Int. J. Communications, Network and System Sciences, 2010, 3, 826-834
doi:10.4236/ijcns.2010.310112 Published Online October 2010 (http://www.SciRP.org/journal/ijcns)

A Turbo Decoder Included in a Multi-User Detector: A Solution to be Retained

Sylvie Kerouédan[1,2], Makram Touzri[1], Patrick Adde[1,2], Samir Saoudi[1,2]
[1]*Institut Télécom, Télécom Bretagne, UMR CNRS* 3192 *Lab-STICC, Brest, France*
[2]*Université Européenne de Bretagne* (*UEB*), *France*
E-mail: *sylvie.kerouedan@telecom-bretagne.eu, mak_fr@yahoo.fr, patrick.adde@telecom-bretagne.eu, samir.saoudi@telecom-bretagne.eu*
Received July 13, 2010; *revised August* 18, 2010; *accepted September* 20, 2010

Abstract

This paper deals with the presentation of different multi-user detectors in the Universal Mobile Telecommunications System (UMTS) context. The challenge is always to optimize the compromise between performance and complexity. Compared with the solution commonly used today, the rake detector, successive interference cancellation (SIC) detector has better performance despite its higher complexity. Our innovative solution proposes joining detector and channel turbo decoder to get a significant gain in terms of performance. Furthermore, when detection and decoding are implemented in a single function, complexity does not increase much.

Keywords: CDMA, Turbocode, Multi-User Detection, Turbo-CDMA

1. Introduction

There is today a high demand for increasing the number of users in wireless communication systems, and sharing techniques have been implemented. When many users have to share the same spectrum resource, multi-user detection (MUD) algorithms have to be implemented in the receiver. Well-known MUD techniques use time, frequency or code division to share resources between users. In our study we focus on one technique: code division multiple access (CDMA). **Figure 1** summarizes the principle of spread spectrum and code division in a CDMA system in order to give some key notations useful for reading the paper.

On the other hand, outstanding channel coding algorithms, such as turbo techniques, can reach very high data rates, or can offer the possibility of low power emission. Joining a MUD receiver and a turbo decoder in an iterative process has been seen as a good way to merge the advantages of the two techniques. This association can be done in different ways:

- separately, which means doing first a few stages of SIC-receiver (Successive Interference Cancellation) and then several iterations of a turbo decoder (SIC-turbo configuration),
- jointly, which means that the turbo decoder is an inner part of the SIC unit (turbo-SIC configuration).

The second proposal is very interesting in terms of performance but seems to be very complex due to the presence of a turbo decoder in the core of the SIC cell. At present time, in the universal mobile telecommunications system uplink context, the solution generally retained in the base station is the classical rake detection followed by a bank of channel decoders. The goal of our study is to show that the detector and turbo decoder association can be competitive against the simplicity of the classical solution. We compare different architectures: the classical rake receiver (CONFIG 1) [1], the SIC-turbo receiver (CONFIG 2) and the turbo-SIC receiver (CONFIG 3) [2]. The three architectures have been described in C language to get performance curves, and then in VHDL to be synthesized with Synopsys Design Analyzer on ST 90 nm target technology to get complexity data. This study and the results have been widely described and justified in [3].

The paper is organized as follows: in Section 2 we describe the implementation of a successive interference cancellation detector; in Section 3, the different associations of channel decoding and multi-user detector are explained; and the last section is dedicated to the comparisons of the different architectures.

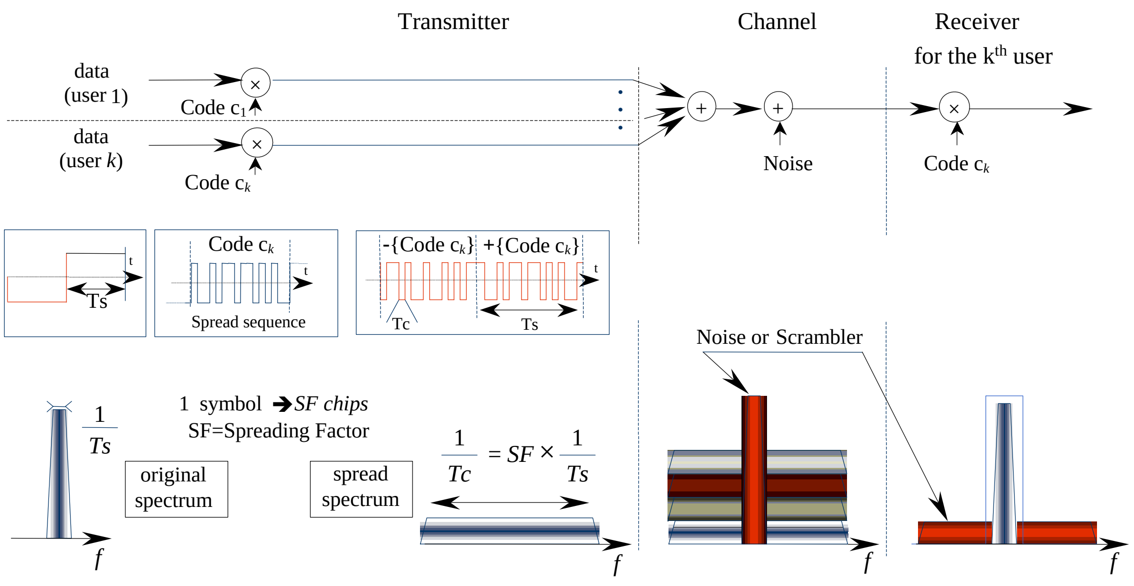


Figure 1. Principle of spread spectrum and code division in CDMA system.

2. Successive Interference Cancellation Detection

2.1. Generalities

In a multi-user context, the goal of interference cancellation is to eliminate interference due to the current user by estimating the transmitted signal and then subtracting it from the received signal. The successive interference cancellation (SIC) detector is based on serial processing of the estimation and the interference cancellation. The SIC detector is a good compromise between performance and complexity compared with parallel or hybrid interference cancellation detectors [4].

SIC structure, shown in **Figure 2**, is composed of *M* steps of *K* interference cancellation units (ICU), where *K* is the total number of users. Inside each ICU, we can find as demonstrated in **Figure 3**:

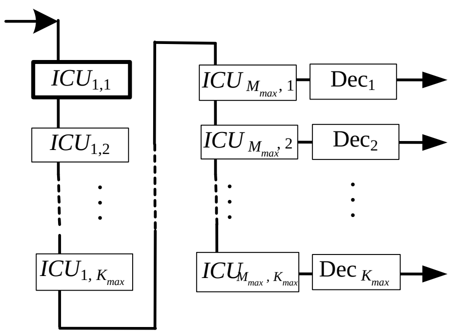


Figure 2. Classical structure of a SIC detector.

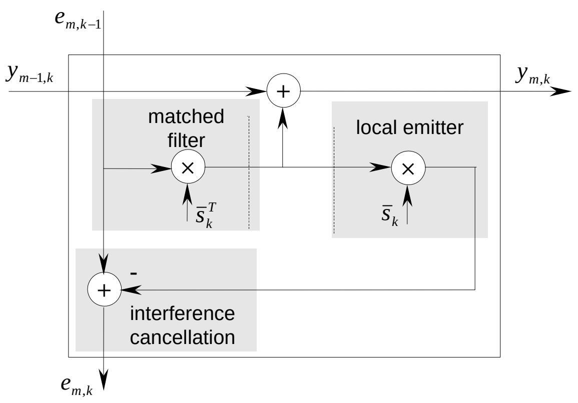


Figure 3. Synoptic of the internal structure of an ICU in the case of a transmission over a Gaussian channel.

- a matched filter linked to the current user *k*,
- a local emitter to regenerate the interference due to the current user,
- an operator which computes the residual signal $e_{m,k}$ after current interference cancellation.

This residual signal is then sent to the following $ICU_{m,k+1}$. The internal structure of an ICU can be more or less complex depending on whether channel decoding is implemented inside or not.

2.2. Implementation of an ICU

In this section, we describe the implementation of the different blocks of the $ICU_{m,k}$ as shown in **Figure 2**. First it is important to notice that the system is clocked either by the chip rhythm (period *Tc*) or the symbol rhythm (period *Ts*) (cf. **Figure 1**).

2.2.1. Matched Filter Block

Figure 4 shows the architecture of the matched filter. The inputs of this block for the current user *k* (*k* = 1, 2, …, *K*) at the step *m* (*m* = 1, 2, …, *M*) are

- the residual signal $e_{m,k-1}$ from previous *k*-1 user,
- the *k* user code (scrambling $s_k^{(S)}$ and spreading $s_{[I]k}^{(w)}$ codes for data link),
- the estimated channel coefficients $c_{k,l}$,
- the delays $t_{k,l}$ coming from the *L* channel paths.

The output of this block is the residual estimation of the received symbol. For each branch the input sequence of SF chips is multiplied by the conjugate of the scrambling code to select the current user *k*. The resulting sequence is then despread (step 2). A multiplication by the channel coefficients corrects the effect of the multiple paths (step 3). The result is then normalized.

The implementation of the block can focus either on delay (combinational architecture) or on surface (sequential architecture). The match filter complexity depends on the number *L* of multiple paths performed during the computation:

- the larger the *L*, the higher the number of gates if *L* branches are implemented;
- the larger the *L*, the slower the circuit if only one branch is implemented.

For our implementation, we choose to use only one branch.

2.2.2. Local Emitter Block

This block delivers a sequence image of the interference of the current user *k* which is part of the residual signal at the input of ICU *k*. Among several architectures, we choose to implement a combinational function. In this solution, described in **Figure 5**, a parallel process sends SF_k (spreading factor of user *k*) chips to the interference cancellation block in order to take into account the multiple paths.

2.2.3. Interference Cancellation Block

This block receives the residual signal coming from the previous user *k*-1 and the image of the interference generated by the local emitter in order to compute the residual signal of the current user *k*. As described in **Figure 6**, SF*max* operators of subtraction are implemented to compute interference cancellation during *L* clock cycles. Thanks to the combinational structure, this block is not very complex.

2.2.4. Time Analysis of the ICU

Depending on architecture choices, the time required to compute a data symbol can be different. In **Figure 7** we give the processing delays if we choose to implement sequential or combinational operators. In the analysis we consider *L* = 6 paths and a sliding window [5] of size 5. Thanks to the sliding window, we can recover the estimation of previously processed symbols to reduce the latency of the process. Thus the required number of clock cycles is 306 to process the interference cancellation.

2.2.5. Complexity of the Logic Glue in the ICU

To evaluate the complexity, we describe the ICU in VHDL and then synthesize the design with Synopsys Design Analyzer. The target technology is ST microelectronics 90 nm. In **Table 1**, we give common parameters.

Figure 8 shows the area of the different part of the

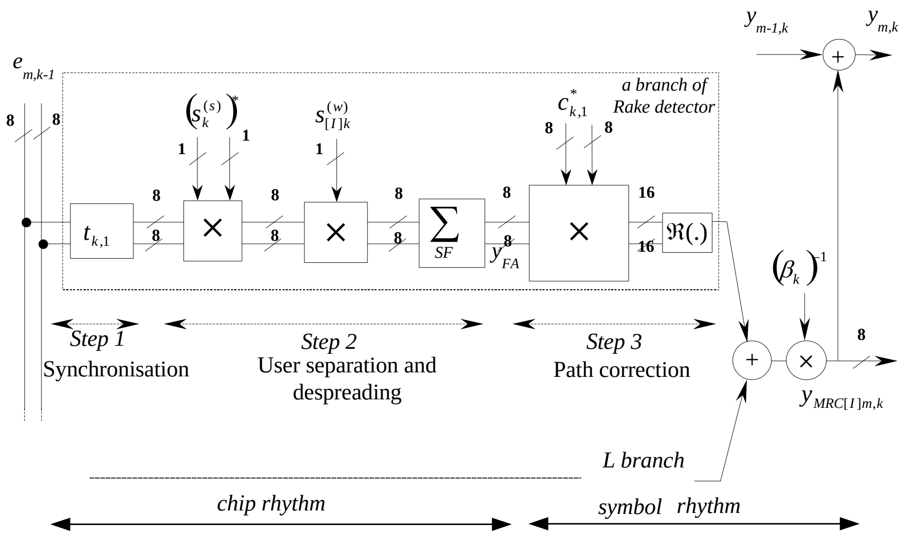


Figure 4. Operators and timing control of the matched filter cell.

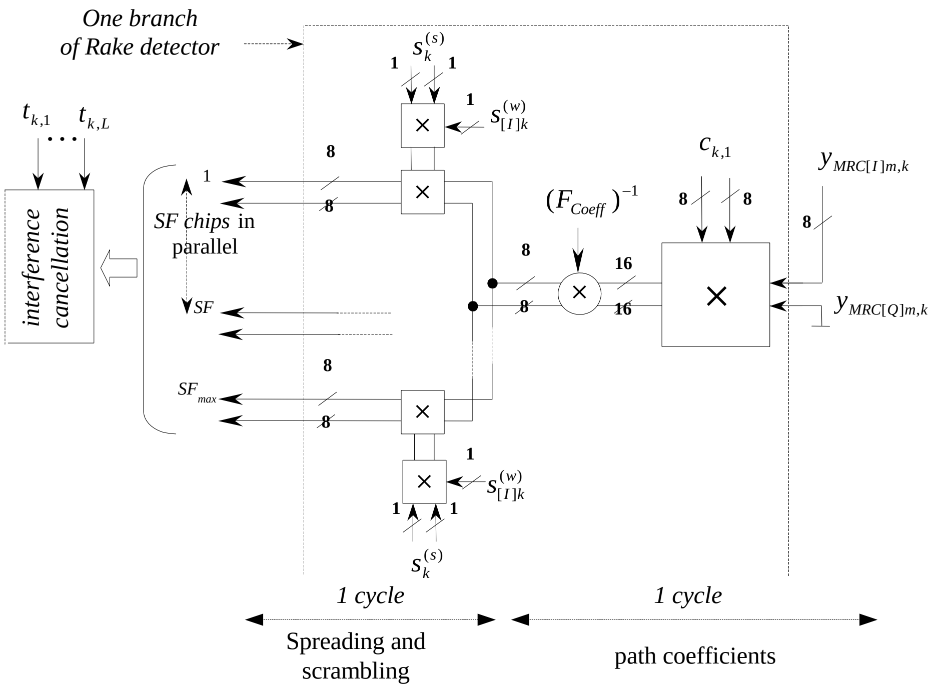


Figure 5. Architecture of the local emitter block.

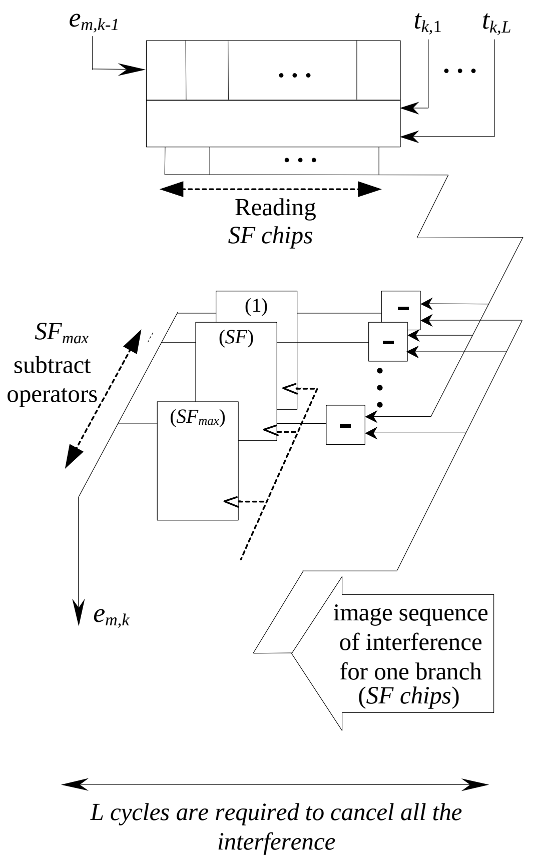

$e_{m,k-1}$
$t_{k,1}$
$t_{k,L}$
Reading SF chips
SF_{max} subtract operators
(1)
(SF)
(SF_{max})
$e_{m,k}$
image sequence of interference for one branch (SF chips)
L cycles are required to cancel all the interference

Figure 6. Architecture of the interference cancellation block in Gaussian channel.

Table 1. Value of parameters for the implementation.

Maximum multiple paths	L*max* = 6
Maximum spreading factor	SF*max* = 16
load factor	100%

ICU, taking into account the choices made for implmentation. The total area of an ICU is then around 15700 gates.

2.2.6. Memory Requirement for the ICU

Each ICU has to exchange data with the previous and following ICUs. As detailed in chapter 3 of [3], there are eight quantization bits for the input signal. RAM cells are required:

- MY_{FA} stores the imaginary part and the real part of the symbol after dispreading. Its size is then 2 × 8 bits.
- MY_{MRC} stores the received symbol correction by channel coefficients (Y_{MRC}). When the SIC detector is followed by a decoder, this memory stores only one symbol, thus its size is 1 × 8 bits.
- $Me_{m,k}$ contains the chips of received signals flowing along the ICU. It is updated during the interference cancellation process. Its size is a function of sliding window length (here 5) and spread factor: 5 × SF × 2 × 8 bits.

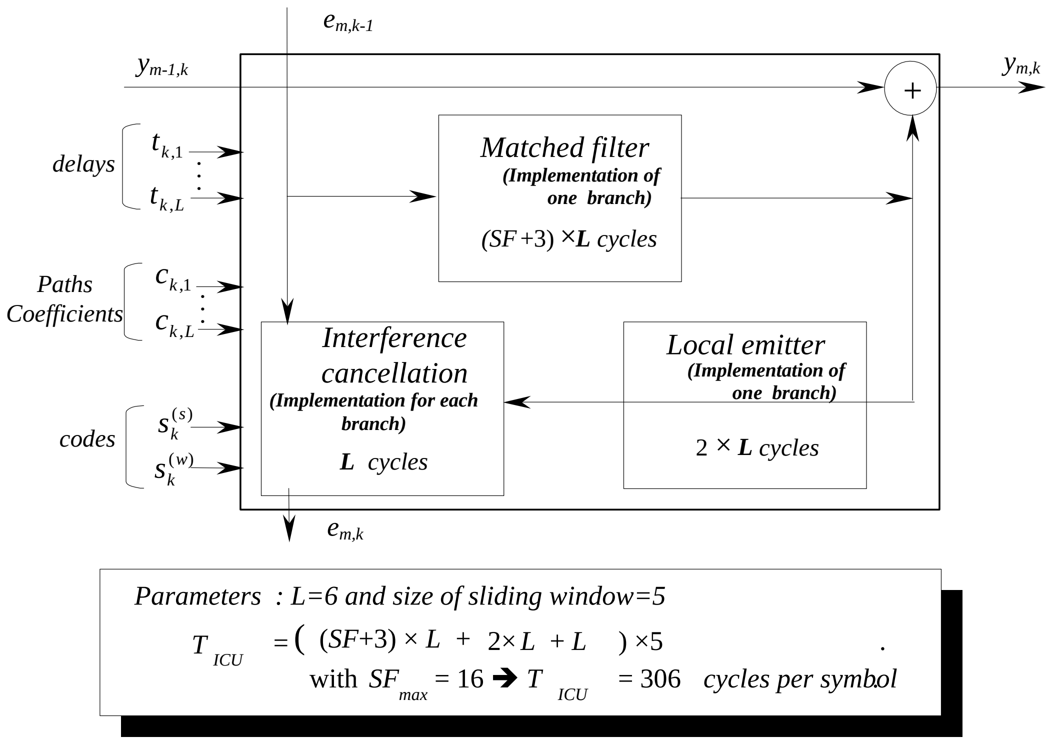

$e_{m,k-1}$
$y_{m-1,k}$
$y_{m,k}$
+
delays
$t_{k,1}$
$t_{k,L}$
Matched filter
(Implementation of one branch)
$(SF+3)\times L$ cycles
Paths Coefficients
$c_{k,1}$
$c_{k,L}$
Interference cancellation
(Implementation for each branch)
L cycles
Local emitter
(Implementation of one branch)
$2\times L$ cycles
codes
$s_k^{(s)}$
$s_k^{(w)}$
$e_{m,k}$
Parameters : L=6 and size of sliding window=5
$T_{ICU} = \left((SF+3)\times L + 2\times L + L \right)\times 5$
with $SF_{max} = 16 \rightarrow T_{ICU} = 306$ cycles per symbol

Figure 7. Timing analysis of an ICU.

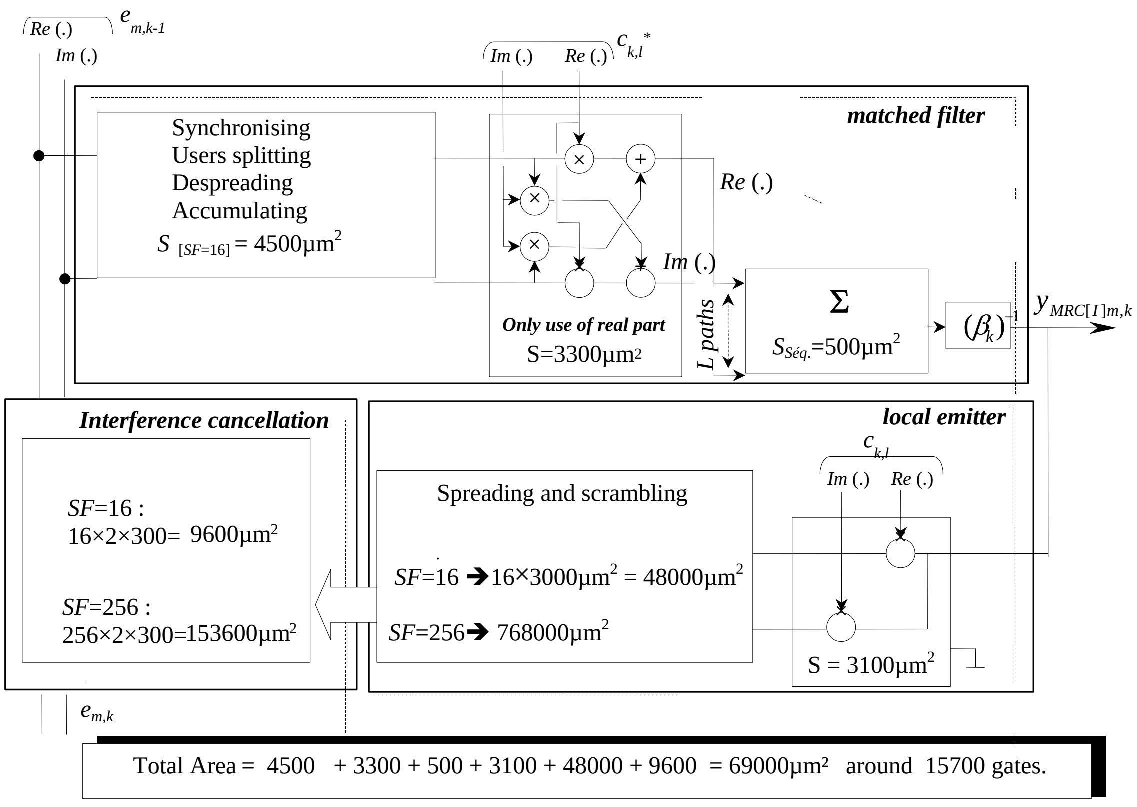


Figure 8. Complexity estimation of an ICU.

- ♦ $MY_{m,k}$ contains the results of the symbol detection for $ICU_{m,k}$. Its size depends on implementation choices. In some configurations, parameter $y_{m,k}$ can be stored through a bus common to the K_{max} stages and an adder. Its size is then $K_{max} \times$ 8 bits.

Sizes of the different RAMs are summed up in **Table 2**.

3. Detection and Channel Decoding

The goal here is to analyze the three different associateions between detection and channel decoding:

- ♦ In the first one, named CONFIG 1, a bank of channel decoders follows a rake detector;
- ♦ In the second one, named CONFIG 2, the channel decoders follow a 3-stage SIC detector;
- ♦ In the third one, named CONFIG 3, an *M*-stage joined SIC detector (M = 2, 3 or 4) and decoder is implemented. That means that the decoder is inside each interference cancellation unit.

As we can see in **Figure 9**, in terms of BER, CONFIG 3 is really more outstanding than CONFIG 1 or CONFIG 2. Now the question is to see whether the complexity increases dramatically or reasonably. That is the goal of this third part.

3.1. Some Words about the Channel Decoding

Nowadays the benefits of channel encoding are well-known: reducing the power emitted and the error rate. Among different channel coding techniques, we find the turbo codes family invented by Berrou *et al.* [6] in 1992.

On the encoder side, the principle is to code the data with two recursive systematic convolutional (RSC) codes separated by an interleaver.

On the decoder side (**Figure 10**(**a)**), two soft-in soft-out (SISO) elementary decoders work alternately. Each of them benefits from the other through extrinsic information. The iterative process gives performance close to the Shannon limit. Turbo codes are detailed more in [7]. **Figure 10(b)** shows the architecture of our turbo decoder implementation:

Table 2. Description of the different RAM required in each ICU.

RAM name	size
MY_{FA}	2 × 8 bits
MY_{MRC}	1 × 8 bits
$M\_Y_{m,k}$	K_{max} × 8 bits
$Me_{m,k}$	5 × SF × 2 × 8 bits

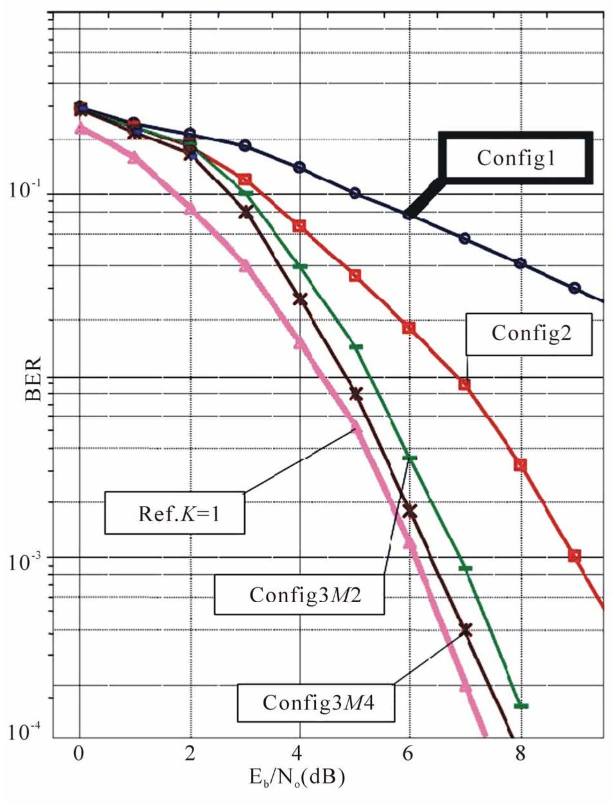


Figure 9. BER vs SNR for different configurations with comparison with single-user performance (spreading Factor = 16, K = 16 users, load rate = 100%).

- ♦ input memory to store the word to decode,
- ♦ a single decoder to perform the iterative process,
- ♦ internal memory to exchange the extrinsic information,
- ♦ output memory to store the decoded word.

3.2. Conventional Detection (CONFIG1)

At present, detectors implemented in base stations involve a bank of matched filters (rake detector) followed by a bank of channel decoders. Then a hard decision function determines the received sequence for each user as described in **Figure 11(a)**. To ensure reduced complexity, we choose to implement one branch and then accumulate L times. The architecture is shown in **Figure 11(b)** where it should be noted that the area is around 1900 gates and the processing time is around 110 clock cycles.

3.3. SIC Detector Followed by Turbo Decoder (CONFIG2)

We propose three architectures to implement *M*-stage of SIC detector for *K* users followed by a bank of turbo

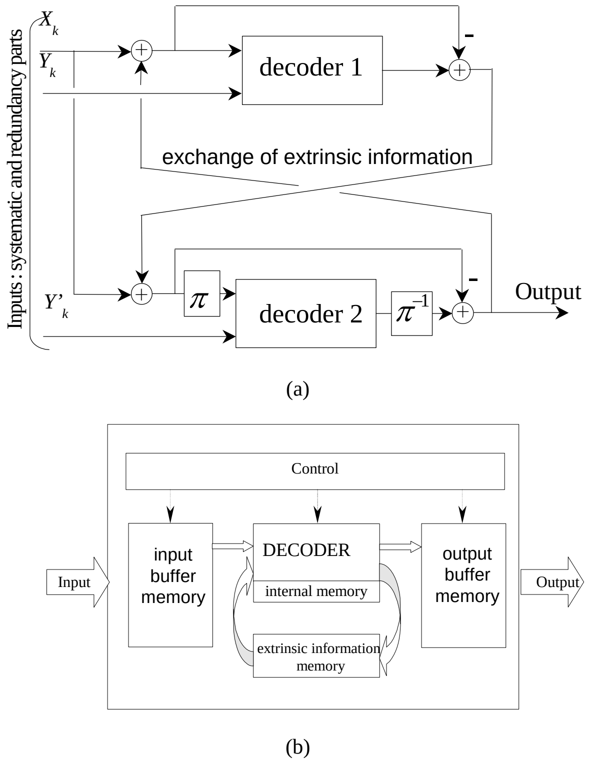


Figure 10. Turbo decoder (a) principle; (b) architecture for implementation.

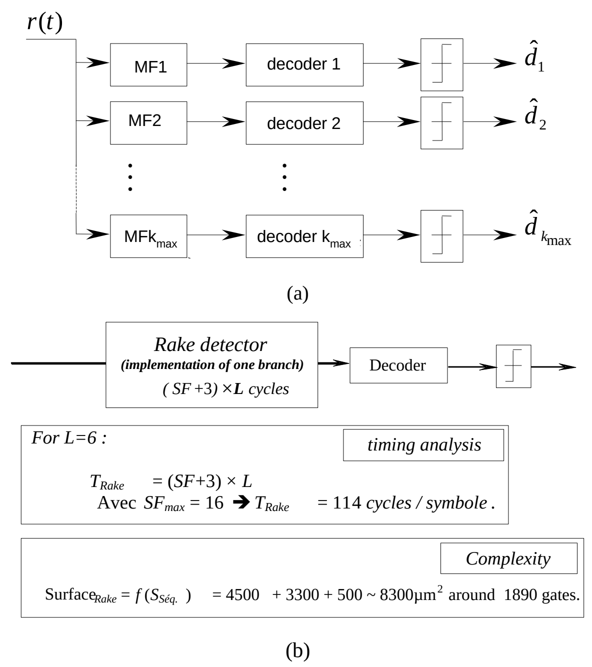


Figure 11. Rake conventional detection: (a) bank of matched filters followed by bank of decoders; (b) lowest complexity, implementation of one branch followed by an L-loop accumulator.

decoders. To process the complete detection function we can choose between:

Architecture A: Implementation of one ICU and processing $K \times M$ loops;

Architecture B: Implementation of one stage of K ICUs and processing M loops;

Architecture C: Implementation of M stages of K ICUs.

Table 3 gives the timing and complexity analysis of the three architectures proposed. We can notice that the latency is the same for all the architectures. The process time can be greatly increased, depending on M or K values, except for Architecture C. But the total area is inversely proportional to the required time to process data. In CONFIG 2, it is essential to implement a memory unit in each decoder as described in **Figure 10(b)** to allow correct transfer of the received sequence.

3.4. Turbo Decoder inside the Interference Cancellation Unit (CONFIG 3)

As shown in **Figure 9**, the turbo decoder is placed inside the ICU. This configuration ensures a better estimation before the interference cancellation function. In terms of bit error rate, this structure allows for better results. For the complete implementation, we can also choose between the three architectures A, B and C described in Subsection 3.3. This configuration does not require an external channel decoder, which results in a simpler global architecture.

To process the decoding function, knowledge of the whole frame is required. That is why ICU$_k$ processes the frame before sending information to ICU$_{k+1}$. This is an important difference from the previous configurations.

The internal structure of the ICU described in **Figure 3** has to be modified as shown in **Figure 12**. What about the impact on area?

- ♦ The area of a turbo decoder is around 450,000 m^2.
- ♦ The area of the local emitter and the interference cancellation do not change in comparison with CONFIG 2.
- ♦ For the matched filter it is essential to implement a combinational structure because we have to process a whole frame, so the area is increased by a factor of 10.
- ♦ We have to insert 2 adders.

Thus the computational area of an ICU in CONFIG 3 is around 165,000 mm^2 or 37,600 gates including the turbo decoder. The data given in **Table 3** are correct except for the area of turbo decoding, which are now included in S$_{ICU.}$

Table 3. Comparisons of area and timing for the three architectures considered.

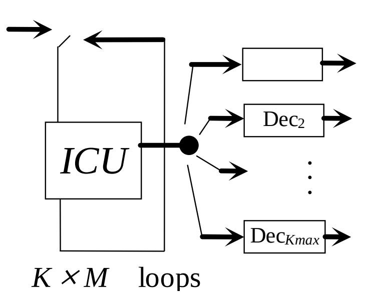
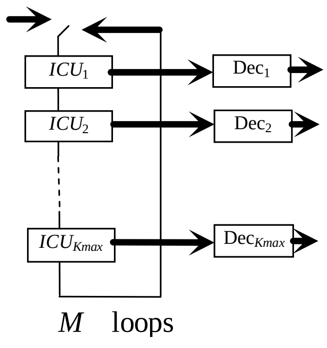
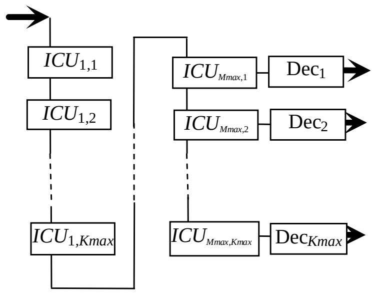
	A	B	C
Architecture		 ICU_1 ICU_2 ICU_{Kmax} Dec₁ Dec₂ Dec$_{Kmax}$ M loops	 $ICU_{1,1}$ $ICU_{1,2}$ $ICU_{1,Kmax}$ $ICU_{Mmax,1}$ $ICU_{Mmax,2}$ $ICU_{Mmax,Kmax}$ Dec₁ Dec₂ Dec$_{Kmax}$
Area for SIC detector	S_{ICU}	$S_{ICU} \times K_{\max}$	$S_{ICU} \times K_{\max} \times M_{\max}$
Area for turbo decoding		$K_{\max} \times S_{dec}$	
Process time	$f(K, M, T_{ICU})$	$f(M, T_{ICU})$	$f(T_{ICU})$
Latency of the detector		$f(K \times M \times T_{ICU})$	

Table 4. Comparisons of the different configuration.

	CONFIG 1	CONFIG 2	CONFIG 3
Architecture chosen	a rake, a decoder , $K_{\max}$ loops	a step of $K_{\max}$ ICU, then $K_{\max}$ decoders	a ICU with a decoder inside $K \times M$ loops
Surface	around 0.5 mm^2	around 10 mm^2	around 1.2 mm^2
Processing time (500 MHz)	2.5 ms	2.4 ms	8 ms
SNR required to reach BER = 10^{-3}, with load rate 100%	Not reached	8.8 dB	7 dB

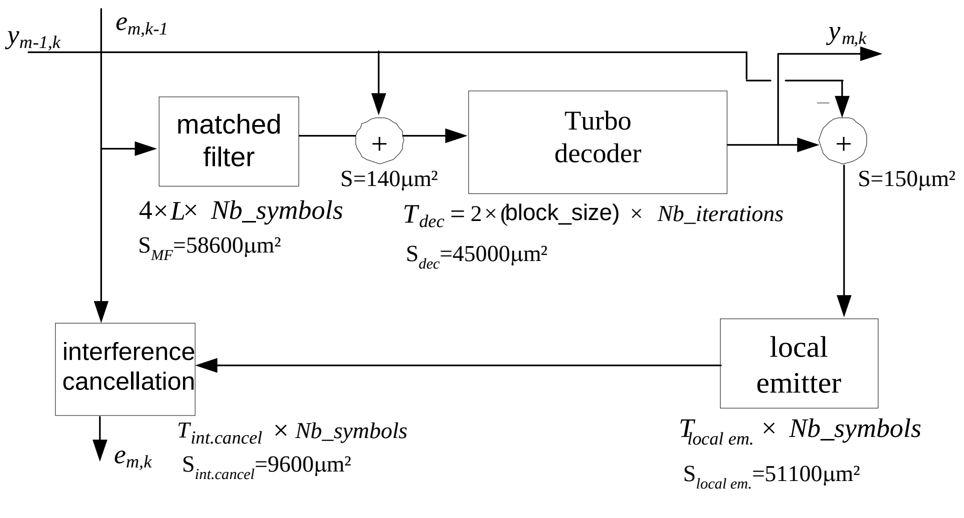

$y_{m-1,k}$
$e_{m,k-1}$
$y_{m,k}$
matched filter
+
Turbo decoder
–
+
S=140µm²
S=150µm²
$4 \times L \times Nb\_symbols$
S_{MF}=58600µm²
$T_{dec} = 2 \times$(block_size) $\times$ $Nb\_iterations$
S_{dec}=45000µm²
interference cancellation
local emitter
$e_{m,k}$
$T_{int.cancel} \times Nb\_symbols$
$S_{int.cancel}$=9600µm²
$T_{local\ em.} \times Nb\_symbols$
$S_{local\ em.}$=51100µm²

Figure 12. Timing analysis and complexity of the ICU implemented in CONFIG 3.

4. Comparison and Conclusions

In the previous section we describe three different configurations. In this section, we update the data for a real case study in order to see what can be the best choice. The parameters in the UMTS-FDD context are:

- ♦ received rate: 3.84 Mchip/s;
- ♦ frame length: 10 ms;
- ♦ delay from point to point: from 150 ms to 300 ms.

To evaluate the required time to compute one frame for a load rate of 100% (K_{max}= 16 users and SF = 16), we apply a frequency of 100 MHz or 500 MHz to the circuit.

In the case of CONFIG 1, which is the solution presently implemented in the base station, we compare the solution of computing the K users successively or simultaneously:

- ♦ The first solution is less complex and it is possible to process one frame in less than 10ms. Indeed, if the clock frequency is 500 MHz, taking into account the results given in **Figure 11(b)** for the rake and in **Figure 12** for the decoder, the delay required to compute one frame is around 5 ms.
- ♦ If we choose to implement a parallel process to compute the K users, the delay is less than 1ms but the complexity increases by almost K.

In the case of CONFIG 2, only architectures B and C described in Subsection 3.3 can compute the frame in less than 10 ms. To optimize the surface in CONFIG 2, we choose to implement one step of ICU and K decoders (Architecture B).

In **Table 4**, we sum up the performance, area and computing time for the different architectures retained. **Figure 9** gives the performance in terms of BER. We compare the different solutions by indicating the required SNR to reach a BER of 10^{-3} when K_{max} = 16 users and SF = 16 (load rate = 100%).

The solution implemented today cannot reach the performance required in UMTS context unlike CONFIG 2 and CONFIG 3. What is more, the complexity and timing analysis studies show that architecture A can be retained for CONFIG 3, whereas we have to choose architecture B for CONFIG 2. Thus, the final result is that it is possible to implement a turbo decoder inside the interference cancellation unit required for detection. Indeed, the area is only three times higher for a beneficial gain in term of BER.

5. References

[1] M. Ammar, S. Saoudi and T. Chonavel, “Iterative Successive Interference Cancellation for Multiuser DS-CDMA Detectors in Multipath Channels,” *Annales des Télécommunications*, Vol. 57, No. 12, 2002, pp. 105-124.

[2] S. Saoudi, M. Ammar and T. Chonavel, “Dispositif et Procédé de Décodage de Données AMRC, Système Correspondant,” *Institut National de la Propriété Industrielle*, Patent FR 03 14938, 2003.

[3] M. Touzri, “Étude d’implantation de Détecteurs Multi-utilisateurs CDMA: Application à l’UMTS,” PhD: Electronique: Institut TELECOM; TELECOM Bretagne, Université de Bretagne Occidentale: 2007, 2007telb0031. p. 130.

[4] P. Patel and J. Holtzman, “Analysis of DS-CDMA Successive Interference Cancellation Scheme Using Correlations,” *IEEE Globecom*’93, Houston, Vol. 1, 1993, pp. 76 -80.

[5] L. C. A. Hui and K. B. Letaief, “Successive Interference Cancellation for Multiuser Asynchronous DS-CDMA Detectors in Multipath Fading Links,” *IEEE Transactions on Communications*, Vol. 46, No. 3, 1998, pp. 384-391.

[6] C. Berrou, A.Glavieux and P. Thitimajshima, “Near Shannon Limit Error-Correction Coding and Decoding Turbo Codes,” *IEEE International Conference on Communications* (*ICC*’93), Vol. 2, 1993, pp. 1064-1070.

[7] C. Berrou, Ed., “Codes and Turbo Codes,” Springer, Germany, 2010.