Performance Enhancements to the UMTS (W-CDMA) Initial Cell Search Algorithm

Mario Kiessling, Syed Aon Mujtaba

Wireless Systems Research Department,
Agere Systems,
600 Mountain Avenue,
Murray Hill, NJ 07974

Abstract – **The purpose of the cell search algorithm in UMTS is to estimate the spreading code of the serving base-station and its corresponding timing offset. The search procedure consists of 3 sequential and distinct stages: (1) slot-boundary synchronization, (2) frame-boundary synchronization with code-group identification, and (3) scrambling code identification. Algorithms that have appeared in literature for cell search have confined the "code-time" estimation in each stage to a single hypothesis. Furthermore, these algorithms have neither considered the benefits of oversampling, nor the detrimental effects of non-ideal sampling that may arise due to clock jitter and/or residual frequency offset.**

In this paper, we study the performance benefits of estimating multiple "code-time" hypotheses in each stage of the cell-search process. In addition, we also study the effect of oversampling and non-ideal sampling. Our results indicate that, in the presence of non-ideal sampling, performance improves significantly if the received signal is oversampled by a factor of 4 or more. We also show that estimating 4 "code-time" hypotheses instead of 1 in the cell-search stages reduces the search-time (i.e. the code-acquisition time) considerably, in particular at low SINR.

I. Introduction

In the asynchronous W-CDMA system [1], each base-station is identified by a unique scrambling code. The mobile station has to synchronize to the scrambling code of the serving base station in order to descramble the downlink traffic channels. The synchronization process is commonly referred to as the cell search procedure [2][3][5][6][11].

A three-step hierarchical cell search process has been introduced in the UMTS standard that is supported by several auxiliary synchronization channels. These include the Primary Synchronization Channel (P-SCH), the Secondary Synchronization Channel (S-SCH), and the Common Pilot Channel (CPICH). The cell search procedure is split into three stages – stage 1 performs slot synchronization, stage 2 performs frame synchronization and scrambling code group identification, and stage 3 acquires the cell-specific scrambling code. In this paper, we address the initial cell search procedure (as opposed to the target cell search), which is carried out when the mobile station is switched on. It is assumed that the mobile station has no preliminary information about the scrambling code of the serving cell, and a frequency offset of 20 kHz or 10 ppm at a carrier frequency of 2 GHz is present (worst case scenario). An additional DFT stage following stage 3 can bring down the frequency offset to 200 Hz with a high degree of reliability [4].

In our study, we focus on minimizing the acquisition time while maintaining a given probability of false alarm. Prior work presented in [2][3][5][6] has considered chip-rate sampling at the receiver (i.e. an oversampling ratio of 1), and neglected the impact of non-ideal sampling that may arise due to clock jitter and/or residual frequency offset. Furthermore, performance results from these studies were obtained with 1 time-code candidate passed between the various stages of the cell search process. In this study, we extend the previously published results by investigating the impact of the following factors:

- oversampling of the received signal
- non-ideal sampling of the received signal
- passing multiple time-code candidates between the cell search stages

We consider coherent detection in stage 2, and assume that a frequency offset of 20KHz is present in the received signal. Our simulation results show that considerable performance improvements can be realized by estimating 4 code-time hypotheses with an oversampling factor of 4 or more, in particular at low SINR.

II. Cell Search Overview

The cell search procedure is supported by three channels (Fig. 1): *Primary Synchronization Channel* (P-SCH), *Secondary Synchronization Channel* (S-SCH), and *Common Pilot Channel* (CPICH) [9]. At the beginning of each slot (1 slot=2560 chips), the same 256-chip long *Primary Synchronization Code* (PSC) is transmitted on the P-SCH. Slot synchronization at the mobile is achieved by correlating the received signal with the PSC.

There are 16 unique *Secondary Synchronization Codes* (SSC), each 256 chips in length, that appear on the S-SCH. A sequence of 15 SSCs – one in each slot – creates a code-group (or code-word). A total of 64 code-groups are constructed such that their cyclic shifts are unique, both to itself and to other code-groups. Thus, the start of a frame (= 38400 chips = 10 ms) can be detected by identifying the prevailing code-group on the S-SCH. This is achieved by a slot-wise correlation of the S-SCH with the SSCs, where the slot boundary is obtained from stage 1.

The 512 primary scrambling codes have been divided into 64 code-groups (defined in [10]), with 8 scrambling codes in

0-7803-7400-2/02/$17.00 © 2002 IEEE

each group. Each S-SCH code-word corresponds to a code-group, and hence identification of the S-SCH sequence can be jointly performed with frame boundary synchronization.

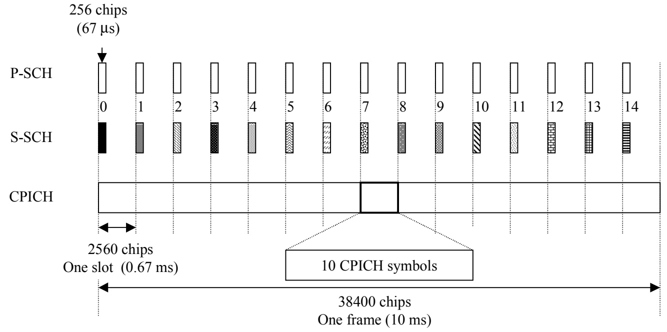


Fig. 1: Synchronization channels

Unlike the P-SCH and S-SCH, the CPICH is scrambled with the primary scrambling code. One slot of the CPICH is subdivided into 10 symbols, each 256 chips long. There are a total of 150 symbols in the frame. Once the code-group has been identified in stage 2, the scrambling code can be identified in stage 3 by correlating with 8 possible scrambling code candidates. TABLE I gives an overview of the code and timing candidates that are estimated and returned by current techniques in each stage.

TABLE I
CURRENT TECHNIQUES

	Timing Candidates	Code Candidates
Stage 1	1 of 2560 (slot boundary)	--
Stage 2	1 of 15 (frame boundary)	1 code-group of 64
Stage 3	1 of 1 (frame boundary)	1 of 8 scrambling codes

Our proposed cell search process is detailed in TABLE II. Extending the procedure above, we pass to stage 2, *C* slot boundary candidates corresponding to the top *C* candidates from a rank-ordered list of P-SCH correlation values. Furthermore, we assume oversampling of the received signal by O_s. Oversampling increases the timing resolution and the robustness against errors due to non-ideal sampling at the receiver (see also Fig. 2). Since stage 1 determines the timing resolution for the following stages, processing in stages 2 and 3 can be performed at the chip rate. A reduction in the sampling rate in stages 2 and 3 clearly reduces the computational complexity of our proposed scheme.

TABLE II
PROPOSED TECHNIQUE

	Timing Candidates	Code Candidates
Stage 1	C of O_s×2560 (slot boundary)	--
Stage 2	C of C×15 (frame boundary)	C code groups of C×64
Stage 3	1 of C (frame boundary)	1 code of C×8

C code-group and frame-boundary candidates (corresponding to the slot boundary candidates detected in stage 1) are passed from stage 2 to stage 3. It is shown in the remainder of this paper that the proposed scheme has the potential to improve the performance of the cell search algorithm, especially at low SINR. We should mention that a more flexible scheme would allow the number of time-code candidates between stage 1 and 2 to differ from that between stage 2 and 3.

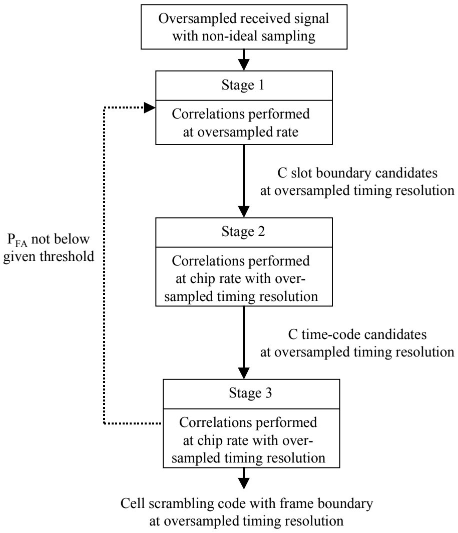

Oversampled received signal with non-ideal sampling
Stage 1
Correlations performed at oversampled rate
C slot boundary candidates at oversampled timing resolution
P_{FA} not below given threshold
Stage 2
Correlations performed at chip rate with over-sampled timing resolution
C time-code candidates at oversampled timing resolution
Stage 3
Correlations performed at chip rate with over-sampled timing resolution
Cell scrambling code with frame boundary at oversampled timing resolution

Fig. 2: Information flow between stages

The duration, i.e. the number of correlation windows, of all three stages is chosen to be a constant at N_t slots, which is set to 15 slots (= 1 frame) in all our simulations.

III. Code Acquisition

We use the algorithms for code acquisition presented in [2][3] as baseline for benchmarking our enhanced algorithms. Since the frequency acquisition stage presented in [4] can be used without modification after stage 3, we do not consider it in this paper. In our study, we assume an oversampled received signal at the mobile. However, as was mentioned above, after slot synchronization in stage 1, the signal is down-sampled to chip-rate for further processing.

A. Slot Synchronization

Conventional detection of the slot boundary entails: (a) correlating the received data over 256 chips with the PSC (there are a total of 2560*O_S correlation hypotheses), (b) then performing this correlation over N_t slots, (c) then accumulating all the N_t correlation values, and (d) finally selecting the hypothesis that corresponds to the maximum correlation value. From Table II, the total number of timing hypotheses for the slot boundary are *Q=2560*O_S* , where O_S is the over-sampling ratio. For each timing hypothesis *k*, the correlator output is calculated for each time slot *s* as:

$$y_k^s = \left| \sum_{l=0}^{255} r_{k+O_s l}^s \left[c_l^{(0)} \right]^* \right| \quad , \quad k = 0,1,\ldots,Q-1 \qquad (1)$$
$$s = 0,1,\ldots,N_t - 1$$

where r_k^s is the k:th chip in the s:th slot of the received baseband signal (including noise and interference). The PSC c_p is indexed by the superscript 0, and the subscript corresponds to the chip values $c_p = (c_0^{(0)}, c_1^{(0)}, \ldots, c_{256}^{(0)})$. The individual time slot metrics y_k^s are then averaged over N_t

slots to further minimize noise variance, and the following decision metric is obtained:

$$Y_k = \sum_{s=0}^{N_t-1} y_k^s \quad , \quad k = 0, 1, \ldots, Q-1 \tag{2}$$

Conventionally, the slot boundary candidate is obtained by selecting the maximum decision metric:

$$k_{slot} = \arg\max(Y_k) \tag{3}$$

However, in our analysis, we select a vector of decision metrics corresponding to the *C* most-likely decision metrics:

$$\vec{k}_{slot} = (k_{slot}^0, k_{slot}^1, \ldots, k_{slot}^{C-1}) \tag{4}$$

Probability of error can be decreased further by increasing the stage duration N_t , but that results in longer acquisition times. Another limiting factor is the drift of the free running sampling clock in a real system, which leads to errors in the accumulation process and a broadening of the maximum of the detection metric.

B. *Analytical Results for Stage 1*

Performance of the search procedure can be evaluated analytically using state diagram techniques with a modification that takes into account maximum search instead of threshold setting as proposed in [5]. We outline the computation of the error probability of stage 1. Assuming that Y_0 corresponds to the metric of the correct slot boundary, we get:

$$\begin{aligned} Y_0 &\approx \sum_{s=0}^{N_t-1} \left| X_s + n_s^0 \right|^2 \quad (\textit{correct hypothesis}) \\ Y_k &\approx \sum_{s=0}^{N_t-1} \left| n_s^k \right|^2 \quad (\textit{incorrect hypothesis}) \end{aligned} \tag{5}$$

where X_s is the signal component at the correlator output in the s:th slot, and n_s^k is the total interference plus noise at the correlator output. To simplify the analysis, noise is assumed to be an i.i.d. complex Gaussian process with unit variance. As a result, X_s becomes a random variable with variance E_s/I_o, where E_s is the average symbol energy, and I_o is the noise plus interference power.

In the following derivations, we assume that the autocorrelation of the PSC is a delta function. Simulations show that this simplification has only a minor influence on the results.

It is well known that Y_k (noise plus interference only metric) is chi-squared distributed with $2N_t$ degrees of freedom

$$p(y_k) = \frac{1}{\Gamma(N_t)} y_k^{N_t-1} \exp[-y_k]. \tag{6}$$

For Y_0 we get different distributions depending on the channel model. In case of an AWGN channel we get a non-central chi-squared distribution with $2N_t$ degrees of freedom

$$p(y_0) = \left(\frac{y_0}{E_s/I_0} \right)^{\frac{N_t-1}{2}} \exp[-(E_s/I_0 + y_0)] I_{N_t-1}\left(2\sqrt{E_s/I_0}\sqrt{y_0}\right) \tag{7}$$

where $I_x(.)$ is Bessel function of the first kind with order x. For a 1-tap channel that is independent Rayleigh faded (such that different correlation blocks undergo independent Rayleigh fading), the X_s is i.i.d. Gaussian, and we get the chi-squared distribution with $2N_t$ degrees of freedom:

$$p(y_0) = \frac{1}{(1 + E_s/I_0)^{N_t} \Gamma(N_t)} y_0^{N_t-1} \exp\left[\frac{-y_0}{E_s/I_0} \right] \tag{8}$$

where Γ(.) is the gamma function. Finally, for 1-tap block Rayleigh fading (i.e. constant channel gain over the entire interval of N_t slots) we arrive at [5]:

$$p(y_0) = \int_0^\infty \exp[-(N_t x + y_0)] I_0\left(\sqrt{N_t y_o x}\right) \frac{e^{-\frac{x}{E_s/I_0}}}{E_s/I_0} dx . \tag{9}$$

Given the probability density functions, the probability of error, P_e (i.e. detection of a wrong slot boundary), of stage 1 can be calculated. With one timing candidate (C=1), the probability of correct detection, P_d=1-P_e , is given by:

$$\begin{aligned} P_d^{(1)} &= P(y_k < y_0, \quad 1 \le k \le Q-1) \\ &= \int_{-\infty}^{\infty} p(y_0) [P(y_k < y_0)]^{Q-1} dy_0 \end{aligned} \tag{10}$$

Evaluation of (10) requires numerical integration. Theoretical results agree well with simulations, as shown in Fig. 3 for the AWGN channel at chip rate sampling without interference.

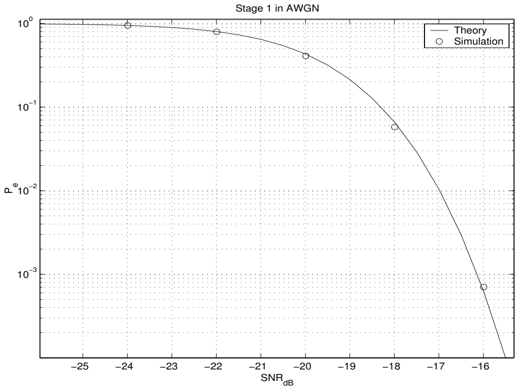


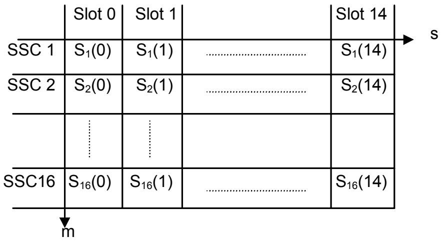
Fig. 3: Stage 1 in AWGN

The analysis presented above can be extended to account for several timing candidates. If we define the probability of correct detection in stage 1 as the probability that the slot boundary is a member of the list of C maximum metrics, i.e. there are at most C-1 metrics y_k larger than y_0, we get:

$$\begin{aligned} P_d^{(C)} &= P(y_k < y_0, \quad 1 \le k \le Q-C) \\ &= \int_{-\infty}^{\infty} p(y_0) [P(y_k < y_0)]^{Q-C} dy_0 \end{aligned} \tag{11}$$

C. *Code Group and Frame Boundary Detection*

For each of the *C* slot boundary candidates selected in stage 1, the mobile performs the following operations. With 64 possible code-groups and 15 cyclic shifts corresponding to 15 frame boundary hypotheses, there are a total of 960 time-code hypotheses in the detection process of stage 2. In a brute force approach, the mobile station would correlate the slot-aligned chip-rate received signal with each of the code-groups and its 15 corresponding cyclic shifts. As there are only 16 possible SSCs in each slot, the mobile performs 16×15 = 240 correlations per frame, as shown in Fig. 4.



	Slot 0	Slot 1		Slot 14
SSC 1	$S_1(0)$	$S_1(1)$	....	$S_1(14)$
SSC 2	$S_2(0)$	$S_2(1)$	....	$S_2(14)$
	⋮	⋮		
SSC16	$S_{16}(0)$	$S_{16}(1)$	....	$S_{16}(14)$

Fig. 4: Secondary Synchronization “Code-Time” Matrix

The correlations with the SSC are performed at chip rate (r_j^s is assumed to be slot-aligned and decimated to chip rate, i.e. j=0 corresponds to the slot boundary estimated in stage 1):

$$S_m(s) = \left| \sum_{j=0}^{255} r_j^s c_j^{(m)} \right| \quad , \quad m = 1, 2, \ldots, 16 \qquad (12)$$
$$s = 0, 1, \ldots, 14$$

where the 16 SSCs are indexed by the superscript *m*:

$$c_s^m = (c_0^{(m)}, c_1^{(m)}, \ldots, c_{256}^{(m)}) \, , \quad m = 1, 2, \ldots, 16 \qquad (13)$$

The 960 “time-code” hypotheses are computed [3] using the unique 240 $S_m(s)$ values and the S-SCH code-groups defined in the standard [10]. Finally, a maximum search yields the desired candidate(s).

Due to correlations between the different code-groups, the i.i.d. assumption of the code-group metrics is not valid, and the error probability of stage 2 has to be obtained through simulations. However, as an approximation, the techniques introduced for the performance evaluation of stage 1 can be applied to stage 2.

D. Scrambling Code Identification

A simple threshold based scheme is applied in stage 3. The mobile performs the following computations for each time-code candidate from stage 2 [3]. The frame-aligned CPICH at chip rate is despread one-symbol-at-a-time with each of the eight primary scrambling codes from the code-group estimated in stage 2. Using union bound techniques, the decision threshold is adjusted based on the target-false-alarm-probability. This scheme avoids calculating the SINR for the threshold setting while providing sufficient performance. The final step is to select the scrambling code candidate with the maximum metric that fulfills the threshold criteria.

IV. SIMULATION RESULTS

In the simulation results that follow, we study the system performance in terms of overall acquisition time T_a. The stage duration is held constant at N_t=15 (= 1 frame). The target false alarm probability is chosen to be $P_{fa}=10^{-4}$ for all simulations. We assume a frequency offset due to receiver oscillator inaccuracies of 20 kHz. Non-ideal sampling is introduced by means of a fractional delay filter that introduces a delay of half a sampling period (worst case). The advantage of oversampling becomes evident in Fig. 5 for propagation conditions corresponding to case I [8]. Chip rate sampling leads to unacceptable performance at lower SNR. Clearly, oversampling in the presence of non-ideal sampling reduces T_a dramatically. Furthermore, simulation results show a saturation of performance with increasing oversampling factors. Therefore, we assume an oversampling factor of O_s=4 in the remainder of the paper.

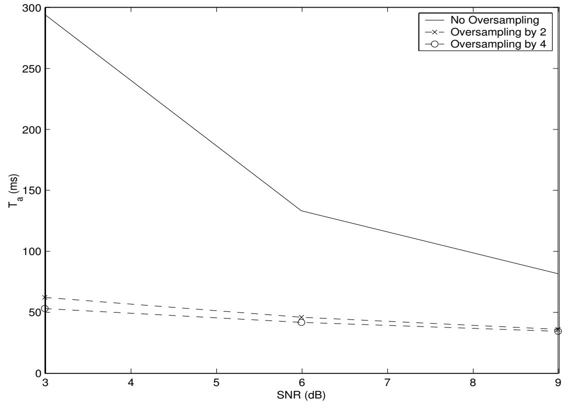


Fig. 5: Effect of oversampling (N_t=15, case I, non-ideal sampling, $P_{fa}=10^{-4}$)

The dependence of the acquisition time T_a on the number of slot boundary candidates passed between stage 1 and stage 2 is illustrated in Fig. 6 for propagation conditions corresponding to case II [8]. For this simulation result, the number of candidates between stage 1 and 2 is equal to the number between stage 2 and stage 3.

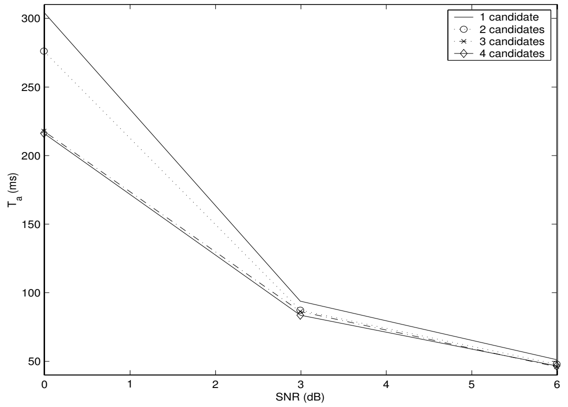


Fig. 6: Influence of the number of candidates (O_s=4, N_t=15, case II, ideal sampling, $P_{fa}=10^{-4}$)

Clearly, the performance gain obtained by passing several candidates between stages increases at lower SNR, and it reaches 30% at 0dB compared to the case with a single time-code candidate. Furthermore, no noticeable performance improvements are observed beyond 3 time-code candidates. Similar performance gains are observed in Fig. 7, which considers non-ideal sampling.

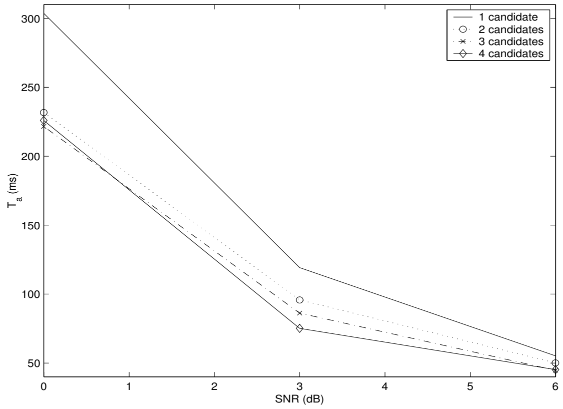


Fig. 7: Effect of the number of time-code candidates (O_s=4, N_t=15, case II, non-ideal sampling, P_{fa}=10^{-4})

The simulation results for ideal and non-ideal (with delay) sampling for the propagation case I [8] are summarized in Fig. 8. Compared to case II, the performance gain is smaller in case I. This can be explained by the fact that there is only 1 dominant propagation path in case I compared to 3 paths in case II. Furthermore, we note that the performance with non-ideal sampling and 4 time-code candidates is better than with ideal sampling at lower SNR values. Sampling at ½ a sampling period before and after the ideal sampling instant leads to two samples that have slightly less SNR than the peak value, but these 2 samples achieve greater immunity against fading by offering a 2-fold diversity.

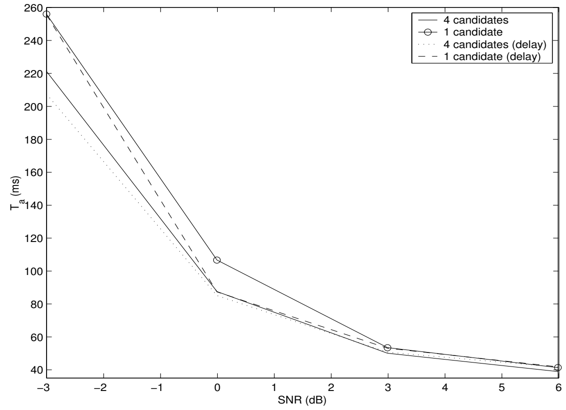


Fig. 8: Case I (O_s=4, N_t=15, P_{fa}=10^{-4})

With only one main multipath component and weak secondary paths in propagation scenario case III [8], the cell search algorithm is very sensitive to non-ideal sampling at low SNRs (Fig. 9). In this case, 4 code-timing candidates reduce the acquisition time by 50%. Assuming ideal sampling, the gain is still about 30%.

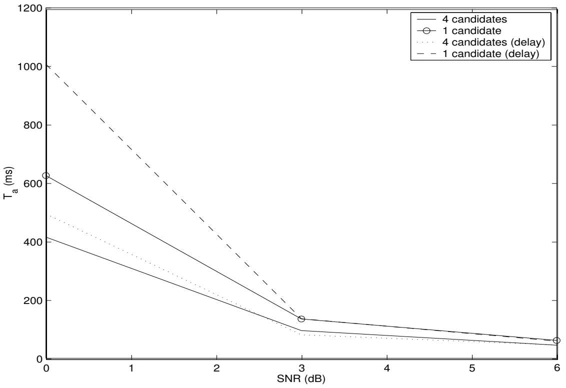


Fig. 9: Case III (O_s=4, N_t=15, P_{fa}=10^{-4})

V. Conclusions

Our study has shown that oversampling of the received signal can have a significant impact on the cell search performance in the presence of non-ideal sampling. We found that an oversampling factor of 4 was sufficient to mitigate the detrimental effects of non-ideal sampling, whereas chip-rate sampling leads to unacceptable performance. Furthermore, it was shown that the performance of the cell search algorithm in 3GPP UMTS can be improved significantly by passing several "code-time" candidates between the three stages of the hierarchical procedure. Our results show that with 4 candidates, saturation-performance is achieved for the propagation scenarios defined by the standard. Depending on the scenario, acquisition time can be reduced by up to 50% at low SNR values compared to the single candidate case.

References

[1] H. Holma, A. Toskala, "WCDMA for UMTS", John Wiley, © 2000

[2] Y.-P. E. Wang, T. Ottoson, "Cell search algorithms and optimization in W-CDMA", *IEEE VTC,* pp. 81-86, 2000

[3] Y.-P. E. Wang, T. Ottoson, "Cell search in W-CDMA", *IEEE JSAC,* vol. 18, no. 8, pp. 1470-1482, August 2000

[4] Y.-P. E. Wang, T. Ottoson, "Initial frequency acquisition in W-CDMA", *Proc. IEEE VTC,* Amsterdam, pp. 1013-1017, Sep. 1999

[5] S. Sriram, S. Hosur, "An analysis of the 3-stage search process in W-CDMA", *Proc. IEEE VTC 2000,* Boston.

[6] A. O. Nielsen, S. Korpela, "WCDMA initial cell search", *Proc. IEEE VTC 2000,* Boston.

[7] "3G TS 25.101 – Technical specification group radio access networks: UE radio transmission and reception (FDD)", www.3GPP.org, Release 1999.

[8] "3G TS 25.104 – Technical specification group radio access networks: UTRA FDD Radio Transmission and Reception (FDD)", www.3GPP.org, Release 1999.

[9] "3G TS 25.211 – Technical specification group radio access networks: physical channels and mapping of transport channels onto physical channels (FDD)", www.3GPP.org, Release 1999.

[10] "3G TS 25.213 – Technical specification group radio access networks: spreading and modulation (FDD)", www.3GPP.org, Release 1999.

[11] "3G TS 25.214 – Technical specification group radio access networks: physical layer procedures (FDD)", www.3GPP.org, Release 1999.