Blind Identification of the Uplink Scrambling Code Index of a WCDMA Transmission and Application to Femtocell Networks

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

Blind Identification of the Uplink Scrambling Code Index of a WCDMA Transmission and Application to Femtocell Networks

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 WCDMA femtocell networks. More specifically, the uplink 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 the knowledge of the uplink scrambling code index of the interferer. Unfortunately, if the femto BS is in a closed access mode, there are no signalling links with the surrounding macro BSs. The femto BS has to estimate blindly this scrambling code index. An algorithm which performs a blind identification of the uplink scrambling code index of a WCDMA 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 technology [3]. In this paper, an algorithm which estimates blindly the uplink scrambling code index of a User Equipment (UE) 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 uplink macro to femto interference issue in WCDMA 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.

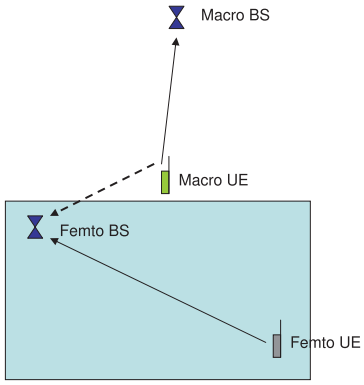

Fig. 1. Interference generated by a macro UE on a femto BS

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 uplink 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, which is chosen from a set of scrambling codes identified by their index.

The scrambling code index of the macro UE is allocated by the macro BS and is signalled to the UE in a dedicated control channel [7]. 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 scrambling code index of an interfering macro UE. It has to estimate this index blindly. To the authors' knowledge, [8] 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 CDMA 1x systems [7][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.

In this paper, a different decoding strategy is proposed. 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 the same initial processing step as in [8]. Then the scrambling code index can be obtained with a proper used of the "Shift and add" and decimation properties of m-sequences. In addition, in [8], the authors focused on the CDMA 1x system, while the WCDMA system is addressed in this article.

The paper is organized as follows. Section I presents the procedure used to generate the uplink WCDMA signal. 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. Uplink WCDMA signal generation

In this section, the modulation, spreading and multiplexing operations implemented in the uplink of the WCDMA system are first described. Then, the method standardized by 3GPP for constructing the scrambling code is detailed [7]. It is included here for completeness, since several specific features of the scrambling will be exploited by the proposed blind estimation algorithm.

A. Spreading and multiplexing

The uplink of the WCDMA system implements a Code Division Multiple Access (CDMA) scheme. The data and control channels are mapped on a BPSK constellation and I/Q multiplexed. In the standard, these channels are referred to as the Dedicated Physical Data CHannel (DPDCH) and Dedicated Physical Control CHannel (DPCCH). The bits are first spread by a channelization sequence (C_d and C_c) which ensures orthogonality between channels. The spreading factor of the DPCCH signal is equal to $SF_{\text{DPCCH}} = 256$, while for DPDCH, SF_{DPDCH} is variable and depends on the service data rate. When a DPDCH is transmitted, its channelization sequence C_d is a periodic repetition of the pattern "$+1, +1, -1, -1$". This is due to the selection of the column number $SF_{\text{DPDCH}}/4$ of the Walsh-Hadamard matrix [7]. The channelization sequence C_c used for DPCCH is the all '+1' sequence.

Each spread signal is then weighted by a factor β to set its power, and I/Q multiplexed. The resulting complex signal is eventually multiplied by the scrambling code S_n, where index n indicates that the scrambling code is user specific. At chip time index k, the transmitted signal can be modeled as follows:

$$E(k) = S_n(k)(\beta_d X_{\text{DPDCH}}(k) + j\beta_c X_{\text{DPCCH}}(k)) \qquad (1)$$

X_{DPDCH} and X_{DPCCH} models the DPDCH and DPCCH bits spread by sequences C_d and C_c.

In order to simplify the description, we have considered the case where only one DPDCH is transmitted. However, we note that the proposed algorithm also works when 2 DPDCHs are transmitted and I/Q multiplexed, as detailed [7].

B. Scrambling code generation

The scrambling code s_n is built from two m-sequences x_n and y. Fig. 2 shows a Linear Feedback Shift Register (LFSR) sequence generator according to the Fibonacci representation [12].

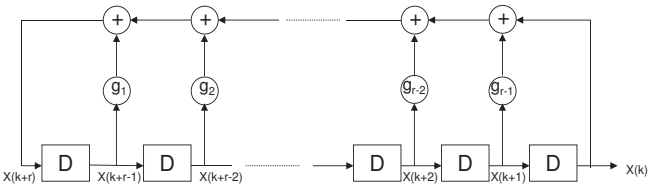

Fig. 2. LFSR sequence with the Fibonacci representation

The two m-sequences x_n and y are defined by their primitive polynomial :

$$g_x(D) = D^{25} + D^{22} + 1$$
$$g_y(D) = D^{25} + D^{24} + D^{23} + D^{22} + 1$$

These polynomials have a different definition than in the standard [7]. This is due to the notation used in Figure 2, which is the conventional notation for LFSR sequences defined in [12] and [13]. The polynomials defined in the standard are obtained by exchanging coefficient g_i by g_{r-i}.

The scrambling sequence is identified by an index n. Let $n_{23}, \cdots, n_0$ be the 24 bits binary representation of the index n with n_0 being the least significant bit. The registers of the sequences x_n and y are reset at the beginning of each frame (every 38400 chips), with the following values :

$$x_n(0) = n_0, x_n(1) = n_1, \ldots, x_n(23) = n_{23}, x_n(24) = 1$$
$$y(0) = y(1) = y(2) = \cdots = y(23) = y(24) = 1$$

The initial state of a LFSR sequence is defined by the content of its shift registers at time $k = 0$. As a consequence, the initial state of the sequence y at the beginning of each frame is known.

The complex scrambling code is built as follows ($k = 0, 1, \ldots, 38399$):

$$\begin{aligned} S_n(k) &= C_1(k)\left(1 + j(-1)^n C_2\left(2\lfloor \tfrac{k}{2} \rfloor\right)\right) \\ C_1(k) &= Z_n(k) \\ C_2(k) &= Z_n((k + 16777232) \bmod (2^{25} - 1)) \\ Z_n(k) &= 1 - 2(x_n(k) \oplus y(k)) \end{aligned} \qquad (2)$$

where $\lfloor x \rfloor$ is the closest integer inferior or equal to x and $\oplus$ denotes the XOR operation.
M-sequences have two interesting properties that will be exploited by the proposed algorithm [13]:

- "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:

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

According to the "shift and add" property of m-sequences, it can be proved that for $\tau = 16777232$:

$$\begin{array}{l} x_n(i+\tau) = x_n(i+4) \oplus x_n(i+7) \oplus x_n(i+18) \\ y(i+\tau) \ = y(i+4) \oplus y(i+6) \oplus y(i+17) \end{array} \quad (3)$$

This property is used to generate the sequence C_2.
The vector $(n_0, \cdots, n_{23})$ is allocated by the macro BS to the macro UE in a signalling message and is unknown to the femto BS. The objective of the algorithm proposed in this paper is to perform a blind estimation of this vector.

II. Blind identification of the scrambling code

The algorithm is split in 3 steps:

- step 1 : chip level processing which results in a direct observation of a modified version of the sequence x_n, denoted by $\tilde{x}$.
- step 2 : estimation of the initial shift registers state of sequence $\tilde{x}$ with an iterative message-passing decoder.
- step 3 : determination of the initial state of the sequence x_n with the use of a transposition matrix.

In order to ease the comprehension of the algorithm, we restrict the description to an Additive White Gaussian Noise (AWGN) channel model. The robustness to multipath channel will be discussed in section II-E and be tested by means of simulation. In addition, we also focus the description to a signal sampled at the chip rate.

A. Chip level processing

The received signal is modeled as follows (see Eq. (1)):

$$T(k) = e^{j\theta} S_n(k)(\beta_d X_{\text{DPDCH}}(k) + j\beta_c X_{\text{DPCCH}}(k)) + n(k) \quad (4)$$

where θ is the phase rotation introduced by the channel and $n(k)$ the additional noise modeling thermal noise as well as other sources of interference. First, a differential multiplication is performed on the received signal:

$$\begin{array}{rcl} U(k) & = & T(k+1)T(k)^* \\ & = & S_n(k+1)S_n(k)^* \{\beta_d^2 X_{\text{DPDCH}}(k+1) X_{\text{DPDCH}}(k) \\ & & + \beta_c^2 X_{\text{DPCCH}}(k+1) X_{\text{DPCCH}}(k) \\ & & + j\beta_c\beta_d (X_{\text{DPCCH}}(k+1) X_{\text{DPDCH}}(k) \\ & & - X_{\text{DPDCH}}(k+1) X_{\text{DPCCH}}(k))\} + I(k) \end{array}$$

where $I(k)$ contains all the noise cross-product terms.
Let define the sequence $A(k) = C_1(2k+1)C_1(2k)C_2(2k)$. The scrambling code $S_n(k)$ satisfies the following relation (see Eq. (2)):

$$S_n(2k+1)S_n(2k)^* = -2jA(k)$$

Moreover, due to the properties of the channelization sequence C_d and C_c, the spread signals $X_{\text{DPDCH}}(k)$ and $X_{\text{DPCCH}}(k)$ satisfy the 3 relations:

$$\begin{array}{l} X_{\text{DPDCH}}(2k+1) X_{\text{DPDCH}}(2k) = 1 \\ X_{\text{DPCCH}}(2k+1) X_{\text{DPCCH}}(2k) = 1 \\ X_{\text{DPCCH}}(2k+1) X_{\text{DPDCH}}(2k) = X_{\text{DPDCH}}(2k+1) X_{\text{DPCCH}}(2k) \end{array}$$

Exploiting these properties, if the differential signal $U(k)$ is decimated, the negative value of its imaginary part satisfies :

$$V(k) = -\text{Im}\,(U(2k)) = 2(\beta_d^2 + \beta_c^2)A(k) + w(k) \quad (5)$$

The variable $w(k)$ contains all the noise cross-product terms. From the construction of the scrambling code detailed in the previous section, it is straight forward to observe that $A(k)$ is the BPSK representation of a binary sequence $a(k)$ which depends on sequence x_n and y (see Eq. 2):

$$a(k) = \tilde{x}(k) \oplus \tilde{y}(k) \quad (6)$$

with

$$\begin{array}{l} \tilde{x}(k) = x_n(2k) \oplus x_n(2k+1) \oplus x_n(2k+16777232) \\ \tilde{y}(k) = y(2k) \oplus y(2k+1) \oplus y(2k+16777232) \end{array}$$

Using the result of Eq. (3), we obtain:

$$\begin{array}{rcl} \tilde{x}(k) & = & x_n(2k) \oplus x_n(2k+1) \oplus x_n(2k+4) \\ & & \oplus\ x_n(2k+7) \oplus\ x_n(2k+18) \\ \tilde{y}(k) & = & y(2k) \oplus y(2k+1) \oplus y(2k+4) \oplus y(2k+6) \\ & & \oplus\ y(2k+17) \end{array} \quad (7)$$

From the "shift and add" and decimation properties, $\tilde{x}$ and $\tilde{y}$ are shifted version of the m-sequences x_n and y. The sequence a is thus a Gold sequence [14]. As a consequence, Eq. (5) tells us that $V(k)$ is the observation of the BPSK representation of a Gold sequence with additive noise. It is known from the literature that it is possible to estimate the initial state of a LFSR sequence with a standard iterative message-passing algorithm [11][15][16]. The decoding strategy proposed in this paper is however different.
In order to identify the initial state of the sequence x_n, the proposed algorithm exploits the a priori knowledge of the initial state of the sequence y at the beginning of each frame (all '1'). At each time instant q, the receiver assumes that it is synchronized with the beginning of the frame. It is thus possible to generate the sequence $\tilde{y}$ by using (3). Then, the elements of vector $(V(q), \cdots, V(q+M-1))$ are multiplied chip by chip with sequence $\tilde{Y}$, the BPSK representation of the sequence $\tilde{y}$:

$$R_q(k) = V(k+q)\tilde{Y}(k) \quad k = 0, \ldots, M-1 \quad (8)$$

This operation eliminates $\tilde{y}(k)$ from (6), and hence $R_q(k)$ is a noisy observation of the BPSK representation of sequence

$\tilde{x}$.
Since $\tilde{x}$ is a shifted version of sequence x, it can be decoded with an iterative message-passing algorithm which parity check matrix is matched to the sequence x (see section II-B). The vector $(R_q(0), \cdots, R_q(M-1))$ feeds this decoder, and if it fails to find a codeword, this means that either the frame synchronization assumption is not valid or a decoding failure happened. In both cases, the procedure is restarted when the next chip is received. If the decoder finds a valid codeword, the initial state of sequence x_n is found by a simple matrix multiplication (see section II-C).

B. Iterative message-passing decoding

A m-sequence x 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 primitive polynomial $g(D)$:

$$H = \begin{bmatrix} g_r & \cdots & g_0 & 0 & \cdots & \cdots & 0 \\ 0 & g_r & \cdots & g_0 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & g_r & \cdots & g_0 & 0 \\ 0 & \cdots & \cdots & 0 & g_r & \cdots & g_0 \end{bmatrix} \quad (9)$$

Once the parity check matrix has been defined, it is possible to decode the received vector $(R_q(0), \cdots, R_q(M-1))$ 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 greatly the decoder performance. If $g(D)$ is the sequence polynomial in GF(2), 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]:

$$H_{\text{RGM}} = \begin{bmatrix} H_0 \\ H_1 \\ \vdots \\ H_{n_{\text{RGM}}-1} \end{bmatrix} \quad (10)$$

where n_{RGM} is the number of RGMs used for decoding. These RGMs increase the column weight of the parity check matrix, while keeping constant the row weight.
If the decoding is successful (all parity check equations are satisfied), the soft decision output of the decoder is converted into a binary representation with a hard decision rule. Then, according to the Fibonacci representation (Fig. 2), the first r bits of the codeword represents the content of the shift registers at initialization.

C. Determination of the initial state of the sequence x_n

The decoder provides a vector of r bits representing the initial state of sequence $\tilde{x}(k)$, denoted by $Q_{\tilde{x}}$. We want to find the initial state of sequence $x_n(k)$: $Q_{x_n} = (n_0, n_1, \cdots, n_{24})^T$ knowing that $n_{24} = 1$. This task is achieved in 2 steps. The first step consists in finding the initial state of the sequence $x_{\text{decim}}(k)$, $Q_{x_{\text{decim}}}$, which gives $\tilde{x}$ by decimation by a factor 2:

$$\tilde{x}(k) = x_{\text{decim}}(2k)$$

[19] shows that there is a fixed transposition matrix B between these two vectors and describes the method to compute this matrix:

$$Q_{x_{\text{decim}}} = B Q_{\tilde{x}}$$

The second step eventually provides vector Q_{x_n}. It exploits the "shift and add" relation between the sequences x_{decim} and x_n (see Eq (7)):

$$x_{\text{decim}}(k) = x_n(k) \oplus x_n(k+1) \oplus x_n(k+4) \oplus x_n(k+7) \oplus x_n(k+18)$$

Using the state transition matrix G [12], defined by :

$$G = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & 0 & \cdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 \\ 1 & g_{r-1} & g_{r-2} & g_{r-3} & \cdots & g_1 \end{bmatrix} \quad (11)$$

we have:

$$Q_{x_n} = (I_r + G + G^4 + G^7 + G^{18})^{-1} Q_{x_{\text{decim}}}$$

where I_r is the $r \times r$ identity matrix. These two steps are finally combined in a unique transposition matrix T:

$$Q_{x_n} = T Q_{\tilde{x}}$$

$$T = (I_r + G + G^4 + G^7 + G^{18})^{-1} B$$

It is important to note that matrix T shall be computed once for all and stored in memory.

D. Flow chart of the algorithm

A flow chart of the algorithm is presented in Fig. 3. The decoding operation must be implemented at the chip rate, which requires a very high data rate decoding capability.

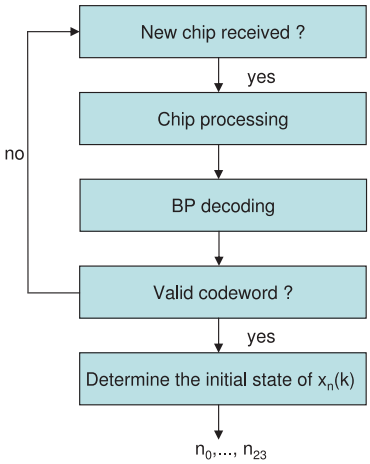

Fig. 3. Flow chart of the algorithm

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-by-chip multiplication with sequence $Y(k)$ in (8). A multipath component delayed by more than one chip will be scrambled by sequence $y(k)$ 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:

$$\begin{aligned} P_d &= P(Ic = 1 \text{ and } \hat{Q}_{x_n} = Q_{x_n} | H_1) \\ P_{fa} &= P(Ic = 1 | H_0) \\ P_m &= 1 - P_d \end{aligned}$$

Ic is the indication function of the decoder:

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

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

Performances are measured with the following simulation configurations:

- The transmitter sends a UL reference measurement channel for the 12.2 kbps service, as specified in [20]: $SF_{\text{DPDCH}} = 64$, $SF_{\text{DPCCH}} = 256$, $\beta_c = 1$ and $\beta_d = 11/15$.
- 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 Min-Sum (MS) message-passing algorithm [18]. The decoder stops when either all the parity check equations are satisfied or the maximum number of iteration N_{iter} is reached.

A. AWGN channel

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

Fig. 4 shows the probability of missed detection as a function of the number of RGMs, for $M = 4000$ variables at the decoder input and $N_{iter} = 20$ iterations at most. P_m is measured as a function of the Signal to Noise ratio (SNR) at the input of the receiver. For an arbitrary defined target $P_m = 0.1$, a gain of about 7 dB is obtained with 7 RGMs with respect to the case with only one graphical model. With 7 RGMs, the target $P_m = 0.1$ is reached at a SNR around -7 dB. According to [20], a UE transmitting a 12.2 kbps service in an AWGN channel shall be received at its serving BS with a SNR larger than -17 dB. Hence, if it is received at the femto BS at a SNR equal to -7 dB, its power is 10 dB larger than a femto UE transmitting the same service. This macro UE is thus a strong interferer, but its scrambling code index can be detected and it can be mitigated by an appropriate interference cancelation algorithm.
Simulations not reported in this paper have also shown that there is no gain to have more than $N_{iter} = 20$ iterations in the decoding algorithm.
Fig. 5 shows the sensitivity of the algorithm to the number of variables M (see (8)). There is no interest to increase the number of variables above 4000 chips. A gain of 0.5dB is achieved between $M = 2000$ and $M = 4000$. Since this gain is small, this leaves some room for a performance/complexity trade-off.

B. Multipath channel

The probability of detection is defined by the probability to detect the signal when the search window contains the beginning of the frame (see Section II-E). The performance are evaluated with 2 static multipath channels (Static A and B), having respectively 2 and 3 multipath with equal gains. The delays of the multipath are equal to $\{0, 2\}$ and $\{0, 2, 4\}$ chips for Static A and B channels. The search widow is set to $W = 10$ chips, which is larger than the channel delay spread. Fig. 6 compare the performance of the detection algorithm with AWGN, Static A and B channels. Compared to a gaussian channel, there is a degradation of about 3 dB and 8 dB at $P_m = 0.1$ for Static A and B. This is due to the diminution of the SNR per multipath. Even if the degradation is noticeable, the algorithm is still operational when the channel contains multipath.

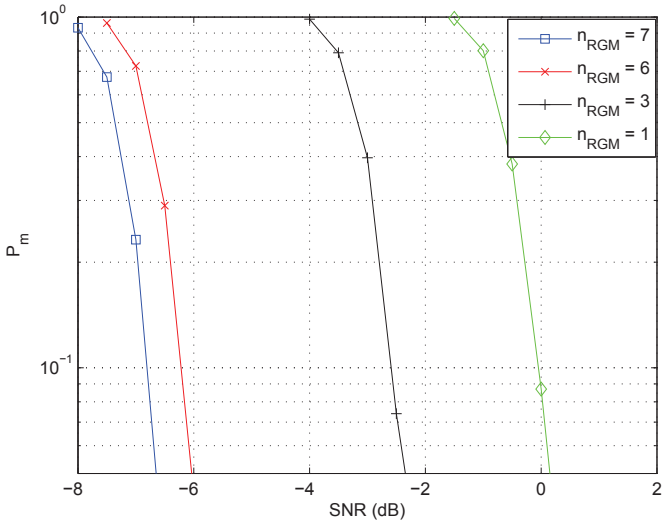

Fig. 4. Sensitivity to the number of RGM - AWGN channel - $N_{iter} = 20$ and $M = 4000$

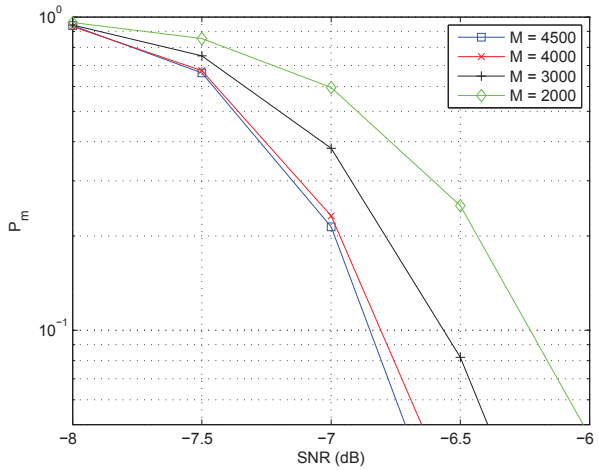

Fig. 5. Influence of the number of variables M - AWGN channel - $N_{iter} = 20$ and $n_{\text{RGM}} = 7$

IV. Conclusion

An algorithm which performs a blind identification of the uplink scrambling code of a WCDMA transmission has been presented. It exploits the framing, spreading and multiplexing specificitions of the WCDMA standard and is thus dedicated to this system. The simulation results show that it is possible to obtain a reliable estimation at a SNR as small as -7 dB in an AWGN channel. Thus, the scrambling code of a strong interferer can be identified at the femto BS, and interference mitigation techniques can be implemented to cancel this interferer. The robustness of the algorithm to static multipath has also been validated.

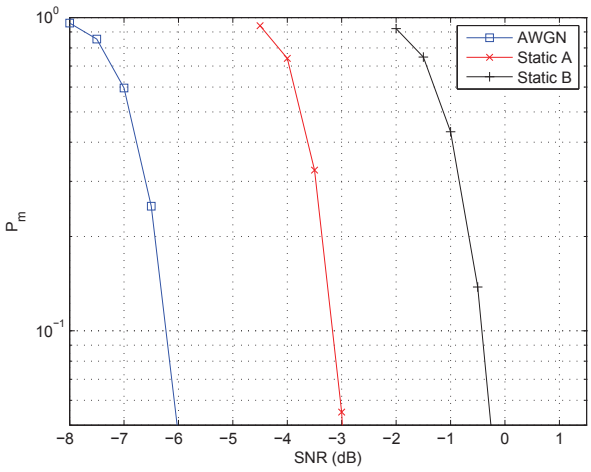

Fig. 6. Performance with multipath channels - M = 4000

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

[8] 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.

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

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

[14] R. Gold, "Optimal binary sequences for spread spectrum multiplexing," *IEEE Trans. on Information Theory*, vol. IT. 13, no. -, pp. 619–621, 1967.

[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] *TS25.104 v.4.9.0 BS Radio Transmission and Reception (FDD)*, 3GPP, 2007.