TSG-RAN Working Group 1 meeting #7 **TSGR1 # 7(99)d42**

Hannover, Germany

August 30 - September 3, 1999

Agenda Item:

Source: **Nortel Networks**

Title: **Synchronization Channel with cyclic hierarchical sequences in UTRA TDD**

Document for: Ad Hoc 1

Abstract

This contribution proposes replacing the Comma Free Codes in the Secondary SCH of the PSCH with Cyclic Hierarchical Sequences (CHSs) for the UTRA TDD mode. In place of other alternatives currently being debated, the proposed change will improve the overall synchronization in the TDD mode by decreasing search time while maintaining primary synchronization channel performance and peak to average ratio of power. Further, the proposed codes present an unlimited number of CHSs with good correlation properties that can be used for future extension of the standard.

1. Summary

The current UTRA TDD scheme uses a (16, 3) CFC Comma Free Codes (CFC) in the Secondary SCH of the PSCH, which was designed for the UTRA FDD. This (16,3) code is decimated by two and resulting length 8 code is stretched out in time over 4 (or 8) frames. It was shown in [4] that the performance of the second step of the acquisition is poor. Several new proposals were suggested concerning the secondary sequences to improve performance of their acquisition [4-7]. These proposals have several disadvantages in that they require modifications to the primary sync channel, degrade the primary sync channel performance and increase the peak to average ratio of power. In addition, the need to decode over a multiplicity of frames introduces a longer decoding delay.

This contribution proposes replacing only the Comma Free Codes in the Secondary SCH of the PSCH with Cyclic Hierarchical Sequences (CHSs) [8] for the UTRA TDD mode. In place of other alternatives currently being debated, the proposed change will improve the overall synchronization in the TDD mode by decreasing search time while maintaining primary synchronization channel performance and peak to average ratio of power. Further, the proposed codes present an unlimited number of CHSs with good correlation properties that can be used for future extension of the standard.

The PSCH with CHSs uses a modulation with an enlarged alphabet to uniquely identify code group and position of Cs in the second step of the acquisition. Because of the larger alphabet only one secondary sequence Cs is enough to uniquely identify code group and frame timing thus enabling faster acquisition.

2. Proposal

It is recommended that cyclic hierarchical sequences be adopted for the secondary synchronization sequences (codes) in UTRA TDD mode. This will require changes to section *7.1 Code Generation*, section *7.2.1 Code allocation for case 1 and 2* and section *7.2.2 Code allocation for case 3* in *TS.25.223 Spreading and modulation (TDD).*

3. Background

In the TDD mode, there are three cases of PSCH and CCPCH allocation as follows [1]:

Case 1) PSCH and CCPCH allocated in TS#k, k=0...14

Case 2) PSCH in two TS and CCPCH in the same two TS: TS#k and TS#k+8, k=0...6

Case3) PSCH in two TS, TS#k and TS#k+8, k=0...6, and the primary CCPCH TS#i, i=0...14, pointed by PSCH. Pointing is determined via the SCH from the higher layers.

Figure 1 is an example for the transmission of PSCH, k=0, in Case 2 or Case 3. As depicted in Figure 1, the PSCH consists of a primary and secondary code sequence of length 256 chips. The sequences Cp and Cs are the same as in FDD mode.

The time offset t_{gap} is the time between the primary synchronization code and the secondary synchronization code. It provides enough time for calculations and a better interference distribution, since the codes do not superimpose.

Due to mobile to mobile interference, it is mandatory for public TDD systems to keep synchronization between base stations. Consequently, a capture effect can occur for the PSCH. The time offset t_{offset} enables the system to overcome the capture effect. The time offset t_{offset} is one of 32 values, depending on the cell parameter and thus on the code group of the cell [2].

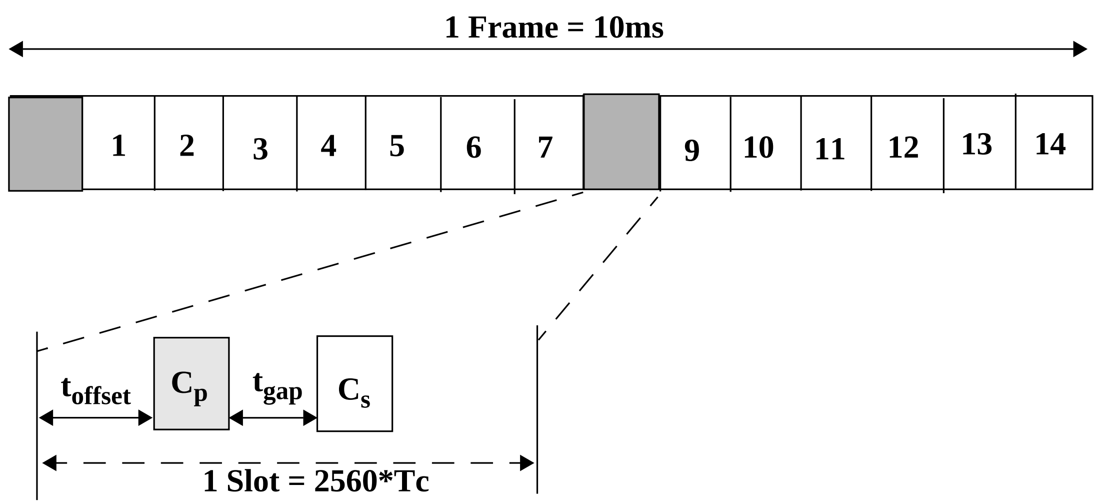


Figure 1: Structure for physical Synchronization Channel PSCH consisting of one primary sequence C_p and one secondary sequence C_s in slot k and k+8 (example for k=0 in Case 2 or Case 3)

For successfully identifying the CCPCH position, the following information is needed from PSCH:

- whether it is the first or second slot in the frame carrying the SCH;

- whether it the first or the second frame of an interleaving period of 2 frames of CCPCH (20 ms synchronization).

For Case 3 of the synchronization alternatives, three additional bits are also needed.

4. Rationale

The most critical part of synchronization is the first step when primary code Cp is acquired. This is because the uncertainty region for search is the whole frame (38400 chips) or half frame (19200 chips) depending on the alternative cases (case 1 or case 2 (case 3)) and probability of "false alarm" is significant. During the first step, the UE selects 1 maximum peak (slot acquisition with some probability of detection) and "goes" to the second step.

During the second step, the UE attempts to acquire the code group and frame timing. We should expect that performance of the correct synchronization in the second step is much better than in the first step because the number of decision variables is significantly less assuming the same signal to noise ratio per chip.

Results provided in [4] (and some others distributed on 3GPP e-mail reflector) confirm this. Let us consider simulation results of the acquisition in the first and in the second steps in fading environment (100 km/h, 6 Rayleigh multipaths, figures 2 and 3) that are presented in [4]. To maintain 30% of incorrect synchronization, approximately -6 dB in the first step is required for the first step and approximately -14 dB in the second step. It is clear that the first step performance is crucial to the acquisition and therefore the performance of the first step should not be compromised to increase performance of second step. In other words, if the slot is acquired correctly during the first step, it almost guarantees a correct decision in the second step.

The proposals [4-7] also suggest transmitting Cs on top of Cp instead of sending Cs after Cp. This might also reduce the primary search performance in the first step. The peak to average ratio of power is worse for these proposals compared to the current 3GPP scheme. In addition, it might negatively impact the accuracy of the channel estimation using midamble depending on the t_{ofsset} because instant power of PSCH is doubled.

This contribution retains the structure of the PSCH and replaces only secondary synchronization sequences Cs using an enlarged alphabet to encode code group and slot position of the PSCH. In other words, each distinctive combination of code group and slot position is assigned its own 256 chips signal.

5. Cyclic Hierarchical Sequences (CHSs) for the Secondary SCH

5.1. Description

Cyclic Hierarchical Sequences (CHSs) are proposed because the procedure to generate the codes is simple and they have a structure that allows very efficient detection (Please see details for generating CHSs in Appendix A). Note that it is also possible to generate practically unlimited number of cyclic hierarchical sequences while using different scrambling codes for the whole set (or subset) of cyclic hierarchical sequences.

The 512 cyclic hierarchical sequences described in Appendix A are unique and it is possible to determine both the correct code group and frame timing by searching only 1 slot with secondary codes after acquiring the slot position. This significantly reduces search time in comparison with the current 3GPP scheme and others that have been proposed.

- For Case 1, 64 (=32*2) different signals are necessary for 32 groups and 2 frames interleaving, resulting in 64 decision variables.
- For Case 2, 64 (=32*2) different signals are necessary for 32 groups and 2 frames interleaving, resulting in 128 decision variables for 2 slots with PSCH if the same signals are used in both slots.

- For Case 3, 512 (=32*2*8) different signals are necessary for 32 groups, 2 frames interleaving and 3 bits (8=23) needed for the location of the primary CCCH. 1024 decision variables result for 2 slots with PSCH if the same signals are used in both slots

For 15 slots per frame, location of the slot with PSCH within the frame is indicated by the relative offsets of the SCH slots (case 2 and case 3). This means that we can use the same signals in the first slot (k) and second slot (k+8).

Case 1

Only 64 cyclic hierarchical sequences (32 pairs) that were generated from the first 4 codes are used. The cyclic hierarchical sequences for 32 code groups are shown in Table 1. In this case, the code group pairs shown in Table 1 are transmitted in Frames 1 & 2 respectively. See also Appendix C.

CHSs for code groups from 1 to 8	CHSs for code groups from 17 to 24
$C^{1,1}$ $C^{1,9}$	$C^{3,1}$ $C^{3,9}$
$C^{1,2}$ $C^{1,10}$	$C^{3,2}$ $C^{3,10}$
$C^{1,3}$ $C^{1,11}$	$C^{3,3}$ $C^{3,11}$
$C^{1,4}$ $C^{1,12}$	$C^{3,4}$ $C^{3,12}$
$C^{1,5}$ $C^{1,13}$	$C^{3,5}$ $C^{3,13}$
$C^{1,6}$ $C^{1,14}$	$C^{3,6}$ $C^{3,14}$
$C^{1,7}$ $C^{1,15}$	$C^{3,7}$ $C^{3,15}$
$C^{1,8}$ $C^{1,16}$	$C^{3,8}$ $C^{3,16}$
CHSs for code groups from 9 to 16	CHSs for code group from 25 to 32
$C^{2,2}$ $C^{2,10}$	$C^{4,1}$ $C^{4,9}$
$C^{2,2}$ $C^{2,10}$	$C^{4,2}$ $C^{4,10}$
$C^{2,3}$ $C^{2,11}$	$C^{4,3}$ $C^{4,11}$
$C^{2,4}$ $C^{2,12}$	$C^{4,4}$ $C^{4,12}$
$C^{2,5}$ $C^{2,13}$	$C^{4,5}$ $C^{4,13}$
$C^{2,6}$ $C^{2,14}$	$C^{4,6}$ $C^{4,14}$
$C^{2,7}$ $C^{2,15}$	$C^{4,7}$ $C^{4,15}$

$C^{2,8}$ $C^{2,16}$	$C^{4,8}$ $C^{4,16}$
Table 1: CHS, case 1 and case 2	

Case 2

In this case, the code group pairs shown in Table 1 are transmitted in Frames 1&2 respectively. The same CHSs are used for both first (k) and second (k+8) slots. See also Appendix C.

Case 3

The whole set of all 512 cyclic hierarchical sequences is used (Table 2). However, in difference from cases 1 and 2, each code group uses CHSs that generated from only one 16 bit code. We are proposing to use 8 pairs to provide information about location of the Primary CCCH (to represent 3 Additional Bits from SCH Transport Channel).

For this case (similar to the case 2), the same CHSs are used for both first (k) and second (k+8) slots in Frame 1 (Frame 2). See also Appendix C.

CHSs for code groups from 1 to 8	CHSs for code groups from 17 to 24
$C^{1,1}$ $C^{1,9}$ $(C^{1,2}$ $C^{1,10}$, ... , $C^{1,8}$ $C^{1,16})$	$C^{17,1}$ $C^{17,9}$ $(C^{17,2}$ $C^{17,10}$, ... , $C^{17,8}$ $C^{17,16})$
$C^{2,1}$ $C^{2,9}$ $(C^{2,2}$ $C^{2,10}$, ... , $C^{2,8}$ $C^{2,16})$	$C^{18,1}$ $C^{18,9}$ $(C^{18,2}$ $C^{18,10}$, ... , $C^{18,8}$ $C^{18,16})$
$C^{3,1}$ $C^{3,9}$ $(C^{3,2}$ $C^{3,10}$, ... , $C^{3,8}$ $C^{3,16})$	$C^{19,1}$ $C^{19,9}$ $(C^{19,2}$ $C^{19,10}$, ... , $C^{19,8}$ $C^{19,16})$
$C^{4,1}$ $C^{4,9}$ $(C^{4,2}$ $C^{4,10}$, ... , $C^{4,8}$ $C^{4,16})$	$C^{20,1}$ $C^{20,9}$ $(C^{20,2}$ $C^{20,10}$, ... , $C^{20,8}$ $C^{20,16})$
$C^{5,1}$ $C^{5,9}$ $(C^{5,2}$ $C^{5,10}$, ... , $C^{5,8}$ $C^{5,16})$	$C^{21,1}$ $C^{21,9}$ $(C^{21}$ $C^{21,10}$, ... , $C^{21,8}$ $C^{21,16})$
$C^{6,1}$ $C^{6,9}$ $(C^{6,2}$ $C^{6,10}$, ... , $C^{6,8}$ $C^{6,16})$	$C^{22,1}$ $C^{22,9}$ $(C^{22,2}$ $C^{22,10}$, ... , $C^{22,8}$ $C^{22,16})$
$C^{7,1}$ $C^{7,9}$ $(C^{7,2}$ $C^{7,10}$, ... , $C^{7,8}$ $C^{7,16})$	$C^{23,1}$ $C^{23,9}$ $(C^{23,2}$ $C^{23,10}$, ... , $C^{23,8}$ $C^{23,16})$
$C^{8,1}$ $C^{8,9}$ $(C^{8,2}$ $C^{8,10}$, ... , $C^{8,8}$ $C^{8,16})$	$C^{24,1}$ $C^{24,9}$ $(C^{24,2}$ $C^{24,10}$, ... , $C^{24,8}$ $C^{24,16})$
CHSs for code groups from 9 to 16	**CHSs for code group from 25 to 32**
$C^{9,1}$ $C^{9,9}$ $(C^{9,2}$ $C^{9,10}$, ... , $C^{9,8}$ $C^{9,16})$	$C^{25,1}$ $C^{25,9}$ $(C^{25,2}$ $C^{25,10}$, ... , $C^{25,8}$ $C^{25,16})$
$C^{10,1}$ $C^{10,9}$ $(C^{10,2}$ $C^{10,10}$, ... , $C^{10,8}$ $C^{10,16})$	$C^{26,1}$ $C^{26,9}$ $(C^{26,2}$ $C^{26,10}$, ... , $C^{26,8}$ $C^{26,16})$
$C^{11,1}$ $C^{11,9}$ $(C^{11,2}$ $C^{11,10}$, ... , $C^{11,8}$ $C^{11,16})$	$C^{27,1}$ $C^{27,9}$ $(C^{27,2}$ $C^{27,10}$, ... , $C^{27,8}$ $C^{27,16})$
$C^{12,1}$ $C^{12,9}$ $(C^{12,2}$ $C^{12,10}$, ... , $C^{12,8}$ $C^{12,16})$	$C^{28,1}$ $C^{28,9}$ $(C^{28,2}$ $C^{28,10}$, ... , $C^{28,8}$ $C^{28,16})$
$C^{13,1}$ $C^{13,9}$ $(C^{13,2}$ $C^{13,10}$, ... , $C^{13,8}$ $C^{13,16})$	$C^{29,1}$ $C^{29,9}$ $(C^{29,2}$ $C^{29,10}$, ... , $C^{29,8}$ $C^{29,16})$

$C^{14,1}\ C^{14,9}\ (C^{14,2}\ \ C^{14,10}\ , \ldots ,\ C^{14,8}\ C^{14,16})$	$C^{30,1}\ C^{30,9}\ (C^{30,2}\ \ C^{30,10}\ , \ldots ,\ C^{30,8}\ C^{30,16})$
$C^{15,1}\ C^{15,9}\ (C^{15,2}\ \ C^{15,10}\ , \ldots ,\ C^{15,8}\ C^{15,16})$	$C^{31,1}\ C^{31,9}\ (C^{31,2}\ \ C^{31,10}\ , \ldots ,\ C^{31,8}\ C^{31,16})$
$C^{16,1}\ C^{16,9}\ (C^{16,2}\ \ C^{16,10}\ , \ldots ,\ C^{16,8}\ C^{16,16})$	$C^{32,1}\ C^{32,9}\ (C^{32,2}\ \ C^{32,10}\ , \ldots ,\ C^{32,8}\ C^{32,16})$

Table 2: CHS, case 3

5.2. Correlation Properties

Analysis of the auto- and cross-correlation properties of the primary sequence Cp and the proposed secondary sequences Cs has revealed that their correlation proprieties are very good [8]. Cross correlations between primary sequence and secondary sequences are small and approximately the same as in the current proposal. Please see Appendix D for some typical examples. Typical maximum aperiodic cross-correlations between different secondary cyclic hierarchical sequences are also small and approximately equal to the maximum sidelobe of the auto-correlation function of the primary Cp. Only a small amount (3-5%) of these cross-correlations reach 25 percent of the main peak. The correlation properties are quite acceptable for synchronization purposes and we can expect good results for the acquisition of the primary Cp and the secondary Cs with cyclic hierarchical sequences.

It is possible to improve cross correlation properties of the secondary sequences while scrambling each secondary sequence Cs with some (random or pseudo-random) sequence (e.g., m-sequence, hadamard sequence). It is clear that this scrambling does not increase complexity of decoding because each secondary sequence Cs of the whole set of sequences will be scrambled with the same sequence.

It is also possible also generate practically unlimited number of cyclic hierarchical sequences while using different scrambling codes for the whole set (or subset) of cyclic hierarchical sequences.

5.3. Cell Search

The cell search procedure is similar to the current 3GPP procedure and is carried out in three steps [3, section 4.5] as described in Appendix B.

Only the relevant aspects of Step 2 of the cell search for the case where PSCH and CCPCH are in time slots k and k+8 with k = 0...6, are given here.

During the second step of the cell search procedure, the UE uses the cyclic hierarchical sequence of Secondary Synchronization Codes to find frame synchronization and identify one of 32 code groups. Each code group is linked to a specific t_{offset}, and thus to a specific frame timing.

In cases 2 and 3 it is required to detect the position of the next synchronization slots. To do so, the secondary synchronization code is correlated with the received signal at offsets of 7 and 8 time slots from the position of the primary code that was detected in the first step.

Note that only one secondary synchronization code (cyclic hierarchical sequence) needs to be decoded to find synchronization of frames and code group identification. However, integration of several slots can also be used if necessary to further improve performance of detection in the second step.

5.4. Complexity evaluation

Results of complexity evaluation are summarized in Table 3 (please see Appendix B for details). The complexity is normalized to Step-1 where matched filter (or similar device) is used.

It is clear that Step-2 has a very small complexity in comparison to the first step of cell search. Therefore, any increase in complexity of the second step will have negligible effect on power consumption of UE. The complexity of implementation of the Second-ary SCH might be even less depending on its implementation.

Case	Step-1 Complexity	Step-2 Complexity Initial acquisition	Step-2 Complexity Handoff
Case 1	100%	0.2%	less than 0.2%
Case 2	100%	0.8%	less than 0.8%
Case 3	100%	6.5%	2%

Table 3: Complexity summary in percent relative to Step-1

5.5. Performance

- **Performance of Step-1** is maintained as in the current 3GPP proposal.
-
- **Performance of Step-2** is significantly improved because of the enlarged alphabet of secondary sequences Cs with a minor increase in complexity. The search during second step will be much faster than in the current and others proposals. Typically, integration of several slots is not required during the second step and only one slot is enough to correctly decode code group and frame timing.

- **PAR problem**: peak to average ratio of power as good as in the current 3GPP proposal
- **Flexibility**: *unlimited* possibility to increase the number of CHSs with good correlation properties for future extensions.

The benefits stated above compensate for the slight increase in complexity of this method.

6. Conclusion

This contribution proposed replacing the Comma Free Codes in the Secondary SCH of the PSCH with Cyclic Hierarchical Sequences (CHSs) for the UTRA TDD mode. The procedure to generate the codes is simple and they have a structure that allows very efficient detection. For future extension of the standard, it is also possible to generate practically unlimited number of cyclic hierarchical sequences with good correlation properties while using different scrambling codes for the whole set (or subset) of cyclic hierarchical sequences.

In place of other alternatives currently being debated, the proposed change will affect only the secondary codes. It will improve the overall synchronization in the TDD mode by decreasing search time while maintaining primary synchronization channel performance and peak to average ratio of power.

Typically, non-coherent detection of one burst (slot) is required. However, coherent detection and integration of several slots can also be used if necessary to further improve performance of detection in the second step.

7. References

[1] TS25.221 V1.2.1 Physical channels and mapping of transport channels onto physical chan nels (TDD)

[2] TS 25.223 V2.1.1 Spreading and modulation (TDD)

[3] TS 25.224 V1.0.1 Physical layer procedures (TDD)

[4] A BPSK Modulated Secondary Synchronization Code based Cell Search in UTRA TDD, TSGR1#5(99)578, InterDigital Communications Corporation

[5] TDD Cell Search and Text Proposals for 25.221, 25.223 and 25.224, TSGR1#6(99)975, InterDigital Communications Corporation

[6] TDD Synchronization scheme based on modulated secondary codes-additional results, TSGR1#6(99)976, InterDigital Communications Corporation

[7] A New Comma Free Code Scheme for TDD Synchronization, TSGR1#6(99)815, Texas Instruments

[8] Synchronization Channel with Cyclic Hierarchical Sequences, TSGR1#2(99)090, Nortel Networks

Appendix A. The procedure for constructing the cyclic hierarchical sequences(CHSs)

The procedure for constructing the *cyclic hierarchical sequences* (CHSs) is similar to that of the *hierarchical sequence* but using specific constituent length 16 codes (see table below).

Note: For hierarchical sequence, we will generate only 1 sequence of length 256 chips but for cyclic hierarchical sequence we will generate 16 different sequences length, each 256 chips (from only 16 bits).

For each 16 bit code, 16 CHSs can be generated, each CHS with length 256 chips. Below we will show procedure of generating these 16 CHS for the code number i.

For CHS 1, the cyclic hierarchical sequence is constructed from two constituent codes $X_{1,i}$ and $X_{2,i}$ of length n_1 and n_2 respectively using the following formula:

$$C^{i,1}(n)= X_{2,i}\,(n \bmod n_2)+X_{1,i}\,(n \operatorname{div} n_1) \text{ modulo } 2,\ n=0...\ (n_1*n_2)-1 \qquad (1)$$

where i is the code number

The constituent sequences $X_{1,i}$ and $X_{2,i}$ in each code group are chosen to be identical and to be the following length 16 sequence (X_2 is inner constituent code, X_1 is outer constituent code):

Length 16 constituent codes $X_{1,i}$ and $X_{2,i}$ for	
codes from 1 to 8	code from 17 to 24
0 0 0 1 1 1 0 1 1 0 0 1 0 1 0 0	0 1 0 0 1 0 1 0 0 0 1 0 0 0 1 0
0 1 0 0 1 0 0 0 1 1 0 0 0 0 0 1	0 0 0 1 1 1 1 1 0 1 1 1 0 1 1 1
0 0 1 0 1 1 1 0 1 0 1 0 0 1 1 1	0 1 1 1 0 1 1 0 0 0 0 1 1 1 1 0
0 1 1 1 1 0 1 1 1 1 1 1 0 0 1 0	0 0 1 0 0 0 1 1 0 1 0 0 1 0 1 1
0 0 0 1 0 0 1 0 1 0 0 1 1 0 1 1	1 1 1 0 0 1 1 0 0 1 0 1 0 0 1 1
0 1 0 0 0 1 1 1 1 1 0 0 1 1 1 0	1 0 1 1 0 0 1 1 0 0 0 0 0 1 1 0
0 0 1 0 0 0 0 1 1 0 1 0 1 0 0 0	1 1 0 1 0 1 0 1 1 0 0 1 1 1 1 1
0 1 1 1 0 1 0 0 1 1 1 1 1 1 0 1	1 0 0 0 0 0 0 0 1 1 0 0 1 0 1 0
codes from 9 to 16	codes from 25 to 32
0 0 1 0 1 1 1 0 0 1 0 1 1 0 0 0	1 0 0 0 1 1 0 0 0 1 1 1 1 1 1 0
0 1 1 1 1 0 1 1 0 0 0 0 1 1 0 1	1 1 0 1 1 0 0 1 0 0 1 0 1 0 1 1
1 0 1 1 1 1 1 0 0 0 0 1 0 1 0 1	1 0 0 0 0 0 1 1 0 1 1 1 0 0 0 1
1 1 1 0 1 0 1 1 0 1 0 0 0 0 0 0	1 1 0 1 0 1 1 0 0 0 1 0 0 1 0 0
0 1 1 1 0 1 1 0 1 1 1 0 0 0 0 1	1 0 1 1 0 0 0 0 0 1 0 0 0 0 1 0

0 0 1 0 0 0 1 1 1 0 1 1 0 1 0 0	1 1 1 0 0 1 0 1 0 0 0 1 0 1 1 1
0 1 1 1 1 0 0 1 0 0 0 1 0 0 0 1	1 0 0 0 1 1 0 0 1 0 0 0 0 0 0 1
0 0 1 0 1 1 0 0 0 1 0 0 0 1 0 0	1 1 0 1 1 0 0 1 1 1 0 1 0 1 0 0
Constituent codes for cyclic hierarchical sequences	

We can see that the procedure for constructing the *cyclic hierarchical sequence* $C^{i,1}$ for CHS 1 is exactly the same as constructing the *hierarchical sequence* Cp for the hierarchical Primary SCH. The sequence $C^{i,1}$ for CHS 1 will be referred to as the *zero cyclic shift sequence* as no shift is applied to the constituent sequence $X_{1,i}$.

For CHSs 2 to 16, the cyclic hierarchical sequences are constructed from the two constituent sequences $X_{1,i,k-1}$ and $X_{2,i,k-1}$ of length n_1 and n_2 respectively using the following formula:

$$C^{i,k}(n)= X_{2,i,k-1}\ (n \bmod n_2)+X_{1,i,k-1}\ (n \operatorname{div} n_1) \text{ modulo } 2,\ n=0...\ (n_1*n_2)-1 \qquad (2)$$

where i is code number, k=2,...,16 is CHS number, n is chip number in CHS

The constituent codes $X_{1,i,k-1}$ and $X_{2,i,k-1}$ are chosen to be the following length 16 sequences from the above Table 1.

- The constituent code $X_{2,i,k-1}$ (inner sequence) is *exactly equal* to the base code $X_{2,i}$ in every CHS, i.e. $X_{2,i,k-1} = X_{2,i}$ at all k.
- The constituent code $X_{1,i,k-1}$ (outer sequence) are formed from the base code $X_{1,i}$ by cyclic shifts of $X_{1,i}$ on k-1 positions (from 0 to 15) clockwise for each CHS number k, from 1 to 16.

For example, for the first code:

$X_{1,1,0} = X_{1,1}$ = < 0 0 0 1 1 1 0 1 1 0 0 1 0 1 0 0 >, k=1 for CHS 1, No cyclic shift

$X_{1,1,1}$ = < 0 0 0 0 1 1 1 0 1 1 0 0 1 0 1 0 >, k=2 for CHS 2, cyclic shift by 1 position

...

$X_{1,1,15}$ = < 0 0 1 1 1 0 1 1 0 0 1 0 1 0 0 0>, k=16 for CHS 16, cyclic shift by 15 positions

The same procedure for forming the cyclic hierarchical sequences will be used for other codes.

Thus, for the 32 codes, we will construct 512 *different* cyclic hierarchical sequences with a length of 256 chips each. In other words, we will construct 16 cyclic hierarchical sequences in each of 32 codes, altogether 512 different cyclic hierarchical sequences.

$C^{1,1}$ $C^{1,2}$ $C^{1,3}$, ... , $C^{1,15}$ $C^{1,16}$

$C^{2,1}$ $C^{2,2}$ $C^{2,3}$, ... , $C^{2,15}$ $C^{2,16}$
. . .
$C^{17,1}$ $C^{17,2}$ $C^{17,3}$, ... , $C^{17,15}$ $C^{17,16}$
. . .
$C^{31,1}$ $C^{31,2}$ $C^{31,3}$, ... , $C^{31,15}$ $C^{31,16}$
$C^{32,1}$ $C^{32,2}$ $C^{31,3}$, ... , $C^{32,15}$ $C^{32,16}$

It can be easily shown that this set of 512 cyclic hierarchical sequences have good correlation properties that make it them good candidates for the Secondary Synchronization Codes (secondary synchronization sequences). See also Appendix D.

These 512 cyclic hierarchical sequences are unique and it is possible to determine both the correct code group and frame timing by search only 1 slot with secondary sequences after acquiring slot position.

Appendix B. Implementation details

In this section, we provide desription a possible implementation of the new scheme and evaluate complexity. Since Step-1 and Step-2 are tied together, we begin by reviewing the implementation and complexity of Step-1.

B1. Step-1 of the cell search

During the first step of the initial cell acquisition, we must use a matched filter (or any similar device) for acquisition of the Primary sequence Cp. The primary sequence is constructed as a so-called generalized hierarchical golay sequence of length 256 chips that is transmitted once (case 1) or two times (case 2 and case 3) per 10 ms frame.

Let us calculate the complexity C_{step-1} of implementing the matched filter of generalized hierarchical sequence (first step).

C_{step-1}=13*2*2*38400 =1,996,800 real additions per one frame

- where 13*2 operations per matched filter output, assumed 2 samples per chip and 38400 chips in one frame.

Thus, in **step-1** of the cell search, we require:

- **1,996,800 real additions in one 10 ms frame**

In addition, we need also a (ring) Primary Buffer with 256 complex memory cells (8-10 bits in each memory cell), possibly some other circuitry, and control logic.

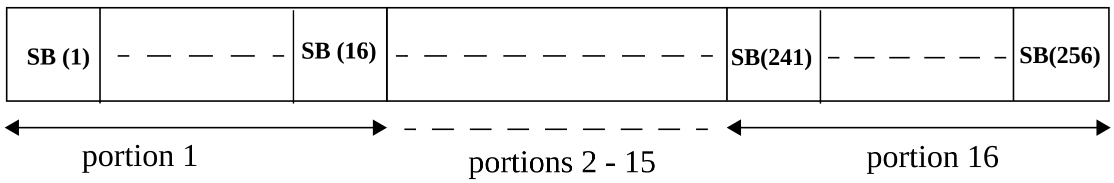
B2. Complexity to decode 16 CHSs of 1 code in the Step-2 of the cell search

During the second step of the initial cell acquisition, we must correlate with the secondary sequences Cs. The secondary sequences are cyclic hierarchical sequences of length 256 chips (please see appendix A for details of their forming). This secondary sequences appear periodically ones or two times a frame.

In the following, we give an example of one possible approach to acquiring the secondary sequence and calculate complexity to decode 16 cyclic hierarchical sequences that generated from one 16 bit code. For convenience, we have separated the acquisition process into 3 stages.

First stage. During the 1st stage of acquiring the secondary sequences, we need to store the input data samples for the SCH (combined primary and secondary secondary sequence) in some buffer with 256 complex memory cells. These input data samples are produced after *waveform* matched filtering and sampling at the chip rate (Note that sampling at the chip rate is possible because we know the beginning of the slot with accuracy better than 1/2 chip after step-1 of the cell acquisition). We will call this buffer, with 256 complex memory cells SB, the *Secondary Buffer* and use it to store input data for every slot. If 4-6 bits are used for A/D conversion at the receiver input, then each memory cell of Secondary Buffer will require 5-7 bits. For convenience, we separate the 256 complex memory cells into 16 portions with 16 complex memory cells in each portion as shown in Figure 2.

Secondary Buffer



portion 1: SB(1), SB(2),..., SB(16)
portion 2: SB(17), SB(18),..., SB(32)
....
portion 16: SB(241), SB(242),..., SB(256)

SB(1) corresponds to the 1-st chip in slot
SB(256) corresponds to the 256-th chip in slot

Figure 2: Structure of Secondary Buffer

Second stage. For the code i during the 2^{nd} stage of the acquiring of the secondary sequence, we need to produce 16 complex *common variables* (CV). These common variables CV should be calculated in a special way (by despreading data *SB* taken from the Secondary Buffer with sequence $X_{2,i}$ and integrating). Sequence $X_{2,i}$ should be taken from Table 1 according to the number i of the code group under test.

- variable CV (1) is calculated from *portion 1* and $X_{2,i}$
- variable CV (2) is calculated from *portion 2* and $X_{2,i}$
- ...
- variable CV (16) is calculated from *portion 16* and $X_{2,i}$

Thus, after despreading with $X_{2,i}$, we produce 16 complex common variables CV(1),..., CV(16). It is evident that we need 256=16*16 complex additions to form these 16 common variables CV. The number of bits necessary to represent the common variables CV with fixed point can be in the range of 9-11 bits (approximately 4 bits more than is required for each memory cell in the Secondary Buffer).

Third stage. For the code i during the 3^{rd} stage of the acquiring of the secondary sequence , we need to calculate 16 complex *correlation outputs* in each slot (using data from common variables CV). This is done by clockwise cyclically shifting the outer constituent sequence $X_{1,i}$ by k-1 positions (k=1...16) to produce $X_{1,i,k-1}$ and correlate each of these codes with the common variables CV(1)... CV(16).

Thus for each CHS of code i, we need to produce 16 complex correlation variables $CorVar_{i,1}$,..., $CorVar_{i,16}$ to determine the correct slot number. It is evident that we need 256=16*16 complex additions to correlate the codes $X_{1,i,k-1}$ with the common variables CV(1)... CV(16) to produce the 16 correlation variables CorVar. Together with the 256 complex additions needed to produce the common variables CV, we need a total of 512=256+256 complex additions to form the correlation variables CorVar for each used code i (or 1024=512*2 real additions per one code i).

After the correlation variables CorVar are formed, they are used to form *decision variables* and the complex memory cells used for the *common variables* and the *correlation variables* can be reused.

Thus to decode 16 cyclic hierarchical sequences that generated from one 16 bit code we require

- **1024 real additions to decode 16 CHS of one code**

B3. Complexity evaluation of Step-2

For complexity evaluation we will use results of section B2 of this Appendix.

Case 1 complexity:

- $C_{step\_2\_1}$ = 1024*4=4096 real additions in one frame
- $C_{step\_2\_1}/C_{step-1}$ = 4096/1,996,800 = 0.0021 = 0.21% of Step-1 complexity

Case 2 complexity:

- $C_{step\_2\_2}$=1024*4*2*2= 16384 real additions
  - 4 means that we are using 4 CHSs, first 2 means that calculations 2 times per frame, second 2 means that we must check 7-th and 8-th slot after acquiring slot position in Step-2
- $C_{step\_2\_2}/C_{step-1}$ = 16384/1,996,800 = 0.0082 = 0.82% of Step-1 complexity

Case 3 complexity: $C_{step\_2\_3}$=1024*32*2*2=131072 real additions

- 32 means that we are using 32 CHSs, first 2 means that calculations 2 times per frame, second 2 means that we must check 7-th and 8-th slot after acquiring slot position in Step-2

$C_{step\_2\_3}/C_{step-1}$ = 131072/1,996,800 = 0.0656 = 6.56% of Step-1 complexity

For the target cell search (during handover) the typical number of code groups is 10. This will further reduce complexity in proportion to 32/10.

Appendix C Cyclic hierarchical sequences for Case 1, Case 2 and Case 3

Case 1

Code group 1:

- Frame 1: $C^{1,1}$
- Frame 2: $C^{1,9}$

...

Code group 32:

- Frame 1: $C^{4,8}$
- Frame 2: $C^{4,16}$

Case 2

Im this case, the code group pairs shown in Table 1 are transmiited in Frames 1&2 respectively. The same CHSs are used for both first (k) and second (k+8) slots.

Code group 1:

- Frame 1: $C^{1,1}$ in first and second slots (k and k+8)
- Frame 2: $C^{1,9}$ in first and second slots (k and k+8)

..

Code group 32:

- Frame 1: $C^{4,8}$ in first and second slots (k and k+8)
- Frame 2: $C^{4,16}$ in first and second slots (k and k+8)

Case 3

Code group 1:

- Additional bits of SCH from transport channel: 000
  - Frame 1: $C^{1,1}$ in first and second slots (k and k+8)
  - Frame 2: $C^{1,9}$ in first and second slots (k and k+8)

- Additional bits of SCH from transport channel: 001

- Frame 1: $C^{1,2}$ in first and second slots (k and k+8)
- Frame 2: $C^{1,10}$ in first and second slots (k and k+8)

...

- Additional bits of SCH from transport channel: 111
  - Frame 1: $C^{1,8}$ in first and second slots (k and k+8)
  - Frame 2: $C^{1,16}$ in first and second slots (k and k+8)

...

Code group 32:

- Additional bits of SCH from transport channel: 000
  - Frame 1: $C^{32,1}$ in first and second slots (k and k+8)
  - Frame 2: $C^{32,9}$ in first and second slots (k and k+8)

- Additional bits of SCH from transport channel: 001
  - Frame 1: $C^{32,2}$ in first and second slots (k and k+8)
  - Frame 2: $C^{32,10}$ in first and second slots (k and k+8)

...

- Additional bits of SCH from transport channel: 111
  - Frame 1: $C^{32,8}$ in first and second slots (k and k+8)
  - Frame 2: $C^{32,16}$ in first and second slots (k and k+8)

Appendix D Typical auto and cross correlation functions

Typical auto and cross correlation functions are presented. As an example, in some cases we used scrambling with gold sequence 1 and gold sequence 2 (first two sequences of the set of 17 orthogonal gold codes 256 chips length used for Secondary Synchronization Codes in proposal “The ETSI UMTS Terrestrial Radio Access (UTRA) ITU-R RTT Candidate Submission, ETSI SMG2, May/June 1998”.

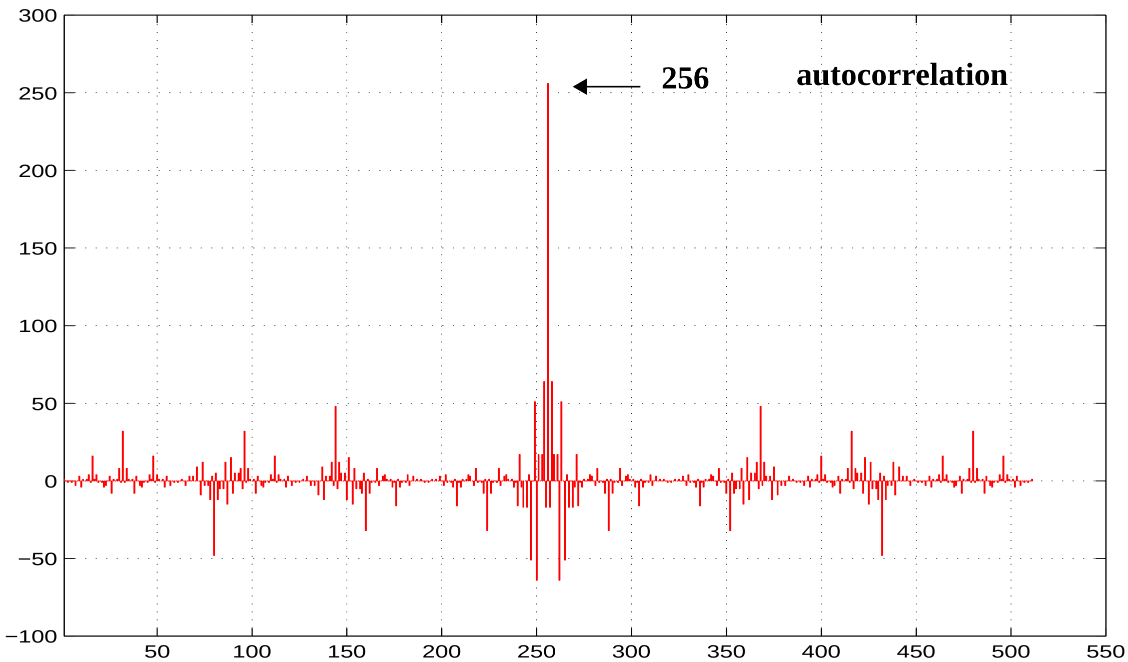


Figure 3: Aperiodic auto correlation function of the generalized hierarchical golay sequence of the Primary code

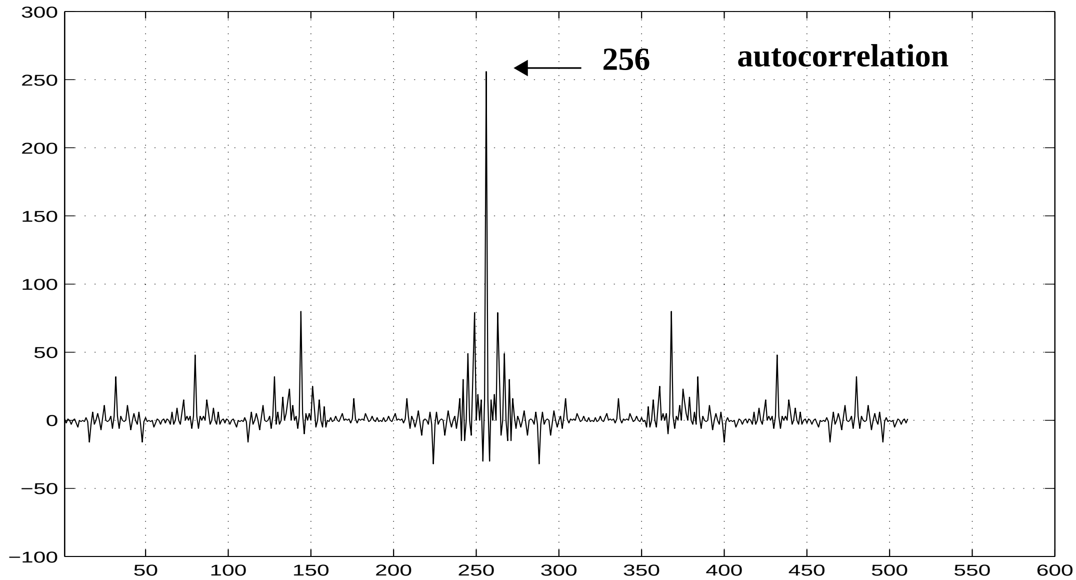


Figure 4: Aperiodic autocorrelation correlation of the cyclic hierarchical sequence C^{21}

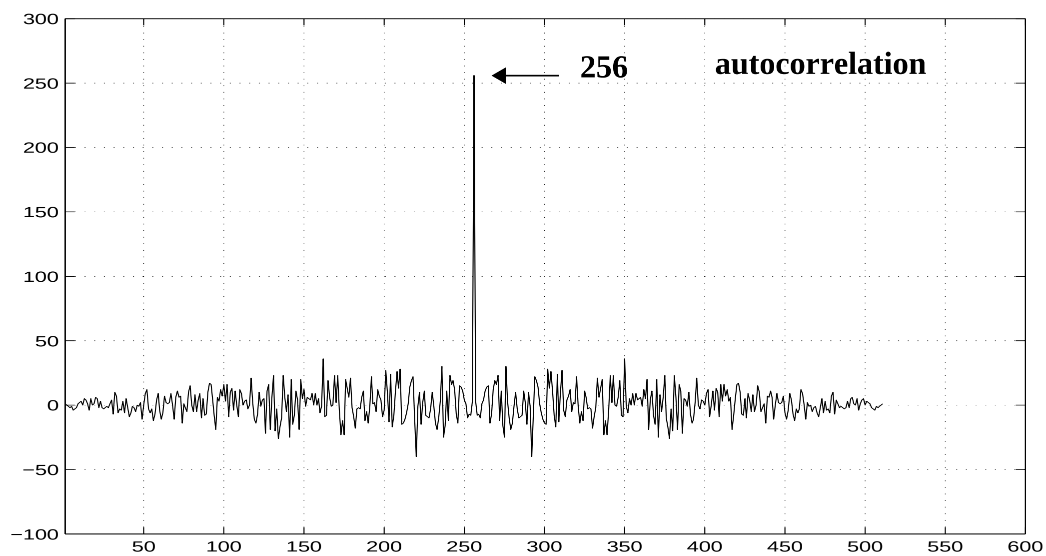


Figure 5: Aperiodic autocorrelation correlation of the cyclic hierarchical sequence C^{21}, scrambled with gold sequence 1

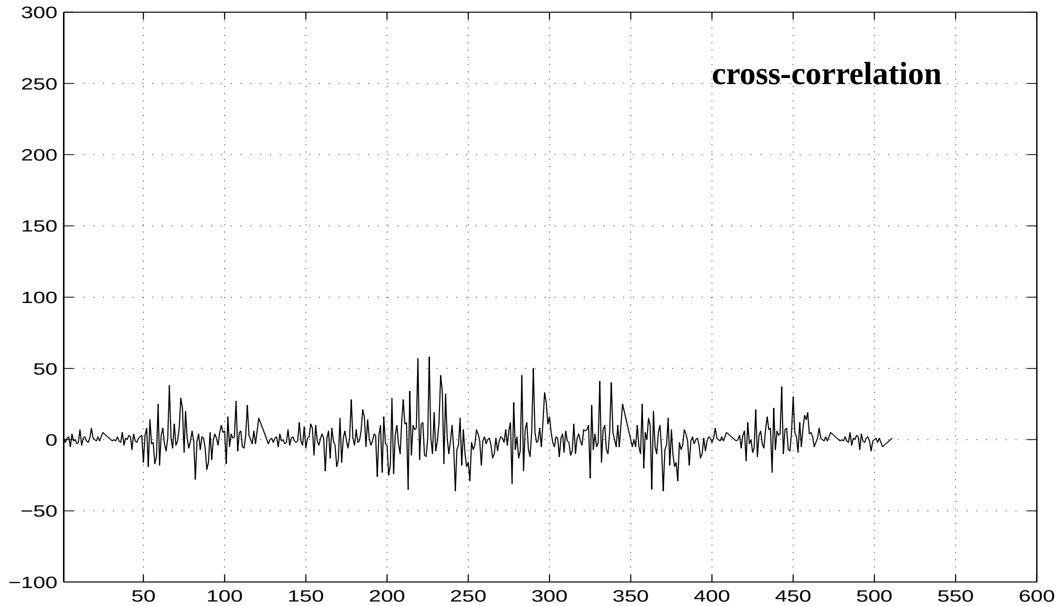


Figure 6: Aperiodic cross-correlation of the Primary code with the cyclic hierarchical sequence C^{21}

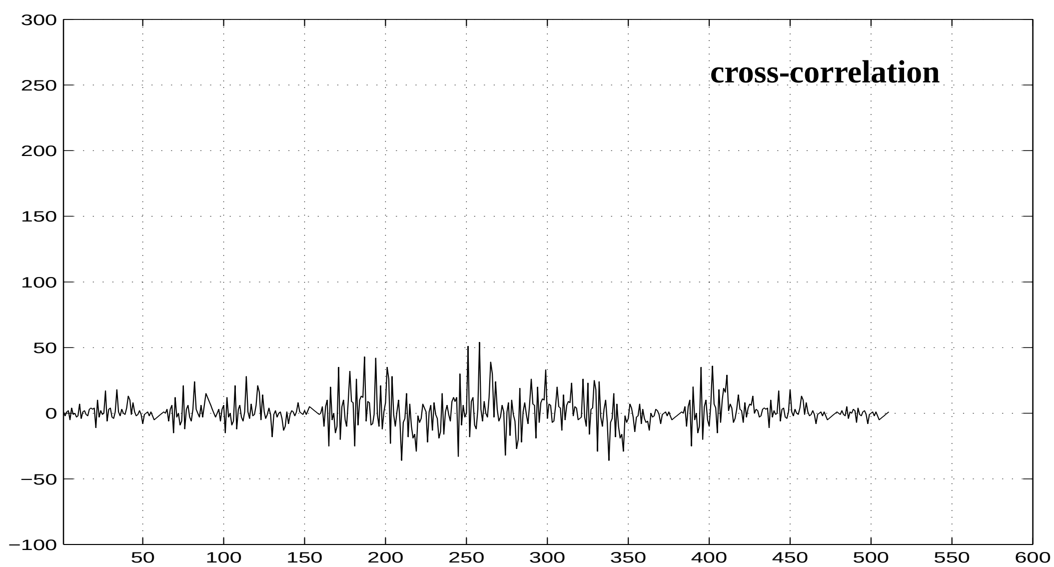


Figure 7: Aperiodic cross-correlation of the Primary code with the cyclic hierarchical sequence C^{23}

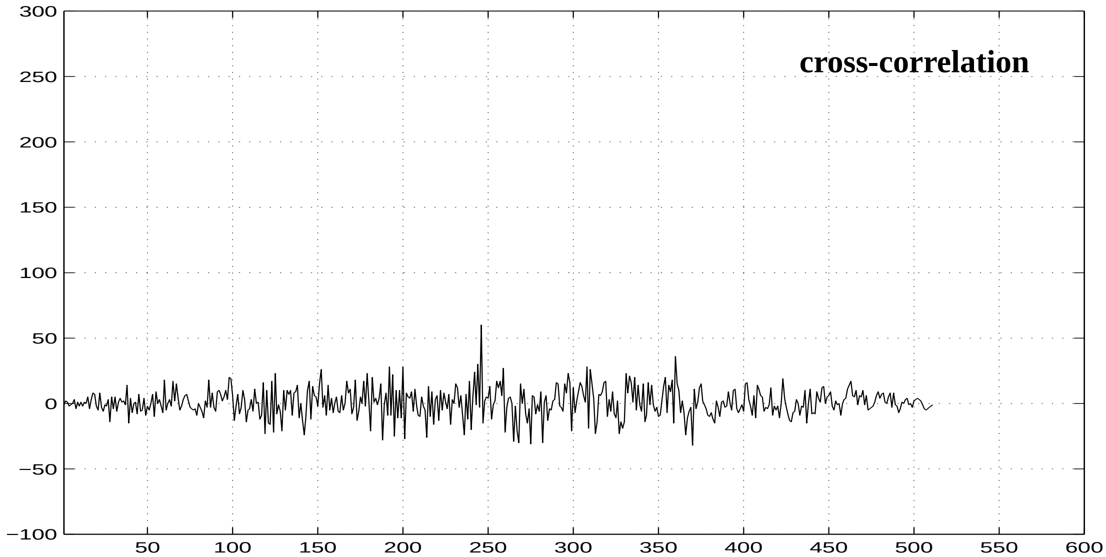


Figure 8: Aperiodic cross-correlation of the Primary code with the cyclic hierarchical sequence C^{21}, scrambled with gold sequence 1

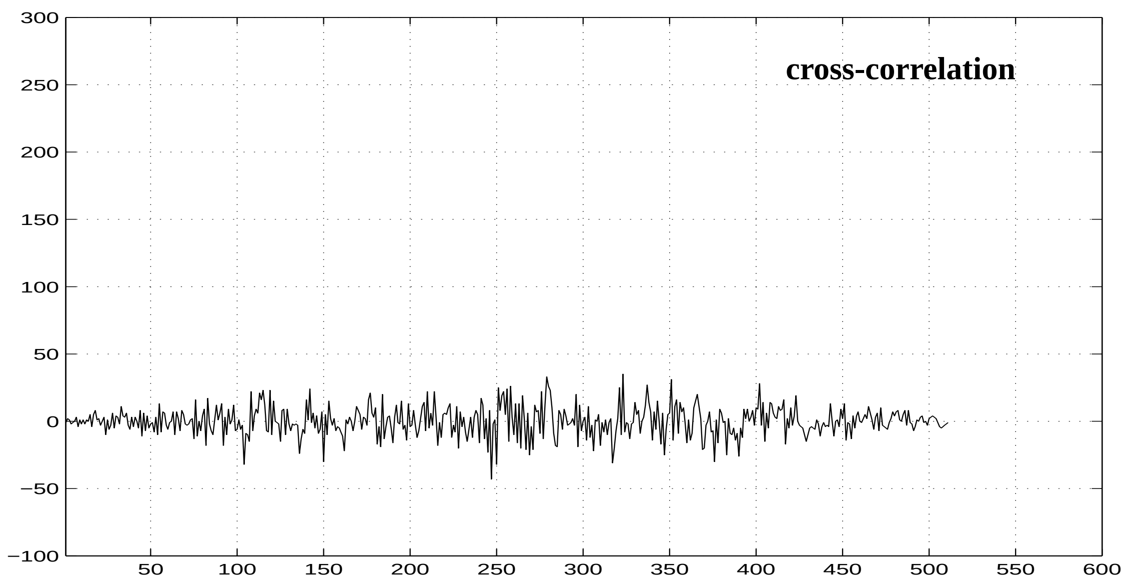


Figure 9: Aperiodic cross-correlation of the Primary code with the cyclic hierarchical sequence C^{23}, scrambled with gold sequence 1

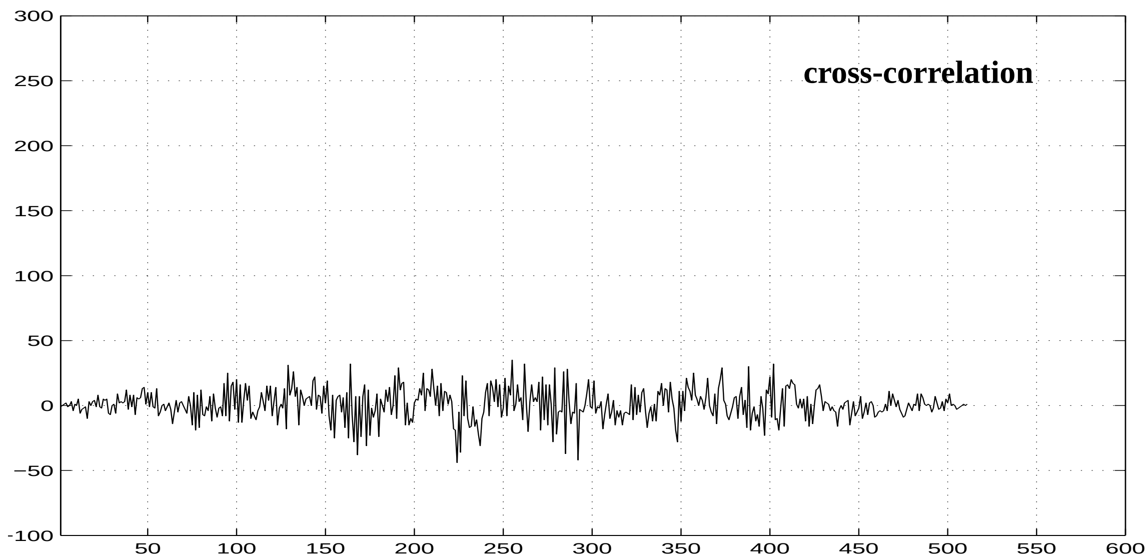


Figure 10: Aperiodic cross-correlation of the Primary code with the cyclic hierarchical sequence C^{21} , scrambled with gold sequence 2

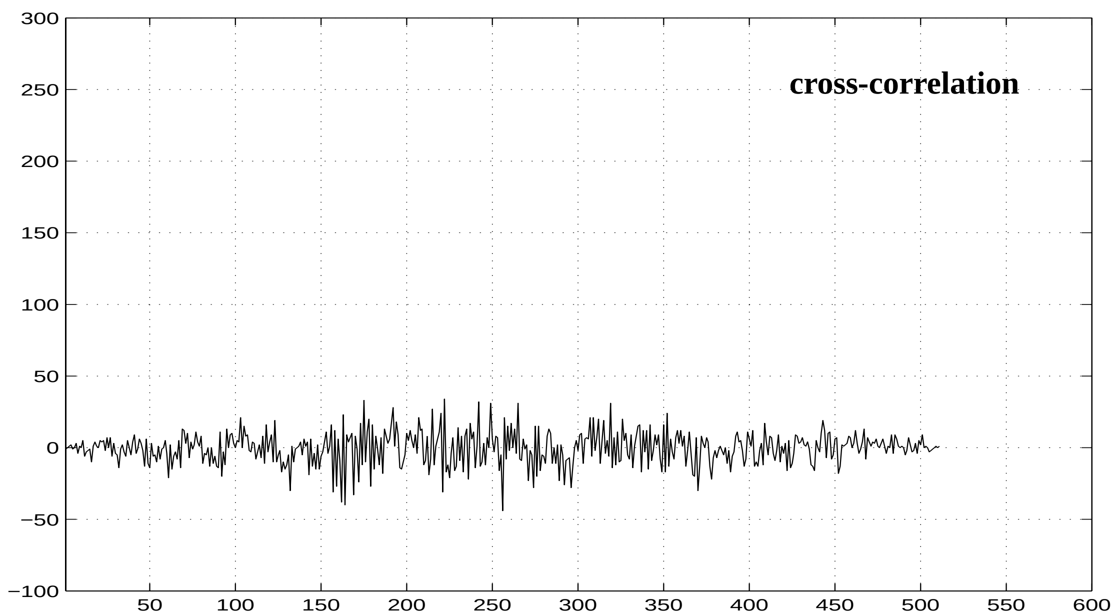


Figure 11: Aperiodic cross-correlation of the Primary code with the cyclic hierarchical sequence C^{23} scrambled with gold sequence 2