Blind Identification of the Scrambling Code of a Reverse Link CDMA2000 Transmission

Mathieu Des Noes, Valentin Savin, Jean-Marc Brossier, Laurent Ros

Blind Identification of the Scrambling Code of a Reverse Link CDMA2000 Transmission

Mathieu des Noes and Valentin Savin
CEA, LETI, Minatec campus
Grenoble, France
Email: {mathieu.desnoes,valentin.savin}@cea.fr

Jean Marc Brossier and Laurent Ros
GIPSA-Lab, BP46, 38402 Saint-Martin d'Hères, France
Email: {laurent.ros,jean-marc.brossier}@gipsa-lab.grenoble-inp.fr

***Abstract*—Interference between macro and femtocells is an important issue for the development of CDMA2000 femtocell networks. More specifically, the reverse link signal of a macro User Equipment may generate an unacceptable level of interference at the femto Base Station. To avoid this situation, interference mitigation techniques could be implemented. All the proposed techniques require to know the state of the scrambling code of the interferer in the reverse link. Unfortunately, it depends on the code mask of the terminal which is unknown by the femto BS. The femto BS has to estimate blindly the state of the scrambling code. An algorithm which performs a blind identification of the scrambling code of a CDMA2000 reverse link transmission is proposed in this article. This gives the possibility to implement interference cancelation algorithm at the femto BS.**

INTRODUCTION

The development of femtocell networks is an important perspective for increasing cellular systems capacity. A femtocell is covered by a small Base Station (BS), designed for typically indoor environments (home, business) which does not require a coordinated deployment [1]. Although most of the current research activities on this topic focus on the LTE system [2], current deployed systems use the UMTS-WCDMA or CDMA2000 technologies [3]. In this paper, an algorithm which estimates blindly the state of the scrambling code of a terminal in the reverse link is proposed. It offers the perspective of implementing interference mitigation techniques at a femto Base Station (BS), and hence providing a solution to the problem of macro to femto interference issue in the reverse link of CDMA2000 femtocell networks.

A macro BS provides the overall coverage, while femto BSs offer better indoor coverage to UEs attached to them. A femto BS can be configured to operate in open or closed access mode to visiting UEs [4]. In open access mode, a visiting UE is allowed to handover from the macro BS to a femto BS in order to send its data. This generates additional complexity to route the data packets and also to ensure communication security. In closed access mode, a visiting UE is not allowed to handover. This simplifies the network architecture, but this may lead to an unacceptable interference level at the femto BS. This situation occurs if a visiting UE transmits at high power, while it is located nearby the femto BS.

This situation is depicted in Fig. 1. A macro UE transmits at high power because it is either at the cell edge or inside a building. Since the power control procedure concerns only the link with the macro BS, it will be received at the femto BS with a power much larger than the power of a femto UE. This is the well known near-far effect in CDMA systems [5]. If the power of this interferer is too large, the femto BS may not be able to demodulate any communication with its attached UEs. This creates a "dead zone" in the network coverage. In order to avoid this situation, it is required to implement interference mitigation techniques. This subject has been deeply studied in the past two decades for CDMA systems [6]. All the proposed techniques exploit the knowledge of the UE's scrambling code. More precisely, it is required to know the state of the UE's scrambling code generator at the beginning of each frame.

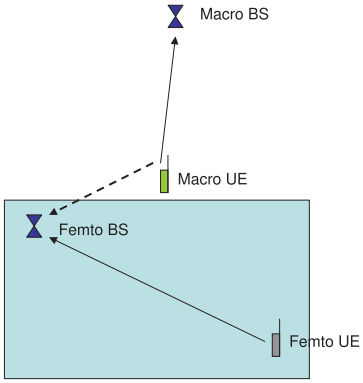

Fig. 1. Interference generated by a macro UE on a femto BS

The state of the scrambling code depends on its "code mask" which is used to generate a user specific delay of a long scrambling code common to all UEs. Unfortunately, in a closed access mode, there is no signalling link between the macro and femto BSs. Hence, a femto BS has no knowledge of the code mask of an interfering macro UE. It has to estimate the state of the scrambling code of a macro UE blindly. To the authors' knowledge, [7] is the first and only article addressing this issue. The authors exploit the specificities of the framing, spreading and multiplexing procedures defined by CDMA systems to process the signal so that the resulting signal can be considered to be a linear code depending on the initial state of the scrambling code generator. This processing step can be applied to both WCDMA and CDMA2000 systems [8][9]. The received codeword is then estimated with a max-

log-MAP algorithm [10]. The initial state of the different spreading code generators is eventually obtained by applying a pseudoinverse of the code generator matrix. Concerning the CDMA2000 system, the solution proposed in [7] is adapted to a BPSK modulation which concerns Radio Configurations (RC) 3 to 6. However, it does not work for the 64-ary orthogonal modulation used by RC 1 and 2. These RCs are important because they are used by voice applications.

This paper addresses the 64-ary orthogonal modulation case and also proposes a different decoding strategy. Exploiting the unique properties of m-sequence, it is shown that an iterative message-passing algorithm [11] can be implemented to decode the received signal after an initial processing step. Then the initial state of the scrambling code can be obtained with a proper used of the "Shift and add" and decimation properties of m-sequences.

The paper is organized as follows. Section I presents the procedure used to generate the reverse link of CDMA2000 signals for RC 1 and 2. The construction of the complex scrambling code is also detailed since its specific properties will be exploited by the proposed algorithm. Section II details the blind identification algorithm, which is split in 3 steps. Section III presents simulation results and Section IV concludes this paper.

Notation: a sequence will be written with upper case in its BPSK representation ($S(k) \in \{-1, +1\}$) and with lower case in its binary representation ($s(k) \in \{0, 1\}$).

I. Reverse link signal generation

In this section, the modulation and spreading operations implemented in the reverse link of the CDMA2000 system are first described. Then, the method standardized for constructing the scrambling code is detailed [9].

A. Modulation and spreading

The reverse link implements a Code Division Multiple Access (CDMA) scheme. Fig. 2 details the modulation and spreading operations for RC 1 and 2. The data are mapped on a 64-ary orthognal modulation, repeated, scrambled by the long code S_n, multiplied by the complexe spreading sequence $C_I + jC_Q$ and eventually filtered by the pulse shaping filter $P(t)$.

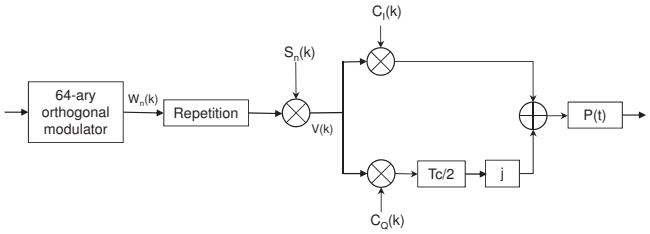

Fig. 2. Modulation and spreading operations

The 64-ary orthogonal modulation consists in mapping 6 consecutive binary data $d_i, d_{i+1}, \cdots, d_{i+5}$ onto a Walsh-Hadamard (WH) sequence of length 64 chips [9]. The WH sequence index is selected as follows:

$$m = d_i + 2d_{i+1} + 4d_{i+2} + 8d_{i+3} + 16d_{i+4} + 32d_{i+5}$$

Then each chip of the WH sequence W_m is repeated 4 times and scrambled by the long code S_n. The subscript n indicates that the long scrambling code is specific to user n. The scrambled signal corresponding to sequence W_m is thus:

$$V(k) = S_n(k) W_m\left(\lfloor k/4 \rfloor\right) \tag{1}$$

$\lfloor x \rfloor$ is the largest integer smaller than x.

The chips of sequences S_n and W_m are anti-modal representation (+1,-1) of binary sequences s_n and w_m.

Sequence $V(k)$ is then multiplied by a complex spreading sequence $C_I + jC_Q$. These 2 sequences are built from m-sequences of period $N = 2^{15} - 1 = 32767$ chips [12]. Their characteristic polynomials are :

$$\begin{aligned} g_I(D) &= D^{15} + D^{13} + D^9 + D^8 + D^7 + D^5 + 1 \\ g_Q(D) &= D^{15} + D^{12} + D^{11} + D^{10} + D^6 + D^5 + D^4 + \\ &\quad D^3 + 1 \end{aligned}$$

The sequences are complemented with a '0' in order to create a frame of $N = 32768$ chips. According to the standard specifications, the initial state of each sequence is known by the receiver at the beginning of each frame.

In this article, the gating procedure associated with power control is not considered. This corresponds to a gating rate equals to 1 in the standard [9]. A smaller gating rate ($1/2$ or $1/4$) would require the implementation of parallel instances of the algorithm proposed in this article.

The real part of the complex signal is filtered by the pulse shaping filter $P(t)$, while the imaginary part is delayed by half a chip before pulse shaping. This implements an offset QPSK modulation. The specifications of $P(t)$ are detailed in [9]. The transmitted signal can eventually be modeled as follows :

$$X(t) = \sum_k V(k) G_k(t - kT_c) \tag{2}$$

with $G_k(t) = C_I(k)P(t) + jC_Q(k)P(t - T_c/2)$.

T_c is the chip period and is equal to 814 ns. $G_k(t)$ could be considered as a time-varying pulse shaping filter applied to sequence $V(k)$.

B. Scrambling code generation

The scrambling code is built from a very long m-sequence a of period 2^{42} chips. Fig. 3 shows the Linear Feedback Shift Register (LFSR) sequence generator according to the Galois representation [13]. Its characteristic polynomial is :

$$\begin{aligned} g_s(D) &= D^{42} + D^{35} + D^{33} + D^{31} + D^{27} + D^{26} + \\ &\quad D^{25} + D^{22} + D^{21} + D^{19} + D^{17} + D^{16} + \\ &\quad D^{10} + D^7 + D^6 + D^5 + D^3 + D^2 + D + 1 \end{aligned}$$

The content of the i^{th} shift register at time k is noted $a_i(k)$. The scrambling code is generated by the modulo-2 inner

product of a 42-bit mask and the 42-bit state vector of the sequence generator:

$$s_n(k) = \bigoplus_{i=0}^{r-1} a_i(k) m_n(i)$$

where $\oplus$ denotes the XOR operation. $m_n(i)$ is the code mask of the n^{th} user. The vector $a_0(k), a_1(k), \cdots, a_{r-1}(k)$ is known by the receiver at some specified time since it is signalled periodically by the network. On the other hand, the code mask is unknown to the receiver and thus it cannot generate sequence s_n with the schematic of Fig. 3.
Due to the "Shift and add" property of m-sequences, s_n is also a m-sequence with characteristic polynomial $g_S(D)$ [12]. It is a shifted version of sequence a: $s_n(k) = a(k+\tau_n)$. τ_n is specific to the n^{th} user. It depends on the code mask and the characteristic polynomial. Using the Fibonacci representation, s_n can be generated with the generator depicted in Fig. 4[12]. Instead of estimating the code mask, knowing the initial state of sequence a, it is simpler to estimate the content of the shift registers of sequence s_n with the Fibonacci representation. This is the objective of the algorithm proposed in this paper.

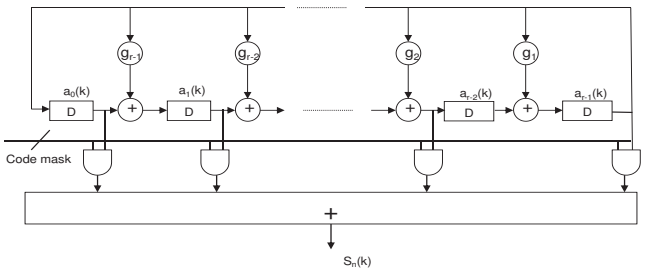

Fig. 3. Long scrambling code generation

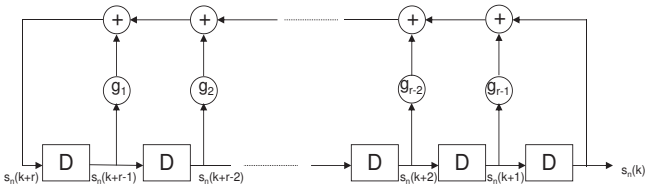

Fig. 4. LFSR sequence with the Fibonacci representation

II. Blind identification of the scrambling code

The algorithm is split in 3 steps:

- step 1 : signal processing which aims at allowing the direct observation of a modified version of the sequence s_n, denoted by $\tilde{s}$.
- step 2 : estimation of the initial state of sequence $\tilde{s}$ with an iterative message-passing decoder.
- step 3 : determination of the initial state of the sequence s_n with the use of a transposition matrix.

In order to ease the comprehension of the algorithm, the description is restricted to an Additive White Gaussian Noise (AWGN) channel model. The robustness to multipath channel will be discussed in section II-E.

A. Signal processing

The received signal is down-converted to baseband and sampled at rate rate $T = T_c/E$, where E is the oversampling factor.
It is modeled as follows (see Eq. (2)):

$$R(q) = e^{j\theta} \sum_k V(k) G_k(qT - kT_c) + N(q) \qquad (3)$$

where θ is the phase rotation introduced by the channel and $N(q)$ the additional noise modeling thermal noise as well as other sources of interference. The objective is to obtained an observation of a modified version of the sequence s_n, and thus get rid off $G_k(t)$, θ and sequence W_m. The pulse shaping filter defined by the standard generates inter-chip interference. As a consequence, a simple matched filter is not sufficient and an equalizer shall be implemented. Hence, the first step consists in implementing an equalizer to the time varying pulse shaping function $G_k(t)$. Let us define this equalizer as a FIR filter with L coefficients: $F_k(0), \cdots, F_k(L-1)$. It can be implemented either with a zero-forcing (ZF) or a minimum mean square error (MMSE) criterion [14].
If the receiver is synchronized with the transmit signal, after equalization and downsampling, the signal is well approximated by :

$$Y(k) = \sum_{l=0}^{L-1} R(kE + L/2 - l) F_k(l) \approx e^{j\theta} V(k) + N_{eq}(k) \quad (4)$$

$N_{eq}(k)$ is the filtered noise.
If the receiver is not synchronized, the equalizer output can be regarded as a random noise sequence. This is due to the correlation properties of m-sequences c_I and c_Q [12].
The second step consists in removing the dependency upon sequence W_m and phase θ. Since, the chips of sequence W_m are repeated 4 times, a differential multiplication is performed within each window of 4 repeated chips. For instance, the differential multiplication using only the first two chips of the 4 chips length window will give :

$$\begin{aligned} Y(4k+1)Y(4k)^* &= V(4k+1)V(4k) + N_{diff}(k) \\ &= S_n(4k+1)S_n(4k) + N_{diff}(k) \end{aligned}$$

where $N_{diff}(k)$ contains all the noise cross-product terms.
In fact there are 7 possible differential products:

$$\begin{aligned} U_{i,j}(k) &= Y(4k+i+j)Y(4k+i)^* \\ &= S_n(4k+i+j)S_n(4k+i) + N_{i,j}(k) \end{aligned} \qquad (5)$$

$i = 0, 1, 2$ and $j = i+1, \cdots, 3$. Let us define the sequence $\tilde{S}_{i,j}(k) = S_n(4k+i+j)S_n(4k+i)$, which is rewritten in GF(2) by : $\tilde{s}_{i,j}(k) = s_n(4k+i+j) \oplus s_n(4k+i)$.
$U_{i,j}$ is thus a noisy observation of sequence $\tilde{S}_{i,j}$. The objective is now to find the relationship between sequences $\tilde{s}_{i,j}$ and s_n. To do so, two interesting properties of m-sequences will be exploited [12]:

- "Shift and add" property : for two given delays τ_1 and τ_2, there exists a unique τ_3 such that:

$$x(k+\tau_1) \oplus x(k+\tau_2) = x(k+\tau_3)$$

- Decimation property: the decimation by a factor 2 of a m-sequence gives a shifted version of this m-sequence. There exists a unique τ such that:

$$x(2k) = x(k+\tau)$$

These 2 properties allow us to conclude that $\tilde{s}_{i,j}$ is a shifted version of sequence s_n. As a consequence, $\tilde{s}_{i,j}$ has the same characteristic polynomial as s_n: $g_s(D)$.
It is known from the literature that the initial state of a LFSR sequence can be estimated with a standard iterative message-passing algorithm exploiting the sequence's characteristic polynomial [11][15][16]. It is thus possible to estimate the initial state of sequence $\tilde{s}_{i,j}$ using its characteristic polynomial $g_s(D)$.
The algorithm's principle is thus to decode the initial state of sequence $\tilde{s}_{i,j}$ and to recover the state of sequence s_n using a fixed state transposition matrix with sequence $\tilde{s}_{i,j}$.
The successive steps of the algorithm are the followings. At each time instant q, the receiver assumes that it is synchronized with the beginning of the frame and equalizes the input signal according to Eq. (4). Then a differential multiplication is done, as in Eq. (5), to compute sequence $U_{i,j}$. Since $\tilde{s}_{i,j}$ is a shifted version of sequence s_n, it can be decoded with an iterative message-passing algorithm which parity check matrix is matched to the characteristic polynomial of sequence s_n (see section II-B). The vector $(U_{i,j}(0), \cdots, U_{i,j}(M-1))$ feeds this decoder. If the decoder fails to find a codeword, this means that either the frame synchronization assumption is not valid or a missed detection happened. In both cases, the procedure is restarted when the next sample is received. If the decoder finds a valid codeword, the initial state of sequence s_n is found by a simple matrix multiplication (see section II-C).

B. Iterative message-passing decoding

A m-sequence x, with characteristic polynomial $g(D)$, satisfies the following parity check equation ($g_0 = g_r = 1$), for all $k \geq 0$:

$$\bigoplus_{i=0}^{r} g_{r-i} x(k+i) = 0$$

A m-sequence is thus a cyclic linear code with rate $\frac{r}{2^r-1}$. A codeword is generated by one initial state of the shift registers. In the context of this paper, only a sequence of M variables, corresponding to M consecutive bits of the codeword, is observed. The parity check matrix of this code depends on the sequence's characteristic polynomial $g(D)$:

$$H = \begin{bmatrix} g_r & \cdots & g_0 & 0 & \cdots & \cdots & 0 \\ 0 & g_r & \cdots & g_0 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & g_r & \cdots & g_0 & 0 \\ 0 & \cdots & \cdots & 0 & g_r & \cdots & g_0 \end{bmatrix} \quad (6)$$

Once the parity check matrix has been defined, it is possible to decode the received message with a standard iterative message passing algorithm [17] [18]. In addition, as it was proposed in [11][15], the use of Redundant Graphical Model (RGM) improves significantly the decoder performance. It relies on the following property : if $g(D)$ is the sequence's characteristic polynomial, it satisfies $g(D^{2^n}) = g(D)^{2^n}$.
This property is exploited to create additional parity check equations. Polynomial $g_n(D) = g(D^{2^n})$ also generates a parity check matrix H_n similar to H. Theses matrices can be concatenated to create a larger parity check matrix H_{RGM} [11]:

$$H_{\text{RGM}} = \begin{bmatrix} H_0 \\ H_1 \\ \vdots \\ H_{n_{\text{RGM}}-1} \end{bmatrix} \quad (7)$$

where n_{RGM} is the number of RGMs used for decoding. These RGMs increase the column weight of the parity check matrix, while keeping the row weight constant.
If the decoding is successful (all parity check equations are satisfied), the soft decision output of the decoder is convert into a binary representation with a hard decision rule. Then, according to the Fibonacci representation (Fig. 4), the first r bits of the codeword are the content of the shift registers at initialization.

C. Determination of the initial state of the sequence s_n

The decoder provides a vector of r bits representing the initial state of sequence $\tilde{s}_{i,j}$, denoted by $A_{\tilde{s}}$. We want to find the initial state of sequence s_n at the beginning of the frame: $A_{s_n} = (s_n(0), \cdots, s_n(41))^T$. This task is achieved in 2 steps. The first step consists in finding the initial state of the sequence s_{decim}, $A_{s_{\text{decim}}}$, which gives $\tilde{s}_{i,j}$ by decimation by a factor 4:

$$\tilde{s}_{i,j}(k) = s_{\text{decim}}(4k)$$

[19] shows that there is a fixed transposition matrix B between the state vectors of a m-sequence and its decimated by 2 version, and detailed the procedure to compute this matrix B. As a consequence:

$$A_{s_{\text{decim}}} = B^2 A_{\tilde{s}}$$

The second step eventually provides vector A_{s_n}. It exploits the "shift and add" relation between the sequences s_{decim} and s_n:

$$s_{\text{decim}}(k) = s_n(k+i+j) \oplus s_n(k+i)$$

Using the state transition matrix G, defined by [13]:

$$G = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & 0 & \cdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 \\ 1 & g_{r-1} & g_{r-2} & g_{r-3} & \cdots & g_1 \end{bmatrix} \quad (8)$$

we have:

$$A_{s_n} = (G^{i+j} + G^i)^{-1} A_{s_{\text{decim}}}$$

These two steps are finally combined in a single transposition matrix T:

$$A_{s_n} = T A_{\tilde{s}}$$
$$T = (G^{i+j} + G^i)^{-1} B^2$$

It is important to note that matrix T could be computed off-line and stored in memory.

D. Flow chart of the algorithm

A flow chart of the algorithm is presented in Fig. 5. The decoding operation must be implemented at the chip rate, which requires a very high data rate decoding capability.

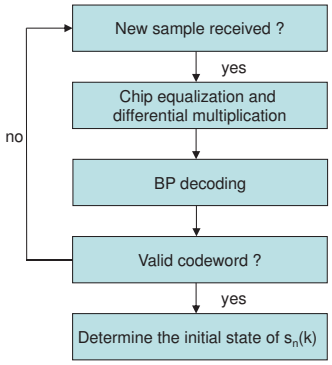

Fig. 5. Flow chart of the algorithm

E. Robustness to a multipath channel

The algorithm is by construction robust to multipath whose delays are larger than a chip period. This is due to the chip level equalization of Eq. 4. A multipath component delayed by more than one chip will be scrambled by sequences c_I and c_Q and will be considered as an additional noise source by the decoder. In order to cope with multipath, the receiver implements a search window of length W chips. At each time q, the receiver runs the detection algorithm for time q to $q + W - 1$. If a valid codeword is detected within this window, the detection is declared successful.

III. Simulation results

Let us first define the conventional synchronization hypothesis :

- H_0 : the receiver is not synchronized with the beginning of the frame.
- H_1 : the receiver is synchronized with the beginning of the frame.

The performance of the algorithm are measured by the probabilities of correct detection P_d, false alarm P_{fa} and missed detection P_m, defined as follows:

$$P_d = P(Ic = 1 \text{ and } \hat{A}_{x_n} = A_{x_n} | H_1)$$
$$P_{fa} = P(Ic = 1 | H_0)$$
$$P_m = 1 - P_d$$

Ic is the indication function of the decoder:

$$Ic = \begin{cases} 1 \text{ if all parity check equations are satisfied} \\ 0 \text{ otherwise} \end{cases}$$

$\hat{A}_{x_n}$ is the estimated initial state of sequence x_n, given by the decoder output.

Performances are measured with the following simulation configurations:

- When measuring P_m, the receiver is synchronized with the beginning of the frame (i.e. hypothesis H_1 is satisfied), while its is not synchronized when P_{FA} is evaluated.
- The decoder implements a Self-Corrected Min-Sum (SCMS) message-passing algorithm [20], which provides a very high data rates decoding capability, while performing close to the "optimal" Belief Propagation (Sum-Product) decoding. The decoder stops when either all the parity check equations are satisfied or the maximum number of iteration N_{iter} is reached.
- The number of variable is $M = 1500$ at the decoder input. This value is selected so that it is smaller than the size of a Power Control Group (PCG) which contains 1536 chips [9]. A PCG is equivalent to a slot in the WCDMA system.
- The channel is AWGN.

Simulations over 10^7 frames did not produce any false alarm. This means that $P_{fa} < 10^{-6}$ with a good confidence level.

Fig. 6 shows the probability of missed detection for a chip level MMSE, ZF or MRC equalizer. The number of RGMs is set to its largest possible value $n_{RGM} = 6$ for $M = 1500$ variables and $N_{iter} = 100$. P_m is measured as a function of the Signal to Noise Ratio (SNR) at the input of the receiver. At $P_m = 0.1$, the MMSE equalizer gives a gain of almost 1 dB and 3 dB with respect to ZF and MRC. Fig. 7 shows the probability of missed detection as a function of the number of RGMs for a receiver configuration with chip level MMSE equalizer. It is not worth increasing the number of RGMs above 4.

According to [21], the target Eb/No at a base station is around 7 dB for the 9.6 kbps service which corresponds to our simulation configuration (all the PCGs are transmitted). Since the combined coding-spreading gain equals 128, the target SNR at a base station is around -14 dB. According to Fig. 7, the algorithm implementing a chip level MMSE equalizer detects the scrambling code of an interfering user at a SNR of 2.5 dB. This corresponds to a very strong interferer since its received power is 16.5 dB larger than a desired user attached to the femtocell. However,the performance of the algorithm is not good enough to detect certain lower but still strong interferes. For instance, an interferer which power is 15 dB larger than the desired user will not be detected even it will have very damaging effects. The bad detection performance of the algorithm is due to the weight of the characteristic polynomial of the scrambling code generator. This effect has been well studied in the framework of fast

correlation attacks [16]. The weight equals 20 while it should be below 4 or 5 to give good decoding performance [22]. One way to improve performance is to generate a parity check matrix which row weight is below 4 or 5. This could be achieved by implementing the technique proposed in [23].

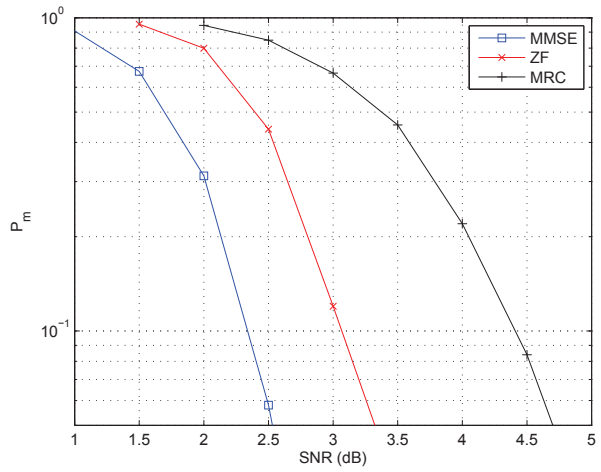

Fig. 6. Influence of the equalizer - AWGN channel - $N_{iter} = 100$, $M = 1500$ and $n_{\text{RGM}} = 6$

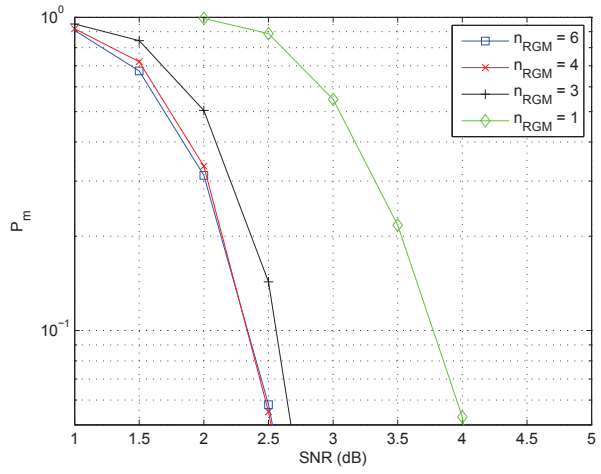

Fig. 7. Sensitivity to the number of RGM - AWGN channel - MMSE equalizer , $N_{iter} = 100$ and $M = 1500$

IV. Conclusion

An algorithm which performs a blind identification of the scrambling code of a reverse link CDMA2000 transmission has been presented. It is dedicated to the 64-ary orthogonal modulation used by the radio configurations 1 and 2 of the standard. The simulation results show that with a BP decoding algorithm it is possible to obtain a reliable estimation at a SNR of 2.5 dB in an AWGN channel. It is also noticeable that the false alarm probability is below 10^{-6} for all the tested SNR values.

References

[1] V. Chandrasekhar, J. Andrews, and A. Gatherer, "Femtocell networks: a survey," *IEEE Communications Magazine*, vol. 46, no. 9, pp. 59–67, 2008.

[2] H. Holma and A. Toskala, *LTE for UMTS: OFDMA and SC-FDMA based radio access*, Wiley, 2009.

[3] H. Holma, A. Toskala, et al., *WCDMA for UMTS*, vol. 4, Wiley, 2000.

[4] P. Xia, V. Chandrasekhar, and J.G. Andrews, "Open vs. closed access femtocells in the uplink," *IEEE Transactions on Wireless Communications*, vol. 9, no. 12, pp. 3798–3809, 2010.

[5] S. Verdu, *Multiuser detection*, Cambridge Univ Pr, 1998.

[6] J.G. Andrews, "Interference cancellation for cellular systems: a contemporary overview," *IEEE Wireless Communications*, vol. 12, no. 2, pp. 19–29, 2005.

[7] R. Kerr and J. Lodge, "Iterative signal processing for blind code phase acquisition of CDMA 1x signals for radio spectrum monitoring," *Journal of Electrical and Computer Engineering*, vol. 2010, pp. 3, 2010.

[8] *TS25.213 v.4.4.0 Spreading and modulation (FDD)*, 3GPP, 2004.

[9] *3GPP2 C.S0002-A - Physical Layer Standard for cdma2000 Spread Spectrum Systems - Release A*, 3GPP2, 2000.

[10] P. Robertson, E. Villebrun, and P. Hoeher, "A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain," in *Proceedings of the IEEE International Conference on Communications (ICC'95)*, 1995, vol. 2, pp. 1009–1013.

[11] O.W. Yeung and K.M. Chugg, "An iterative algorithm and low complexity hardware architecture for fast acquisition of long PN codes in UWB systems," *The Journal of VLSI Signal Processing*, vol. 43, no. 1, pp. 25–42, 2006.

[12] R.J. McEliece, *Finite fields for computer scientists and engineers*, Springer, 1987.

[13] R.L. Peterson, R.E. Ziemer, and D.E. Borth, *Introduction to spread-spectrum communications*, Prentice Hall, 1995.

[14] J.G. Proakis, *Digital communications*, vol. 1221, McGraw-hill, 1987.

[15] F. Principe, K.M. Chugg, and M. Luise, "Rapid acquisition of gold codes and related sequences using iterative message passing on redundant graphical models," in *Proceedings of the IEEE Military Communications Conference (MILCOM'06)*, 2006, pp. 1–7.

[16] M. Mihaljevic, M. Fossorier, and H. Imai, "A low-complexity and high-performance algorithm for the fast correlation attack," in *Fast Software Encryption*. Springer, 2001, pp. 45–60.

[17] F.R. Kschischang, B.J. Frey, and H.A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. on Information Theory*, vol. 47, no. 2, pp. 498–519, 2001.

[18] N. Wiberg, *Codes and decoding on general graphs*, Ph.D. dissertation, Linköping University, Sweden, 1996.

[19] B. Arazi, "Decimation of m-sequences leading to any desired phase shift," *Electronics Letters*, vol. 13, no. 7, pp. 213–215, 1977.

[20] V. Savin, "Self-corrected min-sum decoding of ldpc codes," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT'08)*, 2008, pp. 146–150.

[21] R. Padovani, "The application of spread spectrum to PCS has become a reality: Reverse Link Performance of 1S-95 Based cellular systems," *IEEE Personal Communications*, vol. 1, no. 3, pp. 28, 1994.

[22] A. Canteaut and M. Trabbia, "Improved fast correlation attacks using parity-check equations of weight 4 and 5," in *Advances in Cryptology - EUROCRYPT 2000*. Springer, 2000, pp. 573–588.

[23] P. Chose, A. Joux, and M. Mitton, "Fast correlation attacks: An algorithmic point of view," in *Advances in Cryptology - EUROCRYPT 2002*. Springer, 2002, pp. 209–221.