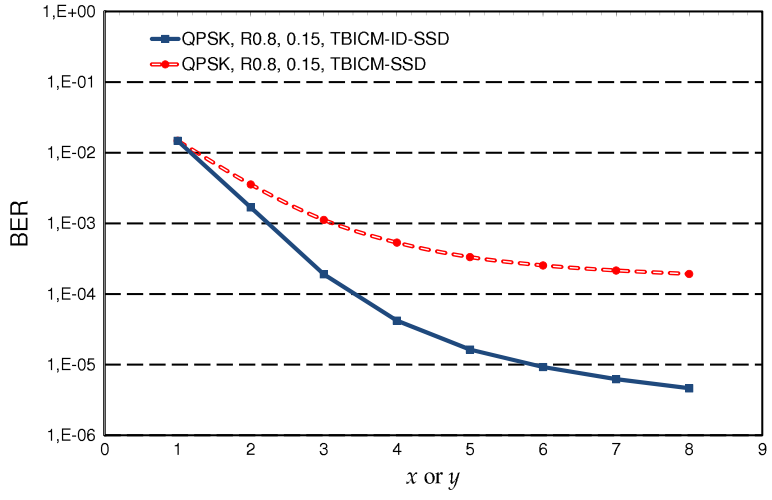

Figure 3 BER performance comparison between TBICM-SSD and TBICM-ID-SSD as a function of number of iterations for the transmission of 1536 information bits frame over Rayleigh fading channel with erasure probability equals to 0.15. QPSK modulation scheme with code rate $\frac{4}{5}$ and E_b/N_0=9.5 dB are considered.

Table 9 Equivalent number of TBICM-ID-SSD iterations *y* for a considered TBICM-SSD *x* iterations to achieve identical BER performances for QPSK, code rate $\frac{4}{5}$ and erasure probability equals to 0.15

x	1	2	3	4	5	6	7	8
y	1	1.7	2.2	2.5	2.75	2.9	2.95	3

value of 19×2^M is required to store the 2^M Euclidean distances quantized on 19 bits each. Therefore the benefits in write access memory operations will be less for high constellation sizes.

Taking an example of QAM64 and code rate $\frac{6}{7}$ for CASE 1 with no erasure. Table 8 shows an increasing in complexity in terms of arithmetic operations (−38.3%), meanwhile positive ratios are seen for read/write access memory. However, it should be noted that the number of required memory access are much less than the arithmetic operations. Thus, those latter are considered as the primary criteria for choosing between the two modes.

We can conclude from the results above that using TBICM-ID-SSD rather than TBICM-SSD for QPSK and QAM16 orders will lead to a significant complexity reduction for almost all code rates.

Finally, as the proposed adaptive iterative receiver targets to reduce the overall normalized processing complexity, this should lead a priori to improved power consumption, throughput and latency. However, analyzing the detailed gains in terms of throughput and latency depends on the heterogeneous architecture and the parallelism degree of the considered demapper and decoder algorithms.

Complexity analysis for different values of *x*

The second part of this study is to look to the gains for different values of x. To that end, and for presentation simplicity, we consider one system configuration which corresponds to QPSK, code rate $\frac{4}{5}$ and erasure probability 0.15.

Figure 3 illustrates the BER performance for both modes as a function of the number of iterations at E_b/N_0=9.5 dB. From this figure, we obtain Table 9 which illustrates the equivalent y iterations for different x values for identical error rate performances.

Using Table 9 and Equation (21), we obtain the complexity reduction curves of Figure 4. Only CASE 2 is considered for presentation simplicity, however the results are similar for CASE 1. The curves of Figure 4 show the variation of the benefits in number of arithmetic operations, read and write memory access as a function of the number of iterations x. In fact, Table 9 shows that for TBICM-SSD number of iterations $x = 1$ the corresponding number of TBICM-ID-SSD iterations $y = 1$. This corresponds to no feedback loop to the demapper, and thus, to identical complexity of the two modes TBICM-SSD and TBICM-ID-SSD. This result is illustrated by Figure 4 where the complexity reduction ratio $G = 0$ for $x = 1$. For $x = 2$, the complexity reduction in terms of arithmetic operations and read access memory is about 10 and 2%, respectively. However, an increased need of write access memory is shown. This is due to the added complexity for storing the 2^M Euclidean distances computed at the first iteration. In fact, the difference in equivalent number of iterations x and y is not big enough to recover this memory write access overhead. However, for $x > 2$, this difference becomes significant and the complexity reduction ratio

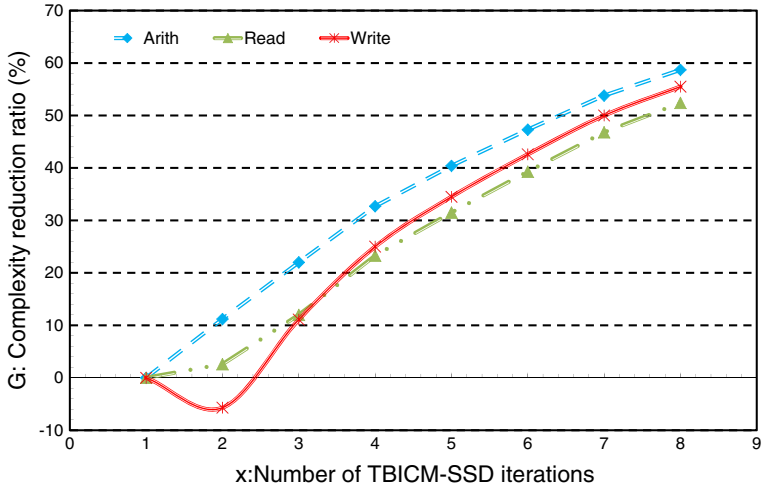

Figure 4 The complexity reduction over iterations for using TBICM-ID-SSD rather than TBICM-SSD for QPSK with code rate $\frac{4}{5}$, erasure probability 0.15, and CASE 2.

increases almost linearly with x to reach between 50 to 60% for $x = 8$. This can be explained from Table 9 where increasing x will increase y but with less speed to attain identical error rate performances.

Conclusion

In this article we have proposed a complexity adaptive iterative receiver performing TBICM-ID-SSD. For low and medium constellation sizes, feedback to the SISO demapper has shown to reduce the complexity in terms of computation and access memory at the receiver side for identical error rate performances. This constitutes a very interesting result as it demonstrates the opposite of what is commonly assumed. In fact, the number of normalized arithmetic operations is reduced in a range between 28.9 and 45.9% for QPSK configuration for using TBICM-ID-SSD rather than TBICM-SSD with 6 iterations over fading channel with erasures. Similarly, the number of read/write access memory is reduced in a range between 29.8% and 47%. This complexity reduction increases significantly for higher turbo decoding iterations and reduces consequently the power consumption of the iterative receiver. On the other had, for high modulation orders, as for QAM64 and QAM256, the TBICM-ID-SSD receiver should be configured in TBICM-SSD mode which provide less complexity for identical error rate performances.

Finally, it is worth to note that for very low error rates, TBICM-ID-SSD configuration should be used as it provides more error correction in the error floor region. Future work targets the extension of this analysis to other baseband iterative applications and its integration into available hardware prototypes.

Received: 28 November 2011 Accepted: 16 May 2012
Published: 3 July 2012

References

1. G Caire, G Taricco, E Biglieri, Bit-interleaved coded modulation. in *IEEE International Symposium on Information Theory*, (1997)
2. X Li, J Ritcey, Bit-interleaved coded modulation with iterative decoding. IEEE Commun. Lett. **1**(6), 169–171 (1997)
3. I Abramovici, S Shamai, On turbo encoded BICM. Ann. des Telecommun. **54**, 225–234 (1999)
4. N Kiyani, J Weber, Iterative demodulation and decoding for rotated MPSK constellations with convolutional coding and signal space diversity. in *IEEE Vehicular Technology Conference, VTC'07-Fall*, 1712–1716, (2007)
5. J Boutros, E Viterbo, Signal space diversity: a power- and bandwidth-efficient diversity technique for the Rayleigh fading channel. IEEE Trans. Inf. Theory. **44**(4), 1453–1467 (1998)
6. C Nour, C Douillard, On lowering the error floor of high order turbo BICM schemes over fading channels. in *IEEE Global Telecommunications Conference, GLOBECOM'06*, 1–5, (2006)
7. CA Nour, C Douillard, Improving BICM performance of QAM constellations for broadcasting applications. in *International Symposium on Turbo Codes and Related Topics*, 55–60, (2008)
8. Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems – Amendment for Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands IEEE Std 80216e/D5-2004,Nov2004
9. S Ten Brink, J Speidel, RH Yan, Iterative demapping and decoding for multilevel modulation. in *Global Telecommunications Conference, 1998. GLOBECOM 98. The Bridge to Global Integration. IEEE, vol. 1*, 579–584, (1998)
10. P Robertson, P Hoeher, E Villebrun, Optimal and sub-optimal maximum a posteriori algorithms suitable for turbo decoding. Eur. Trans. Telecommun. (ETT). **8**(2), 119–125 (1997)
11. S Haddad, A Baghdadi, M Jezequel, Reducing the number of iterations in iterative demodulation with turbo decoding. in *The 19th International Conference on Software, Telecommunications and Computer Networks (SoftCOM 2011)*, (Split and Adriatic Islands. Croatia, 2011)

doi:10.1186/1687-6180-2012-131
Cite this article as: Haddad *et al.*: Complexity adaptive iterative receiver performing TBICM-ID-SSD. *EURASIP Journal on Advances in Signal Processing* 2012 **2012**:131.