3GPP TR 25.836 V4.1.0 (2001-03)

Technical Report

3rd Generation Partnership Project;
Technical Specification Group Radio Access Network;
NodeB Synchronisation for TDD
(Release 4)

Keywords

UMTS, radio, synchronization

3GPP

Postal address

3GPP support office address

650 Route des Lucioles - Sophia Antipolis
Valbonne - FRANCE
Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Internet

http://www.3gpp.org

Contents

Foreword

This Technical Specification has been produced by the 3rd Generation Partnership Project (3GPP).

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

- x the first digit:
  - 1 presented to TSG for information;
  - 2 presented to TSG for approval;
  - 3 or greater indicates TSG approved document under change control.
- y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.
- z the third digit is incremented when editorial only changes have been incorporated in the document.

1 Scope

This TR describes the solution recommended to enable the synchronisation of NodeBs in UTRA TDD beyond that included in Rel. 99.

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[1] 3GPP TS 25.123: "Example 1, using sequence field".

[2] 3GPP TR 29.456 (V3.1.0): "Example 2, using fixed text".

3 Definitions, symbols and abbreviations

3.1 Definitions

3.2 Symbols

3.3 Abbreviations

4 Background and Introduction

NodeB synchronisation for TDD is a release 2000 work item that was agreed in RAN#7 plenary meeting. This work item involves the introduction of functionality to enable nodeBs to be synchronised.

This report identifies the required modifications within the UTRA layers 1/2/3. The method described is in addition to the Rel. 99 feature of the synchronisation port contained in TS 25.402.

5 Motivation

Cell synchronisation is planned for UTRA TDD in order to fully exploit the system capacity. There are several factors, that have an impact on the system capacity. The most important ones are:

- Inter-slot interference: without frame synchronisation there could be leakage from an UL timeslot into a DL timeslot, especially crucial for the UE due to the potentially close distance between UEs and the near-far effect.

- neighbouring cell monitoring: In TDD mode, certain measurements have to be performed in certain parts of certain timeslots of neighbouring cells. Without cell synchronisation, the UE would have to synchronise itself before being able to perform the measurements.

- Handover: The TDD mode may use timing advance in order to align receptions from all UEs at the cell's receiver. After a handover, the UE has to start transmission in the new cell with a timing advance value as good as possible. With the assumption, that the TDD cells are synchronised to each other, the handover performance can be optimised.

6 Accuracy Requirements

Several issues have been identified as of key importance in determining the accuracy requirements that the solution for synchronisation between cells should fulfill:

1) Impact of Time Error on Inter Slot Interference

2) Impact of Time Error on Timing Advance Adjustment for handover

3) Impact of Cell Timing Adjustments on UE receive and tracking performance

The minimum requirement for cell synchronisation accuracy in order to minimise the above impacts on the system performance has been defined by WG 4 to be 3 µs. The accuracy is defined as the maximum deviation in frame start times between any pair of cells that have overlapping coverage areas

In addition to the above requirement, the chosen solution should provide the option, that the accuracy can be enhanced, e. g. via more frequent measurements.

It is advantageous to signal information about the synchronisation accuracy to the UE for handover, so that the UE can apply the proper timing advance procedure as described in section 7.3.

7 Concept of Node B Synchronisation

7.1 General

In addition to proprietary means there are two ways to achieve cell synchronisation in a TDD system:

- Synchronisation of nodes Bs to an external reference via the synchronisation port standardised for Rel. 99

- Synchronisation of cells or Node Bs via the air interface described in this report for Rel. 4

The solution described in this report allows a mixture of both schemes, i. e. some cells may be synchronised over the air, some via the synchronisation port. In general, at least one time reference (e. g. GPS) is needed for each island of cells having connectivity to each other.

The RNC shall be the master of the synchronisation process, since the measurements either performed by a cell or by a UE, shall be signalled to and processed by the RNC.

A new procedure facilitates the transmission of measurements and commands between the RNC and the node B as well as allows the adjustment of the node B timing. Details of this procedure can be found in TR 25.838.

An optional phase may facilitate the frequency acquisision of node Bs at start-up prior to over-the-air synchronisation.

7.2 Synchronisation Procedure

The synchronisation procedure is based on using transmissions of cell synchronisation bursts in predetermined PRACH time slots based on an RNC schedule. Such soundings between neighbouring cells facilitate timing offset measurements by the cells. The timing offset measurements are reported back to the RNC for processing. The RNC generates cell timing updates that are transmitted to the Node Bs and cells for implementation. CEC sequences with multiple offsets are used as cell synchronisation bursts. The synchronisation procedure has three phases, the frequency acquisition phase, the initial phase and the steady-state phase. For Node Bs and cells with high accuracy frequency references, the frequency acquisition phase may be omitted. The procedure for late entrant cells is slightly different and is described separately.

Frequency Acquisition Phase

The procedure for frequency acquisition is used to bring cells of an RNS area to within frequency limits prior to initial synchronisation. This phase would allow cells to use low cost reference oscillators with accuracies in the order of several ppm. No traffic is supported during this phase:

1 The cell(s) identified as master time reference (e.g. containing the GPS receiver) shall transmit continuously cell sync bursts specified by higher layers (i.e. one in every time slot).

2 Initially all other cells shall be considered as unlocked (i.e. not in frequency lock).

3 While in this state, a cell shall not transmit, but shall listen for transmissions from other cells. The cell shall perform frequency locking to any transmission received.

4 When a cell has detected that it has locked its frequency to within 50 ppb of the received signal it shall signal completion of frequency acquisition to the RNC and begin transmitting the specified code. The exact timing of the code phase is unimportant.

5 When the RNC has received completion of frequency acquisition signals from all cells the frequency acquisition phase is completed.

Initial Synchronisation

1. The RNC sends a request over the relevant Iub to the cell(s) with reference clock for a timing signal. The RNC adjusts its clock appropriately, compensating for the known round trip Iub delay.

2. The RNC sends timing updates over the Iub to all the cells, apart from the one containing the reference clock, instructing them to adjust their clocks towards its own timing. Each of the timing offsets is again adjusted by the Iub round trip delay for that cell.

3. At this point, none of the cells is supporting traffic so a large proportion of the time can be given over to achieving synchronisation. It is assumed that there is as yet no information available on which to base the generation of a re-use pattern for sync transmissions. Thus all cells are instructed to transmit their cell sync bursts in turn one after the other with *no re-use*, i. e. the same sync burst sequence and offset is used by all cells.

4. All cells listen for transmissions and those which successfully detect a cell sync burst report their timing and received S/(N+I) to the RNC over the relevant Iub. Knowing the schedule, the RNC is able to determine the cell which made the transmission and place a measurement entry in the relevant place in its measurement matrix. After all cells have made their transmissions, the RNC computes the set of updates which will bring the cells nominally into synchronisation.

5. Steps 3 and 4 are repeated several times (typically 10). This serves two purposes:-

    - The rapid updates allow the correction of the clock frequencies as well as the clock timings to be adjusted in a short space of time. This rapidly brings the network into tight synchronisation.

    - The S/(N+I) values are averaged over this period. This provides more accurate measurements (averaging over noise and fading) which can be used in the automatic generation of a re-use plan.

6. The S/(N+I) values are used, automatically, to plan a re-use pattern. This is performed as follows:-

    - A matrix of minimal connectivity is computed on the basis of designating pairs of cells are minimal neighbours if either their estimated average S/(N+I) exceeds a threshold or if they have mutual neighbours.

    - The set of cells is divided into partitions of cells. Each partition must satisfy the requirement that no pairs of cells within that partition are minimally connected. All cells within a partition transmit the same code offset in parallel.

Steady-State Phase

7. All of the cells in the same partition are arranged to transmit / receive in the same cell sync frames according to the above procedure and they transmit the same code offset in parallel. All cells report the reception times for all relevant code offsets back to the RNC.

8. At the end of each cycle, the RNC collates the information. In general there should always exist a path of bidirectional valid measurements that link every cell either directly or indirectly to the cell with UTC capability. However, the model is arranged such that only those cells which have such a path will be updated on any given occasion.

9. The process of partition transmissions and updating then continues indefinitely.

Late entrant Node Bs

The scheme for introducing new nodeBs into a synchronized RNS is as follows:

1. There is a specialised sync transmission scheduled by higher layers. A single common code (i.e. with the same, nominally zero, shift) is transmitted in parallel by *all* NodeBs addressed. The late entrant NodeB will correlate against the specialised sync transmissions. The late entrant NodeB will take the earliest reception as the timing of the system.

2. Thus, at this point, the late entrant NodeB has obtained system time, subject to an unknown propagation delay between it and its nearest neighbour. The late entrant NodeB cannot, at this time, tell which of its neighbours *is* the nearest. However, this level of synchronisation is good enough that from then on the late entrant NodeB can distinguish the overlaid normal sync transmissions unambiguously for the various code shifts.

3. After this time the late entrant NodeB can measure the timings of sync transmissions received from specific NodeBs and report these to the RNC. In turn, the RNC can give the late entrant NodeB its own schedules for sync transmission and to use one or more of these. The RNC can then use the bi-directional sounding, which will then be available, to compute the true timing error and to instruct the NodeB to adjust its timing appropriately.

If the late entrant has an inaccurate clock the specialised cell sync burst transmission may be repeated often enough to allow full frequency searching.

7.3 Potential Improvements for Handover

For handover the UE may be provided with information about the synchronisation accuracy of the new cell so that it can apply the proper timing advance value and procedure in the new cell:

1. In large cells and low sync accuracy (3 us), HO will be done without TA, so that the maximum timing error would be twice the propagation delay, e. g. 6 us or 24 chips for 1 km radius . It might be advantageous to transmit a special burst type 3.

2. In large cells with medium sync accuracy (e. g. 0.5 us) HO will be performed with TA autonomously calculated by the UE. However, the UE will correct the calculated TA value for the sync inaccuracy. The maximum timing error would be for the example given 2 µs or 8 chips. For an estimation window of 57 chips this is be tolerable. Even the TA step size is 4 chips. The correction value is calculated in the RNC and signalled to the UE before handover execution.

3. In large cells with high sync accuracy (+- 100 ns) autonomous TA calculation in the UE will be used after HO. The maximum timing inaccuracy will be 400 ns.

In small cells no TA shall be used. The maximum timing inaccuracy will be twice the propagation delay.

The figures in the description above are just an example. The difference between small cells and large cells is whether TA has to be applied in these cells or not.

Higher layer signalling determines, whether TA is used in a cell or not and whether it shall be applied in a new cell after handover.

If the synchronisation accuracy is signalled in the handover phase, then the UE shall adjust its autonomously calculated TA value accordingly.

8 Impact on Interfaces

8.1 Uu Interface

There is an impact in the transmitting as well as in the receiving cell.

In the transmitting cell, a RACH timeslot is blocked during synchronisation burst transmission. In the receiving cell the Synchronisation Burst is interfered by PRACH bursts as well as it generates interference for these. In order to avoid these known losses of RACH transmissions, the UEs may be informed about timeslots that are blocked for cell sync burst transmissions.

The cell sync codes to be used are **C**oncatenated periodically **E**xtended **C**omplementary sequences as described in more detail in R1-00-1351 (see Annex A). Different code offsets are used in order to preserve the correlation property of these sequences.

For handover the UE may be provided with information about the synchronisation accuracy so that it can apply the proper timing advance value and procedure in the new cell as described in section 7.3.

8.2 Iub Interface

The messages between a NodeB and the RNC are described in detail in TR 25.838.

In the uplink these messages are:

- Neighbouring cell measurements
- Timing information

In downlink these are:

- Timing adjustment commands
- Transmit and receive schedule

8.3 Iur Interface

Each RNC area is synchronised individually to at least one reference clock (e. g. GPS). This automatically ensures synchronisation between RNC areas. Therefore, no communication over Iur is necessary for cell synchronisation between RNC areas.

9 Impact on network elements

9.1 UE

The UE shall have the capability to take into account the blocking of timeslots in up- and downlink.

It shall support the synchronisation accuracy signalling mechanism and have the capability to correct its TA value for handover.

9.2 Node B

The cells shall support the reception of the Cell Synchronisation Bursts as well as measure the reception time. In addition, the cells have to support the transmission of the Cell Synchronisation Burst.

Furthermore, the cells shall have to provide means for adjusting their timing and optionally the clock rate on command. The changes in the NBAP protocol have to be supported.

9.3 RNC

The RNC has the control of the whole algorithm. It shall initialise, establish and maintain a connectivity plan. It shall collect measurements and compute adjustment commands as well as support the necessary NBAP signalling. It may estimate the synchronisation accuracy between cells and signal the relevant information to the UEs for handover. All algorithms involved in the computation of timing updates and schedules are proprietary.

10 Performance Analysis

A performance analysis is included in Tdoc R1-00-0074. There it is shown, that it is possible, to fulfill the accuracy requirements for support of LCS. The resource stealing for this is in the order of 0.5 % of the RACH capacity. The resource stealing neccesary for the minimum requirements is considerably lower and in the order of 0.04 % of the RACH capacity.

11 Backward Compatibility

UTRAN: The synchronisation over the air in Rel. 4 can be used in addition to and in combination with the synchronisation via the sync port in Rel. 99. Therefore, backward compatibility is ensured for the UTRAN.

UE: The Rel. 99 UE's are affected by the blocking of RACH resources in certain timeslots, i. e. these UEs are not aware, that their RACH transmission in certain timeslots cannot be successful. However, the algorithms involved in the RACH transmission have to cope with a RACH failure in any case.

Annex A:
(no title)

3GPP TSG RAN WG1#17 R1–00–1351

Stockholm / Sweden

November 21st - 24th, 2000

Agenda Item: **AH30**

Source: **Siemens, Mitsubishi Electric**

Title: **CEC sequences with multiple offsets for Node B sync in UTRA TDD**

Document for: **Discussion and Decision**

Summary

It has been proposed to use **C**oncatenated periodically **E**xtended **C**omplementary sequences 0 for inter-base station synchronisation in UTRA TDD. CEC-sequences provide a perfect channel estimation window, i.e. no auto-correlation side lobes at all around the main correlation peak in a window of adjustable size, whilst still exhibiting excellent auto-correlation properties for the overall aperiodic auto-correlation function. Due to the existence of low complexity matched-filter structures for Polyphase complementary pairs, a significant computational complexity reduction can also be achieved for correlation with CEC-sequences.

The option that several Node B's within one RNS transmit their cell sync bursts simultaneously, i.e. in the same PRACH timeslot has been proposed recently 0. The introduction of this option into the current Node B sync concept could allow a more efficient usage of the allocated resources and could also allow more frequent measurement occasions. A straightforward approach for enabling simultaneous reception and detection of more than one neighbouring Node B is to assign them different code offsets by means of cyclically shifted versions of one common basic sequence.

In this contribution, the construction of CEC-sequences is extended to the multiple code offset case and it is shown that these Node B sync sequences offer the same advantages as the original ones in terms of their auto-correlation properties and low-complexity receiver implementation.

Introduction

The construction principle of the original CEC-sequences as proposed in 0 is shown in Figure 1. The basic sequences *s(n)* and *g(n)* make up a Golay or Polyphase complementary pair with an integer power of 2 as length. The sum of the aperiodic auto-correlation functions of a complementary pair yields a perfect Dirac-function.

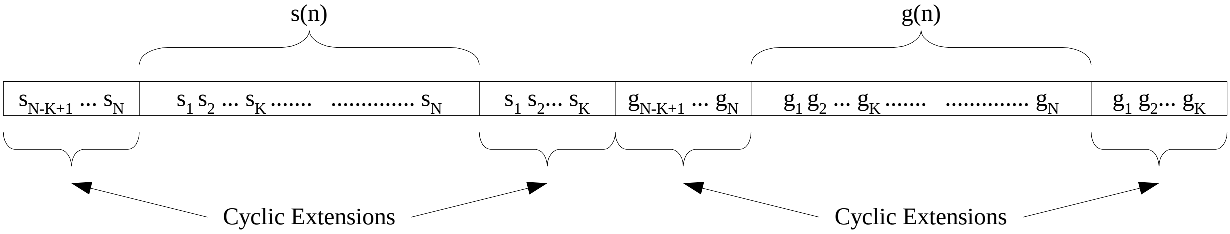

Figure 1: Construction principle of the CEC-sequences without code offset

The receiver in a first step correlates the overall received signal separately with a local replica of *s(n)*. In a second step, it correlates at a *N+2 K* chip offset with a local replica of *g(n)*. Finally, the auto-correlation sum is obtained after adding up corresponding matched-filter outputs.

It can be shown that following the construction principle in Figure 1, a perfect auto-correlation window of size ±*K* can be obtained around the main correlation peak. The size of the perfect auto-correlation window is scalable and dependent on the length of the pre- or post extensions. In addition, the overall aperiodic auto-correlation properties, i.e. outside the window are better than can be obtained by a Gold-sequence of comparable length.

In an alternative way, CEC-sequences could be constructed from a complementary pair by either leaving out the pre- or the post-extension for each of the basic sequences . Without pre-extensions, the overall CEC-sequence would look like shown in Figure 2.

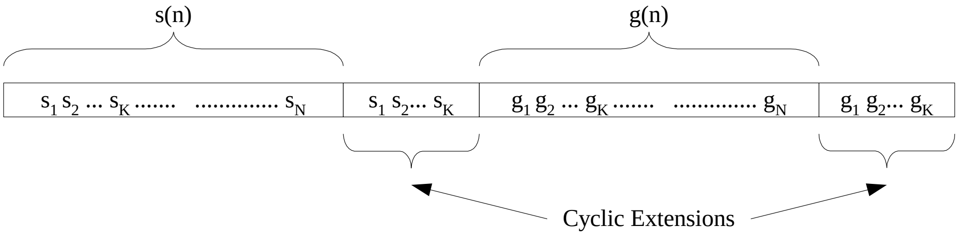

Figure 2: CEC-sequences with post-extensions only

The receiver would in a first step correlate the received signal with a cyclically rotated version of *s(n)*, here denoted as *s'(n)*. The elements of this local replica *s'(n)* are obtained from the original basic sequence *s(n)* as being,

$$s'_i = \begin{cases} s_{(K-1)/2+i} & i \le N-(K-1)/2 \\ s_{(K-1)/2+i-N} & i > N-(K-1)/2 \end{cases}$$

Here, *K* denotes the length of the post-extension and *N* the length of the complementary pair. If *K* is an odd number, the nominal code correlation position starts with s_i where $i = (K-1)/2+1$.

The receiver would in a second step proceed in an analogue way correlate the received signal with a cyclically rotated version of *g(n)* at the time offset *N+K* chips and finally add up corresponding correlation values from the first and second step in order to obtain the auto-correlation sum.

When removing the pre-extension and correlating with a cyclically rotated version of the basic sequences, the size of the perfect auto-correlation window around the main correlation peak is reduced to ±*K*/2. If *K* is not an odd number, the perfect auto-correlation window becomes very slightly asymmetrical.

CEC-sequences derived following either Figure 1 or Figure 2 are equivalent, both have excellent aperiodic auto-correlation properties and for both the possibility to use low-complexity receiver structures for Polyphase complementary pairs exists. However, CEC-sequences with post-extension only, such as shown in Figure 2 are conceptually closer to the multiple code offset case that is described in the next section.

CEC-sequences with multiple code offsets

If CEC-sequences are derived as in Section 0, a single Node B sync sequence is obtained from a single Golay or Polyphase complementary pair. When several Node B's in a RNS shall be enabled to transmit simultaneously, they can be differentiated by either

1) using different complementary pairs or

2) using different code offsets of the same complementary pair

for constructing different Node B sync sequences.

The second option offers the advantage that because of the perfect auto-correlation sum property of complementary pairs, orthogonality is preserved between different Node B sync sequences derived from the same complementary pair by means of a different code offset. It is therefore advantageous to generate a family of CEC-sequences from one particular Golay or Polyphase complementary pair by allowing variable cyclic shifts of the basic sequences *s(n)* and *g(n)*. The construction principle is shown in Figure 3 and Figure 4.

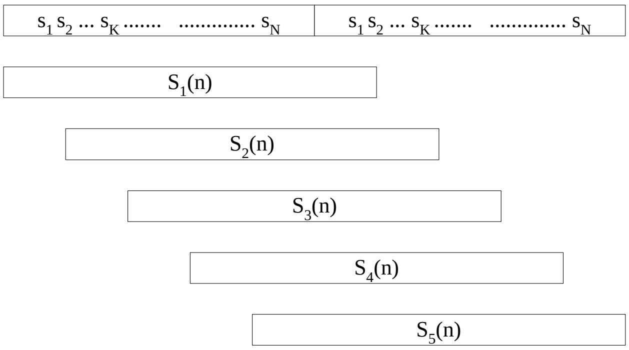

Figure 3: Deriving different code offset versions of the basic sequence s(n)

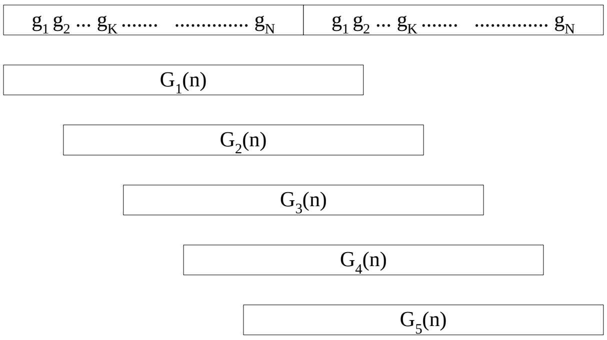

Figure 4: Deriving different code offset versions of the basic sequence g(n)

The cyclically shifted versions of *s(n)* and *g(n)*, referred to as $S_m(n)$ and $G_m(n)$ for code offset *m* are derived by selecting appropriate elements from the repetitions of *s(n)* and *g(n)* respectively. The periodically repeated version of *s(n)* is denoted by $s_e(n)$, with its elements given as $s_{e,i} = \begin{cases} s_i & i \le N \\ s_{i-N} & i > N \end{cases}$.

Then the elements of $S_m(n)$, denoted as $S_{m,i}$ are given by $S_{m,i} = s_{e,i+(m-1)w}$, where *w* is the offset in terms of the number of code elements. Typically *w* is chosen to equal *K* and the total available number of offsets *M* is then given by $M=N/K$ although other relationships are not excluded. The corresponding cyclically shifted versions of $G_m(n)$ are constructed in identical fashion.

The overall Node B sync sequence derived from a particular Golay or Polyphase complementary pair *s(n)* and *g(n)* and corresponding to a particular code offset *m* is finally given by the concatenation of $S_m(n)$ and $G_m(n)$ as illustrated in Figure 5. Node B sync sequences build from CEC-sequences with multiple code offsets have an overall length of $2(N+K)$ chips.

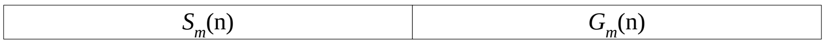

Figure 5: Node B sync sequence derived from a complementary pair s(n) and g(n) for code offset m

The receiver in a first step cyclically correlates the first half of the overall received signal with a local replica *s'(n)* whose elements are derived from *s(n)* as in Section 0. The first *K* chips of the *N+K* chip long segment corresponding to *s(n)* are discarded. This is equivalent to the computation of the periodic auto-correlation with the local replica *s'(n)* by means of a cyclic shift register.

The correlation with *g'(n)* is done in a second step in an analogue manner on the second half of the overall received signal. By discarding the first *K* chips, any undesired cross-correlation between the parts corresponding to *s(n)* and *g(n)* due to a multi-path channel with channel impulse response length smaller than *K* can be avoided. Finally, the auto-correlation sum is obtained after adding up corresponding matched-filter outputs from the first and second step.

A typical auto-correlation obtained for the case of *N=1024* and *M=16* possible code offsets with a resolution of *K=64* chips between different simultaneously transmitting Node B's is shown in the Appendix for the case of code offsets *1, 3* and *7* being present.

Conclusion

CEC-sequences with multiple code offsets have all the benefits of the original proposed CEC-sequences in terms of ease of implementation of the decoder 0 and ideal auto-correlation properties. In addition, multiple code offsets are available for a particular Golay or Polyphase complementary pair. Due to the complementary property of the CEC-sequences their offset versions also remain orthogonal, i.e. without any undesired cross-correlation.

Golay complementary pairs of length *N=1024* seem to be a good choice for building the CEC-sequences, as the EGC-receiver structure simplifies the most for this special binary case of Polyphase complementary pairs. Providing *M=16* possible code offsets for one RNS leaves *K=64* chips of resolution between different Node B's which should be more than sufficient. The overall length of a Node B sync sequence would then be *2176* chips which yields a maximum usage of the available time in the cell sync timeslots. Also, we propose that at least *8* Golay complementary pairs are chosen for deriving the CEC-sequences with multiple code offsets. These basic Golay complementary pairs could be chosen based on their aperiodic auto-correlation properties which are important in the initial Node B sync scenario.

Note also that when applying a continuously increasing phase offset to the elements of a Node B sync sequence derived from CEC-sequences, the order of applying the phase offset and deriving the code offset versions are inter-changeable as all code parameters are multiple's of *4*. The same holds for the receiving side.

References:

[1] R1-00-1349, "Node B synchronisation for TDD – some refinements", Siemens

[2] R1-00-1181, "Sequences for the Node B synchronisation burst", Mitsubishi Electric

[3] R1-00-0946, "Sequences for the Cell Sync burst", Siemens

[4] R1-00-0074, "Node B Synchronisation for TDD", Siemens

[5] R1-99-g42, "Synchronisation of Node B's in TDD via Selected PRACH timeslots", Siemens

[6] M.J.E. Golay, "Complementary Series", IRE Trans. on Information Theory, Vol.IT-7,pp.82-87, April 1961

Appendix

The following figures show a typical auto-correlation sum for CEC-sequences with multiple code offsets. The code offset versions of the CEC-sequence are derived from a Golay complementary pair with weight vector **W**=[W_1 W_2 … W_{10}]=[1 -1 1 1 -1 -1 -1 -1 -1 -1] and permutation vector **P**=[P_1 P_2 … P_{10}] =[9 0 8 1 7 2 6 3 5 4]. Code offsets *1, 3* and *7* were selected.

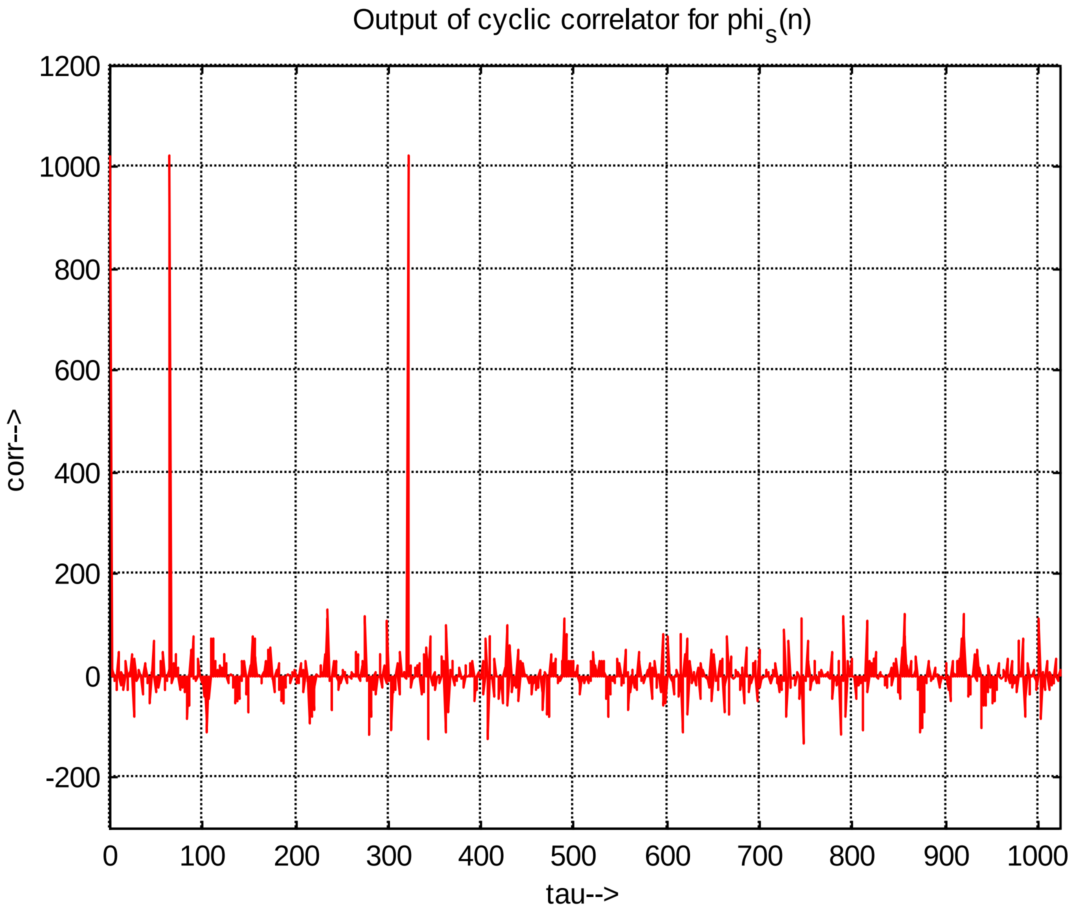

Figure 6: Cyclic correlator output for correlation with s'(n)

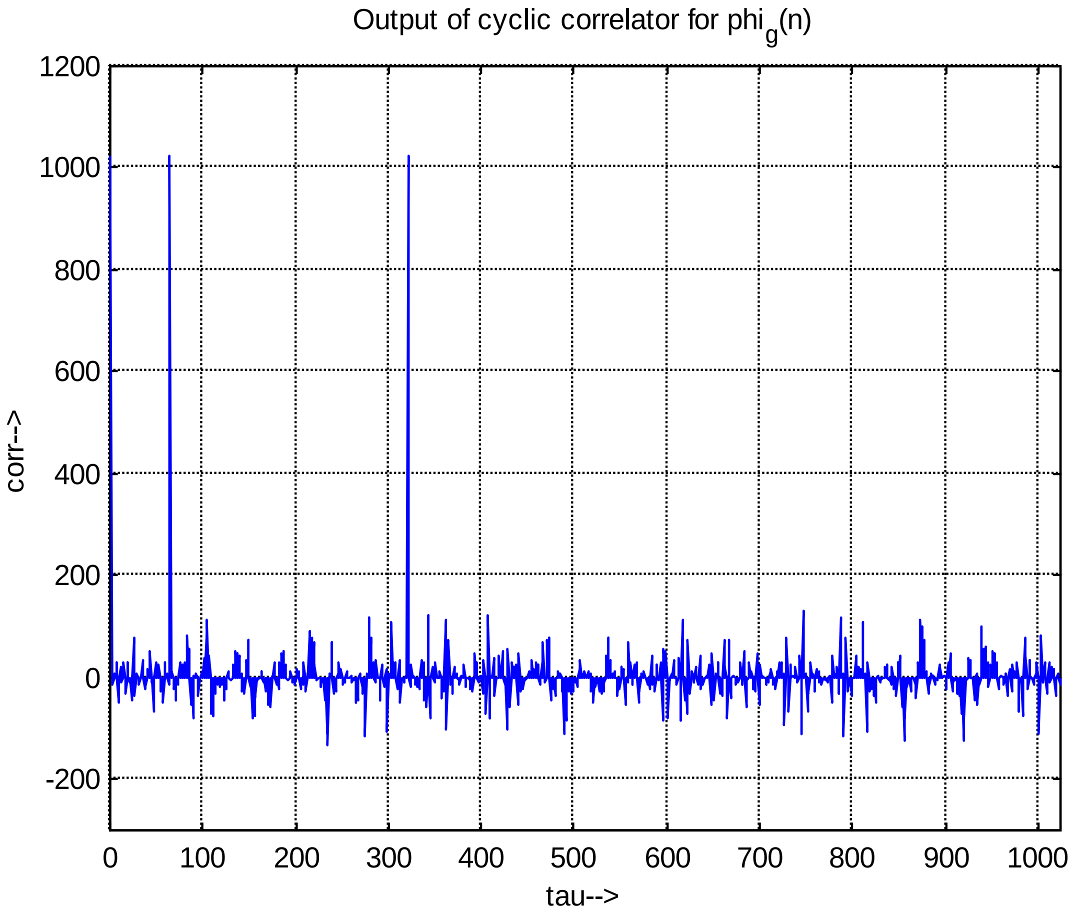

Figure 7: Cyclic correlator output for correlation with g'(n)

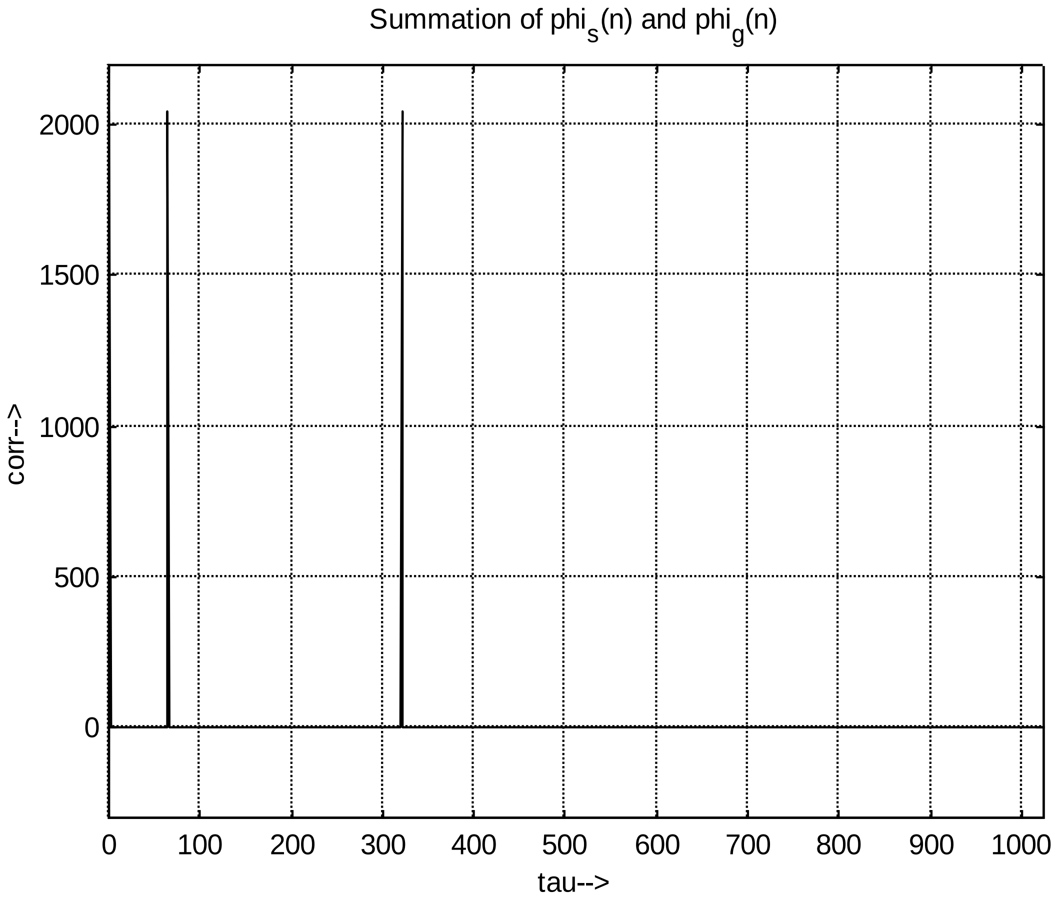

Figure 8: Sum of the correlation outputs obtained by cyclic correlation with s'(n) and g'(n)

Annex A:
Change history

Change history							
Date	TSG #	TSG Doc.	CR	Rev	Subject/Comment	Old	New
15/12/00	RAN_10	RP-000547	-	-	Approved at TSG RAN #10 and placed under Change Control	-	4.0.0
16/03/01	RAN_11	RP-010073	001	1	Additions to the node B synchronisation procedure	4.0.0	4.1.0