Performance Analysis of 3-Stage Cell Search Process in WCDMA System

Shailendra Mishra, Dehradun Institute of Technology, Dehradun, & Uttaranchal Technical University, India; E-mail: skmishra1@gmail.com
Nipur Singh, KGM, II Campus, Gurukul Kangari University, India; E-mail: nipursingh@hotmail.com

ABSTRACT

In this paper, we study the performance benefits of cell search algorithm. 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. Also, 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 an estimating 4 "codetime" hypothesis instead of 1 in the cell-search stages reduces the search-time (i.e. the code-acquisition time) considerably, in particular at low SNR.

Keywords: WCDMA, UMT, SSC, PSC, FHT, CSD,P_{FA}, V_{TH}

1. INTRODUCTION

WCDMA is a wideband Direct Sequence Code Division Multiple Access (DS-CDMA) system, which means that the user information bits (symbols) are spread over a wide frequency bandwidth by multiplying the user data bits with a spreading code sequence of "chips" [1], [2]. In the asynchronous W-CDMA system 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 [3],[4]. The synchronization process is commonly referred to as the cell search procedure, i.e. cell search is the process of synchronization between the mobile and the base station. 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. A three-step hierarchical cell search process has been introduced in the UMTS standard that is supported by several auxiliary synchronization channels [5]. These include the Primary Synchronization Channel (P-SCH), the Secondary Synchronization Channel (S-SCH), and the Common Pilot Channel (CPICH) [9].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.

2. CELL SEARCH ALGORITHM

We use the algorithms for code acquisition presented in [4],[5],[6] as baseline for benchmarking our enhanced algorithms. Since the frequency acquisition stage presented in [6] can be used without modification after stage 3, 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.

The cell search algorithm is divided into the following stages:

(1) Slot boundary synchronization,
(2) Frame synchronization and code group identification,
(3) Scrambling code identification,
(4) Frequency acquisition, and,
(5) Cell identification.

The last two stages need to be performed only during initial cell search [10].

Stage 1: Slot Boundary Synchronization

During stage 1 of the cell search algorithm the mobile station uses the Primary Synchronization Code (PSC) to acquire slot synchronization to a cell. This is typically done with a single matched filter matched to the PSC which is common to all cells. The slot timing of the cell can be obtained by detecting peak values in the matched filter output. However, decisions based on observations over a single slot may be unreliable, when the signal-to-noise ratio (SNR) is low or if fading is severe. Reliable slot synchronization is required to minimize cell search time. In order to increase reliability, observations are made over multiple slots and the results are then combined. This ensures that the correct slot boundary is identified.

Stage 2: Frame Synchronization and Code Group Identification

During stage 2 of the cell search algorithm, the mobile station uses the Secondary Synchronization Code (SSC) to achieve frame synchronization and identify the code group of the cell found in stage 1. This is done by correlating the received signal with all possible SSC sequences and identifying the maximum correlation value. Since the cyclic shifts of the sequences are unique, the code group as well as the frame synchronization is determined in this stage.

Stage 3: Scrambling Code Identification

During stage 3 of the cell search algorithm, the mobile station determines the exact primary scrambling code used by the cell. The primary scrambling code is typically identified through symbol-by-symbol correlation over the CPICH with all codes within the code group identified in stage 2. In this stage, a threshold value is used to decide whether the code has been identified. The threshold value can be predetermined using a parameter called probability of false alarm rate.

2.1 Slot Synchronization

Conventional detection of the slot boundary entails:

(a) correlating the received data over 256 chips with the PSC,
(b) then performing this correlation over Nt slots, which is set to 15 slots (= 1 frame) in all our simulations ,
(c) then accumulating all the Nt correlation values, and
(d) finally selecting the hypothesis that corresponds to the maximum correlation value.

Matching Filter

A basic problem that often arises in study of communication systems is that of detecting a pulse transmitted over a channel that is corrupted by additive noise at the front of the receiver. The filter input X(t) consists of a pulse signal g(t) corrupted by additive noise w(t), as shown

$$X(t) = g(t) + w(t) \qquad 0 \le t \boxtimes T$$

Where T is an arbitrary observation interval. The pulse signal g(t) may represent a binary symbol 1 or 0 in a digital communication system . With the received signal X(t) used as filter input , the resulting filter output , y(t) is defined by the convolution integral :

$$(t) = \int_{-\infty}^{\infty} x(\tau)\, h_j(t-\tau)\, d\tau$$

Slot Matching Filter
The MS first needs to acquire the PSC which is common to all the BSs. These codes are of length 256 chips. The matched filter output is given by,

$$Y = \sum_{J=0}^{255} R_j \cdot C_{pj}$$

Where *Rj* is the jth sample of the received complex signal, and *Cpj* is the jth bit of the PSC.

PSC Sequence
The hierarchical sequences used for generating the PSC are constructed from two constituent sequences X1 and X2 of length n1 and n2, respectively, using the following equation [12]:

Cp (n) =X1(n mod n2) +X2(n div n1) modulo 2, n=0, 1,.., (n1*n2)-1

Where n1=n2=16. The constituent sequences X1 and X2 are both defined as:

X1=X2= (1, 1,-1,-1,-1,-1, 1,-1, 1, 1,-1, 1, 1, 1,-1, 1).

Slot Boundary Detection
A traditional matched filter implementation would require 256 taps and a large adder circuit. This would increase the delay as well as power consumption at the receiver which is not desirable. Thus, a hierarchical structure is used for performing the matched filter operations which will need lesser number of taps, reduced circuitry and lower power consumption. The hierarchical matched filter consists of two concatenated matched filter blocks. The first matched filter receives the input signals serially from the base station. After 16 clock cycles when the shift register 1 is filled, the data stored in the shift register 1 is matched in parallel with the code applied to the taps of the matched filter (tap coefficients). The tap coefficients are the PSC sequences which are the same for all the base stations [11],[12].Hence, the same matched filter structure can be used for all the base stations.

2.2 Frame Synchronization and Code Group Identification
During stage 2 of the cell search procedure, the MS uses the SCHs Secondary Synchronization Code (SSC) to achieve frame synchronization and identify the code group of the cell found in stage 1. This is done by correlating the received signal with all possible SSC sequences and identifying the maximum correlation value. Since the cyclic shifts of the sequences are unique, the code group as well as the frame synchronization is determined. The Secondary SCH consists of 15 sequences belonging to a family of cyclic codes (SSCs), each of length 256 chips. These SSCs are transmitted repeatedly in parallel with the Primary SCH. The procedure for constructing the cyclic codes is similar to that of the hierarchical sequence for the Primary SCH except that it uses specific sequences of length 16 from Table1 for each code group. The procedure for constructing the cyclic hierarchical sequence Csi,1 for slot 1 is exactly the same as constructing the hierarchical sequence Cp for the Primary SCH. The sequence Csi,1 for slot 1 will be referred to as the zero cyclic shift sequence as no shift is applied to the constituent sequence X1i. For slots 2 to 15, the cyclic codes are constructed from the two constituent sequences X1i,k-1 and X2i,k-1 of length n1 and n2 respectively using the following formula[12]

Csi,k(n)=X2i,k-1 (n mod n2)+X1i,k-1 (n div n1) modulo 2, n=0,1,..,(n1*n2)-1

Table 1. Sequences X1i and X2i for Code Groups 1 to 32

Code Group		Code Group	
1	1 1 1-1-1-1 1-1-1 1 1-1 1-1 1 1	17	1-1 1 1-1 1-1 1 1 1-1 1 1 1-1 1
2	1-1 1 1-1 1 1 1-1-1 1 1 1 1 1-1	18	1 1 1-1-1-1-1-1-1 1-1-1-1 1-1-1-1
3	1 1-1 1-1-1-1 1-1 1-1 1 1-1-1-1	19	1-1-1-1 1-1-1 1 1 1 1-1-1-1-1 1
4	1-1-1-1-1 1-1-1-1-1-1-1 1 1-1	20	1 1-1 1 1 1-1-1 1-1 1 1-1 1-1-1
5	1 1 1-1 1 1-1 1-1 1 1-1-1 1-1-1	21	-1-1-1 1 1-1-1 1 1-1 1-1 1 1-1-1
6	1-1 1 1 1-1-1-1-1-1 1 1-1-1-1 1	22	-1 1-1-1 1 1-1-1 1 1 1 1 1-1-1 1
7	1 1-1 1 1 1 1-1-1 1-1 1-1 1 1 1	23	-1-1 1-1 1-1 1-1-1 1 1-1-1-1-1-1
8	1-1-1-1 1-1 1 1-1-1-1-1-1-1 1-1	24	-1 1 1 1 1 1 1 1-1-1 1 1-1 1-1 1
9	1 1-1 1-1-1-1 1 1-1 1-1-1 1 1 1	25	-1 1 1 1-1-1 1 1 1-1-1-1-1-1-1 1
10	1-1-1-1-1 1-1-1 1 1 1 1-1-1 1-1	26	-1-1 1-1-1 1 1-1 1 1-1 1-1 1-1-1
11	-1 1-1-1-1-1-1 1 1 1 1-1 1-1 1-1	27	-1 1 1 1 1 1-1-1 1-1-1-1 1 1 1-1
12	-1-1-1 1-1 1-1-1 1-1 1 1 1 1 1 1	28	-1-1 1-1 1-1-1 1 1 1-1 1 1-1 1 1
13	1-1-1-1 1-1-1 1-1-1-1 1 1 1 1-1	29	-1 1-1-1 1 1 1 1 1-1 1 1 1 1-1 1
14	1 1-1 1 1 1-1-1-1 1-1-1 1-1 1 1	30	-1-1-1 1 1-1 1-1 1 1 1-1 1-1-1-1
15	1-1-1-1-1 1 1-1 1 1 1-1 1 1 1-1	31	-1 1 1 1-1-1 1 1-1 1 1 1 1 1 1-1
16	1 1-1 1-1-1 1 1 1-1 1 1 1-1 1 1	32	-1-1 1-1-1 1 1-1-1-1 1-1 1-1 1 1

where i is code group number, k=2,3,..,15 is slot number, n is chip number in slot, n1=n2=16, and the constituent sequences X1i,k-1 and X2i,k-1 in each code group i are chosen to be the following sequences from Table 1.The constituent sequence X2i,k-1 (inner sequence) is exactly equal to the base sequence X2i in every slot, i.e. X2i,k-1=X2i at all k.

The constituent sequence X1i,k-1 (outer sequence) are formed from the base sequence X1i by cyclic right shifts of X1i on k-1 positions (from 0 to 15) clockwise for each slot number k, from 1 to 15 . The generation of the cyclic codes can be understood clearly by considering the following example.

For the first code group the sequence is given by

X11,0=(1,1,1,-1,-1,-1,1,-1,-1,1,1,-1,1,-1,1,1), k=1 for slot 1, No cyclic shift

X11,1=(1,1,1,1,-1,-1,-1,1,-1,-1,1,1,-1,1,-1,-1), k=2 for slot 2, cyclic right shift by 1 position

X11,14=(1,-1,-1,-1,1,-1,-1,1,1,-1,1,-1,1,1,1,1), k=15 for slot 15, cyclic right shift by 14 positions.

The same procedure for forming the cyclic codes will be used for other code groups. Thus, for the 32 codes groups and 15 slots (in one frame), 512 different cyclic codes with a length of 256 chips each are constructed. These 512 cyclic codes are unique for each code group/slot locations pair. Thus, it is possible to

uniquely determine both the scrambling code group and the frame timing in the second stage of the initial cell search.

2.3 Scrambling Code Identification

After achieving code group and frame synchronization, the scrambling code is identified by correlating the symbols in the CPICH with all possible scrambling codes in the code group. The codes are generated using a scrambling code generator and the descrambling operation is carried out using a descrambler.

Scrambling Code Generator

Each cell is allocated one and only one primary scrambling code. The scrambling code sequences are constructed by combining two real sequences into a complex sequence. Each of the two real sequences are constructed as the position wise modulo 2 sum of38,400 chip segments of two binary sequences generated by means of two generator polynomials of degree 18 [14]. Let x and y be the two sequences respectively. The resulting sequences constitute segments of a set of Gold sequences. The x sequence is constructed using the primitive polynomial $1+X^7+X^{18}$. The y sequence is constructed using the polynomial $1+X^5+X^7+X^{10}+X^{18}$. The sequence depending on the chosen scrambling code number n is denoted as zn. Furthermore, let x(i), y(i) and zn(i) denote the ith symbol of the sequence x, y, and zn, respectively.

The sequences x and y are constructed as:

x(i+18)=x(i+7)+x(i) modulo 2, for i=0,1,..,218 - 20

y(i+18)=y(i+10)+y(i+7)+y(i+5)+y(i) modulo 2, for i=0,1,..,218 - 20

The nth Gold code sequence zn, n=0,1,..,218 - 2, is then defined as [11],[12]:

zn(i)=x((i+n) modulo (218 -1))+y(i) modulo 2, i=0,1,..,218- 2

Finally, the nth complex scrambling code sequence sn is defined as:

sn(i)=zn(i)+jzn((i+131,072) modulo (218-1)), i=0,1,..,38,399

Table 2. Masking functions used in Stage 3: Scrambling code generator

	Masking Function For I Channel Code In LFSR 1	Masking Function For Q Channel Code In LFSR 1
Code1	000000000000000001	001000000001010000
Code2	000000000000000010	010000000010100000
Code3	000000000000000100	100000000101000000
Code4	000000000000001000	000000001000000001
Code5	000000000000010000	000000010000000010
Code6	000000000000100000	000000100000000100
Code7	000000000001000000	000001000000001000
Code8	000000000010000000	000010000000010000
Code9	000000000100000000	000100000000100000
Code10	000000001000000000	001000000001000000
Code11	000000010000000000	010000000010000000
Code12	000000100000000000	100000000100000000
Code13	000001000000000000	000000001010000001
Code14	000010000000000000	000000010100000010
Code15	000100000000000000	000000101000000100
Code16	001000000000000000	000001010000001000

The pattern from phase 0 up to the phase of 38,399 is repeated for every radio frame.

The scrambling codes are divided into 512 sets each of a primary scrambling code and 15 secondary scrambling codes. The primary scrambling codes consist of scrambling codes n=16*i where i=0,1,..,511. The ith set of secondary scrambling codes consists of scrambling codes 16*i+k, where k=1,2,..,15. There is a one-to-one mapping between each primary scrambling code and 15 secondary scrambling codes in a set such that ith primary scrambling code corresponds to ith set of secondary scrambling codes [17]. The set of primary scrambling codes is further divided into 32 scrambling code groups, each consisting of 16 primary scrambling codes. The jth scrambling code group consists of primary scrambling codes 16*16*j+16*k, where j=0,1,..,31 and k=0,1,..,14. In this stage, 16 scrambling codes need to be generated in parallel.

In order to reduce the hardware utilization, in stage 3 of both the designs only one scrambling code generator is used to generate 16 codes in parallel when 32 code groups are used as shown in above figure. Sixteen masking functions are used to generate the codes in parallel. Masking functions can generate codes which have minimum overlap and reduce the hardware circuitry to a single scrambling code generator at the expense of a few logic gates.

Masking function for I and Q Channel Code in linear feedback shift register (LFSR) 2 were kept fixed as 000000000000000001 and 001111111101100000. Besides reducing the hardware from 16 code generators to one code generator, the design also reduces the ROM size to 32X18 from the size 512X18 if 16 code generators were used.

3. SIMULATION AND MODELING

3.1 Slot Synchronization

Figure 1, shows sample transmitter model, fig. 2, shows slot detector model

Figure 1. Transmitter model

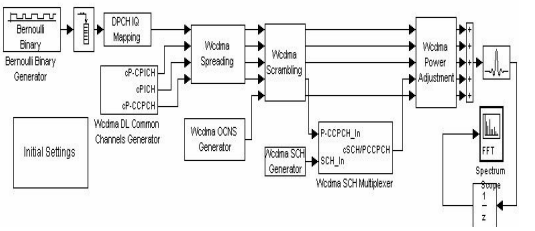


Figure 2. Slot detector model

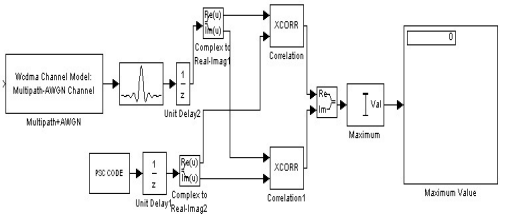
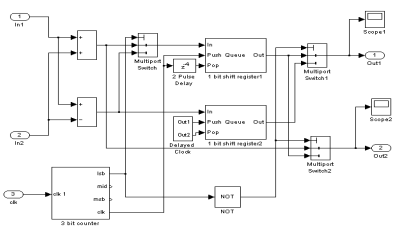


Figure 3. Individual stage of FHT



3.2 Frame Synchronization

Figure 3, shows an individual stage of the FHT. Each stage has an upper and a lower input terminal. The upper input terminal is configured to receive multiple input signals which are either Walsh chips (if the stage is the first stage of the FHT) or intermediate correlation coefficients (if the stage is not the first stage of the FHT)[15]. If an input of N-Walsh chips is to be processed then the upper input terminal receives N/2 input signal bits and the lower input terminal receives the other N/2 input bits.

Figure 4, shows the design for a FHT structure which is used for decoding a 16 chip sequence. The design proposed is a very compact and efficient implementation is used for decoding a 16 chip sequence. The design proposed is a very compact and efficient implementation as compared to previous designs

4. SIMULATION RESULTS

In the simulation results, when the received signal was correlated with the PSC sequences generated at the MS, some peak values were obtained and the maximum of those peak values was displayed as the slot value for that particular frame. In frame synchronization process, a FHT was used to match arbitrary SSC sequence with the frames and 16 values were obtained. In code synchronization process, the 16 values obtained are match with the values generated at the MS and the max of those values will be taken. The results satisfy most of the requirements of the parameters mentioned in the 3GPP specifications.

Figure 4. Design for a 16 chip FHT

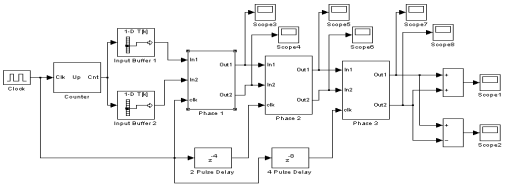


Figure 5. Code group detector model

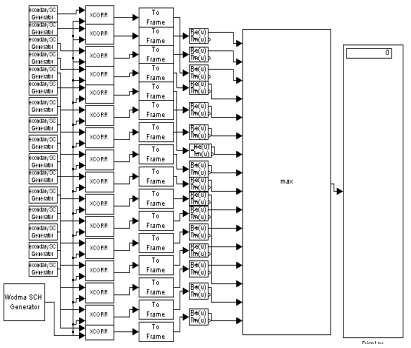


Figure 6. Scrambling code identifier model

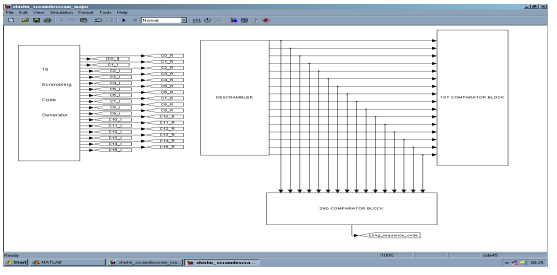


Figure 7. Spectrum for transmitted signal

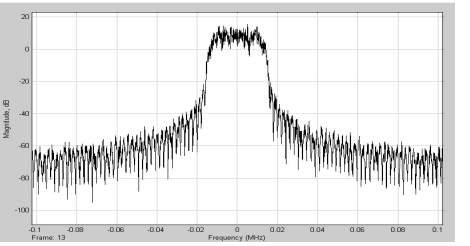


Figure 8. Maximum value in a frame

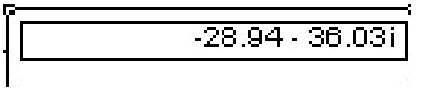


Figure 7, shows the spectrum for transmitted signal. The incoming signal is passed through a real-imaginary block to break signal into real and imaginary components. Similarly, PSC code signal is also broken into real and imaginary components. Those real parts are correlated together and the imaginary parts are correlated together. The final outcome of both the correlations is combined to form a complex value. Figure 8, shows the peak value obtained after correlating input signal with the codes generated by the PSC code generator is -28.94-36.03i. Figure 9,10,11,12 shows the first, second, third and final FHT scope. Figure 13, shows the Code Group values for real part of signal.

Frame Synchronization

In frame synchronization process, a FHT was used to match arbitrary SSC sequence with the frames and 16 values were obtained. Figure 5&6 shows stages

Figure 9. First FHT scope

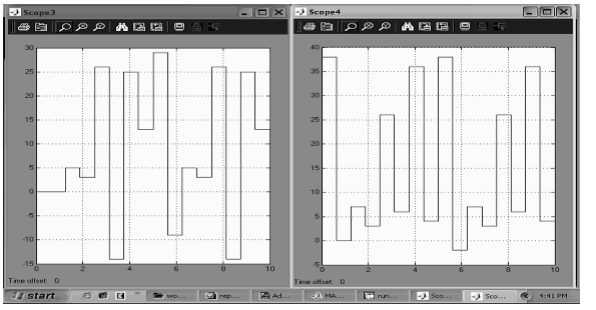


Figure 10. Second FHT scope

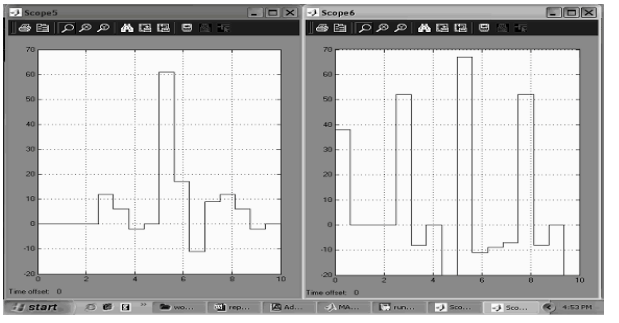


Figure 11. Third FHT scope

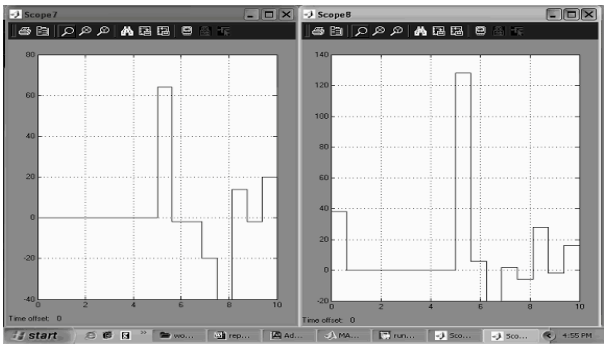


Figure 12. Final FHT scope

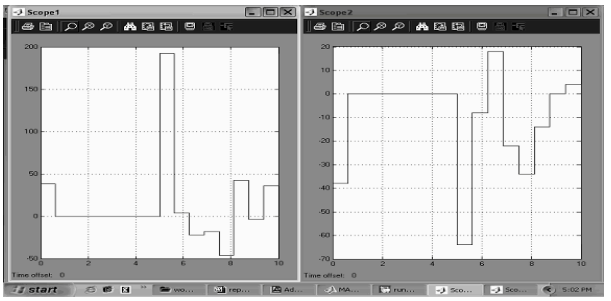
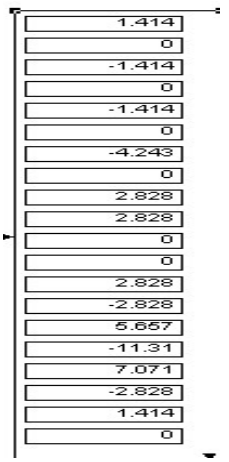


Figure 13. Code group values for real part of signal



of the FHT. Each stage has an upper and a lower input terminal. The upper input terminal is configured to receive multiple input signals which are either Walsh chips (if the stage is the first stage of the FHT) or intermediate correlation coefficients (if the stage is not the first stage of the FHT). If an input of N-Walsh chips is to be processed then the upper input terminal receives N/2 input signal bits and the lower input terminal receives the other N/2 input bits. Figure 4 shows the design for a FHT structure which is used for decoding a 16 chip sequence.

Also, we study the system performance in terms of overall acquisition time Ta. 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

Figure 14. Effect of oversampling (Nt=15, case I, non-ideal sampling, $P_{FA}=10^{-4}$)

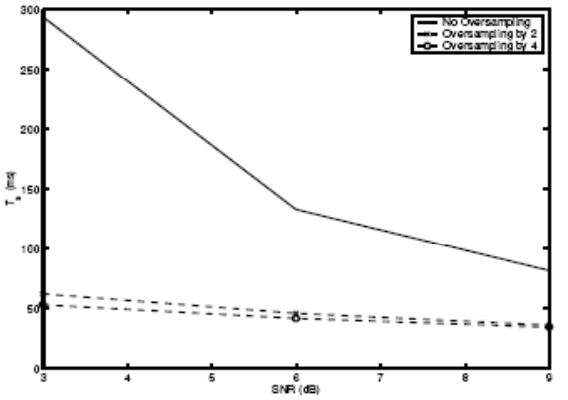
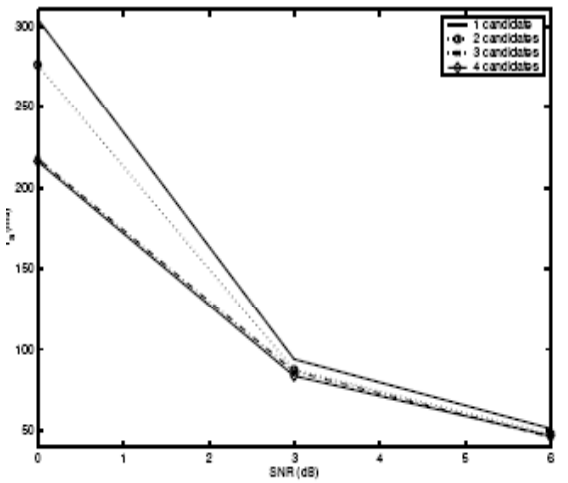


Figure 15. Influence of the number of candidates (Os=4, Nt=15, case II, ideal sampling, $P_{FA}=10^{-4}$)



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. 14. Chip rate sampling leads to unacceptable performance at lower SNR. Clearly, oversampling in the presence of non-ideal sampling reduces Ta dramatically. Furthermore, simulation results show a saturation of performance with increasing oversampling factors. Therefore, we assume an oversampling factor of Os=4 .

The dependence of the acquisition time Ta on the number of slot boundary candidates passed between stage 1 and stage 2 is illustrated in Fig. 15.

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.

CONCLUSIONS AND FUTURE WORK

This paper was able to study the various steps involved in the Cell Search process and an attempt was made to simulate them. Also we investigate the code and time synchronization of the cell search algorithm. In addition to code and time

synchronization, frequency synchronization between the MS and the BS needs to be achieved. When the received signal was correlated with the PSC sequences generated at the MS, some peak values were obtained and the maximum of those peak values was displayed as the slot value for that particular frame. In frame synchronization process, a FHT was used to match arbitrary SSC sequence with the frames and 16 values were obtained.In code synchronization process, the 16 values obtained are match with the values generated at the MS and the max of those values will be taken. 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. There is another cell search called target cell search, which needs to be performed during a call, and when a MS is in motion and moves from one cell to another. VLSI implementations to perform target cell search efficiently need to be investigated.

REFERENCES

[1] T . Ojanpera and R.Prasad, " An overview of air interface multiple access for IMT- 2000/UMTS", IEEE Commun. Mag.,vol. 36, pp. 82-95, Sept. 1998.

[2] E. Dahlman, P.Berning, J.Knutsson, F.Ovesjo, M.Persson, and C.Roobol, "WCDMA the Radio Interface for Future Mobile Multimedia communications", IEEE Trans. Veh. Tech.., vol. 47,pp. 1105-1118, Nov. 1998.

[3] Harri Holma, Antti Toskala, "WCDMA for UMTS: Radio Access for Third Generation Mobile Communications", John Wiley, 2000

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

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

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

[7] Chi-Fang Li ,Wern- Ho Sheen, Ho,J.J.-S.,Yuan-Sun Chu, "ASIC Design for Cell Search in 3GPP W-CDMA",Fall. IEEE VTC 2001, vol.3,pp. 1383-1387.

[8] "3G TS 25.214 – Technical specification group radio access networks: physical layer procedures (FDD)", www.3GPP.org, Release 1999.

[9] S. Sriram, S. Hosur, "An analysis of the 3-stage search process in WCDMA", Proc. IEEE VTC 2000, Boston.

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

[11] K.Higuchi, M.Sawahashi and F.Adachi, " Fast cell search algorithm in DS-CDMA mobile using long spreading codes", in Proc. IEEE 1997 Veh. Technol.Conf.,Phoenix, May 1997, pp. 1430-1434

[12] Kamal Bahl S. "Designing Hardware Efficient Acquisition Units for Initial Cell Search in WCDMA", in Proceedings 3GWireless 2003, San Francisco, CA, May 2003

[13] Nystorn,K.Jamal, Y.-P.E. Wang, and R.Esmailzadeh, " Comparison of cell search methods for asynchronous wideband CDMA cellular system", in Proc. IEEE, Int. Conf.Universal Personal Communication., Florence, Italy, Oct 1998.

[14] Ericsson, "New downlink scrambling code grouping scheme for UTRA/FDD", TSG-RAN Working Group 1 Meeting 6, TSGR1#6(99)884.

[15] A.Amira, A.Bouridane, P.Milligan and M.Roula, "Novel FPGA Implementations of Walsh-Hadamard transforms for signal processing", in Proc. IEE Vision, Image and Signal Processing, Dec 2001, vol. 148,no.6, pp. 377-383.

[16] M. Kiessling and S.A. Mujtaba, "Performance Enhancements to the UMTS (W- CDMA) Initial Cell Search Algorithm", IEEE International Conference on Communications, vol. 1, May 2002, pp. 590-594.

[17] Shailendra Mishra,Nipur "Code & Time synchronization of the Cell Search Design" in HB of Research in Mobile business: Technical, Methodological & Social perspectives , IDEA Group Publishing (IGP) Hershley PA USA pp 173-183,April 2006