University of Rome "La Sapienza"
faculty of Engineering
INFO-COM Dpt.

M.Sc. Course of
Telecommunications

Teaching
Electrical Communications

Synchronization through Cell Search in an asynchronous Ultra Wide Band wireless access network

Thesis by

Simone Ratti

Supervising professor

Maria Gabriella Di Benedetto

Reader

Maria Stella Iacobucci

Degree Year 2001/2002

Abstract

This work addresses the problem of synchronization in impulse radio multiple access mobile networks. The attention is focused on a system whose architecture is composed of Mobile Terminals and asynchronous Access Points. Time hopping techniques are first reviewed, and the synchronization problem is introduced together with a detailed Cell Search procedure. The physical channels for temporal synchronization are presented and an algorithm that achieves complete synchronization between Mobile Terminals and the Access Point is described. Moreover, a choice of suitable time hopping codes for the synchronization channels is proposed, together with simulation results that testify their applicability.

The contents of this thesis are in part illustrated in an article which has been submitted by the author and successively accepted by an international committee for the presentation at the "Fourth IEEE Conference on Mobile and Wireless Communications Networks (MWCN 2002)" which will take place in Stockholm Sept 9-11, 2002. At this regard the article is referenced as follows:

Paper No: 272
Title: Cell search in Ultra Wide Band Time Hopping Asynchronous Networks.

Contents

List of Figures

List of Tables

List of Acronyms

MT Mobile Terminal

UWB Ultra Wide Band

UWB-IR Ultra Wide Band Impulse Radio

IR Impulse Radio

TH-SS Time Hopping Spread Spectrum

PPM Pulse Position Modulation

GPS Global Positioning System

LOS Line of Sight

UI Unfriendly Interceptor

RX Receiver

SNR Signal to Noise Ratio

FL Forward Link

TX Transmitter

RL Reverse Link

NS Next Search

SASPA Sequential Ascending Search Power Algorithm

SBSA Sequential Bayesian Search Algorithm

TSC Time-Sliding Correlator

TIC Time-Integrating Correlator

CI Call Initiating

CT Call Terminating

RXe Receiving Equipment

TXe Transmitting Equipment

ph-CH Physical Channel

l-CH Logical Channel

TCH Traffic Channel

CCH Control Channel

L-frame Logical Frame

L-slot Logical Slot

BCH Broadcast Channel

BCH-id Broadcast Channel Identity

MAC Media Access Control

GSCH Global Synchronization Channel

PSCH Primary Synchronization Channel

PSC Primary Synchronization Code

SSCH Secondary Synchronization Channel

SSC Secondary Synchronization Code

CPC Cyclically Permutable Code

HCC Hyperbolic Congruence Code

HCCX Extended Hyperbolic Congruence Code

UH Undesired Hit

HCfC Hyperbolic Congruence Frequency Hopping Code

HCtC Hyperbolic Congruence Time Hopping Code

HCfCX Extended Hyperbolic Congruence Frequency Hopping Code

HCtCX Extended Hyperbolic Congruence Time Hopping Code

PNC Pseudo Random Code

iPNC Informatically Generated Pseudo Random Code

IPv Internet Protocol version ...

LAN Local Area Network

RACH Random Access CHannel

MT-id Mobile Terminal Identity

EC-id Equipment Class Identifier

AGCH Access Grant CHannel

Part I

General concepts

Introduction

Wireless mobile networks represent a preeminent research field in the telecommunications world. Their flexible structure very often allows them to be exploited very rapidly overriding installation difficulties (for example geographical obstacles) which could prevent traditional networks from being implemented. This advantage is counterbalanced by the necessity to find a solution to a number of problems brought about by the choice of radio resources to convey the communication channels required. Radio systems suffer from the stochastic nature of the electromagnetic transmission means adopted, which imposes a particular attention in the whole project. With *mobility* it is intended the global capacity of the terminals (i.e. *Mobile Terminals MTs*) to move inside the desired environment maintaining real time reciprocal connectivity. It is clear that wireless techniques play a fundamental role if the mobility is an intended goal. No cable can guarantee a sufficient

level of freedom to the user who needs to take advantage of the services provided by the network. Each communication will always require the temporal synchronization among the interacting partners. In a wireless environment all this must be done despite the adversing variability of the transmission delays encountered.

The technical solution adopted here for the physical layer will consist of the resources made available by the Ultra Wide Band radio technique (often referred to as UWB). A lot of studies have been already developed in the past mainly looking at the UWB signal as a new perspective for radar equipments. The approach which follows is directed towards its use as the support for communication systems. Before going over, it is necessary to develop a comprehensive introduction about UWB and in particular about the modulation format which can be adopted to convey the information involved in communications.

1

Ultra Wide Band basic knowledge

1.1 Introduction to impulse radio systems

Ultra Wide Band Impulse Radio (UWB-IR) is first introduced in the extensive works presented in [1], [2] and [3]. The main aspects of UWB-IR will be

described in the following sections mainly drawing on those above-mentioned works which opened the way to the research in the communication field. As well known, the UWB transmitted signal occupies an extremely large bandwidth, even in the absence of modulation. This bandwidth can be considered equal, as a rule of thumb, to a value greater than 1/4 of the center frequency measured at the $-10\ dB$ points. The advantage of this approach includes the possibility of communicating using a very low power spectral density, in contrast to the traditional narrow bandwidth transmissions, since the total power is spread over a wide range of frequencies. The result is a signal that is more covert, has higher immunity to interference effects, and has an improved time-of-arrival resolution. The UWB technique is thus based on Spread Spectrum; however it does not use a sinusoidal carrier and communicates with a baseband time-hopping signal composed of sequences of sub-nanosecond pulses (referred to as *monocycles*). The use of a frequency range which is low positioned on the frequency axis, despite the degradation due to the lack of a carrier moving the signal in a well behaving frequency range, brings the following benefits:

- it improves the capability of penetrating materials which tend to be increasingly opaque at higher frequencies;
- since an UWB monocycle allows to resolve delays of the order of a nanosecond or less and distances on the order of a foot or less it reduces

the multipaths fading effects in indoor environments.

These properties constitute the main reasons for promoting impulse radio technology (IR) as a chief candidate for high quality, fully mobile, short range indoor radio systems. The low level of the power spectral density hopefully ensures that impulse radios systems will not interfere with narrow-band radio systems operating in dedicated bands. At this regard, a main objective in UWB technology is to generate a spectrum as flat as possible in order to appear to other systems as gaussian noise. The modulation of Time Hopping Spread Spectrum (TH-SS) impulse radio is accomplished through time shift of pulses (Pulse Position Modulation, PPM). The transmitted signal is obtained as explained in the following section.

1.2 Time Hopping with Impulses: modulation format

The UWB-IR baseband signal is composed by pulses transmitted with a fixed period of repetition. The modulation of information bits, together with the address or identity of the channel, is assured by a predetermined sequence of shifts of the pulses from nominal positions (PPM). A pulse, or a *monocycle*, indicates a shape with a temporal width on the order of about a nanosecond, similar in shape to one cycle of a sine wave. An interesting example of

received monocycle presented in [1], [2] and [3] has the following expression:

$$w_{rec}(t+0.35) = \left[1 - 4\pi\left(\frac{t}{\tau_m}\right)^2\right] \exp\left[-2\pi\left(\frac{t}{\tau_m}\right)^2\right]. \quad (1\text{-}1)$$

The analytical representation of the baseband signal is obtained by substituting equation (1-1) in:

$$s(t) = \sum_{j=-\infty}^{+\infty} w(t - jT_f) \quad (1\text{-}2)$$

which represents the train of pulses adopted to vehicle the desired information. This is only one element making up the transmitted UWB-IR signal which is composed as follows:

Uniformly Spaced Pulse Train: as told it consists of monocycle pulses $w(t)$ which are spaced by T_f seconds (see equation (1-2)). The *frame time* or *pulse repetition time* T_f typically may be a hundred to thousand times the monocycle width T_m with its largest value constrained in part by the stability of the available clocks. The result is a signal with a very low duty cycle. Multiple access signals composed of uniformly spaced pulses are vulnerable to occasional catastrophic collisions.

Pseudorandom Time Hopping: to eliminate catastrophic collisions in multiple access, each link (or user indexed with k) is assigned a distinct pulse shift pattern $\{c_j^{(k)}\}$ which is called *time hopping code*. These time hopping codes $\{c_j^{(k)}\}$ are periodic pseudorandom codes with period N_p,

i.e., $c_{j+iN_p}^{(k)} = c_j^{(k)}$ for all integers j and i. Each code element is an integer in the range

$$0 \leq c_j^{(k)} < N_h$$

where N_h is the number of possible pulse shifts within a time frame T_f and satisfies the condition $N_h T_c \leq T_f$. The time hopping code provides an additional time shift to each pulse in the train, with the j^{th} monocycle undergoing an added shift of $c_j^{(k)} T_c$ seconds. This can be seen in figure 1.1 where it is displayed the signal derived from the code sequence [2 1 3 5] generated for $N_h = 6$. The monocycles transmitted occupy those T_c, within successive T_f, that correspond to the positions indicated in the sequence. The ratio $(N_h \cdot T_c)/T_f$ indicates the fraction

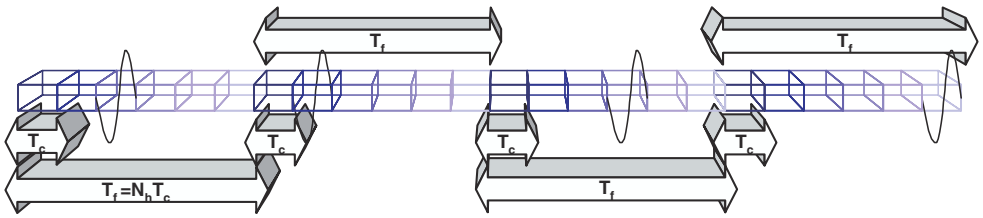

Figure 1.1: Pseudorandom time hopping.

of the frame time T_f over which time hopping is allowed. Since a short time interval is required to read the output of a monocycle correlator and to reset the correlator, we assume that $(N_h \cdot T_c)/T_f$ is strictly less than one. If $N_h \cdot T_c$ is too small, then catastrophic collisions remain a significant possibility. Conversely, with a value of $N_h \cdot T_c$ large enough

and with well designed codes, the multiple access interference can be modeled as a Gaussian random process. When the time hopping is applied to the equation (1-2) it becomes:

$$s^{(k)}(t) = \sum_{j=-\infty}^{+\infty} w\left(t - jT_f - c_j^{(k)} T_c\right). \tag{1-3}$$

Because the hopping code is periodic with period N_p the expression (1-3) is periodic with period:

$$T_p = N_p \cdot T_f$$

i.e. one effect of the time hopping code is to reduce the power spectral density from the line spectral density ($1/T_f$ apart) of the uniformly spaced pulse train down to a spectral density with finer line spacing $1/T_p$ apart. Since the total power is not altered by shifts in time and the lines in the power spectrum have been multiplied in number (they are closer each other) each line has a lower amplitude which implies a lower spectrum density. An interesting approach to the spectrum analysis of the signal can be found in [4].

Data Modulation: as for data modulation the analysis is restricted to binary transmissions. Let the data sequence $\{d_i^{(k)}\}$ be associated to the k^{th} transmitter and with values in the set $[0, 1]$. In order to improve the reception process gain, these information bits are redundantly transmitted N_s times at the cost of reducing the link transmission speed.

The greater is the number N_s the lower is the link speed but the lower the power spectral density can be because the total power received is the summation of N_s pulses. From this it comes out that the parameter N_s is an important project's choice. It is augmented to improve the BER or to obtain a more covert signal or even to slow down transmission. In analytical formulas this redundancy can be shown imposing a slower rate of changes to the data source. This is done inserting the data modulation in the signal with the notation $\{d^{(k)}_{\lfloor j/N_s \rfloor}\}$ where the operator $\lfloor ... \rfloor$ is the lower integer part of ... and where it is assumed that a new data symbol begins with pulse index $j = 0$. The information bit imposes a further shift in time to N_s successive pulses. This time shift is another parameter in our system, it is called δ, it is about $1/4$ of the T_c, and it is applied only in the case in which the data symbol is 1 (true value). If the symbol is 0 (false value) no further shift must be operated (we can visualize this operation in figure 1.2). Data modulation further smoothes the power spectral density of the transmitted signal (once again a good reference is [4]).

Since a single symbol has a duration $T_s = N_s \cdot T_f$, if the frame time T_f is fixed the *binary symbol rate* R_s is obtained through the equation:

$$R_s = \frac{1}{T_s} = \frac{1}{N_s T_f} \text{ sec}^{-1}$$

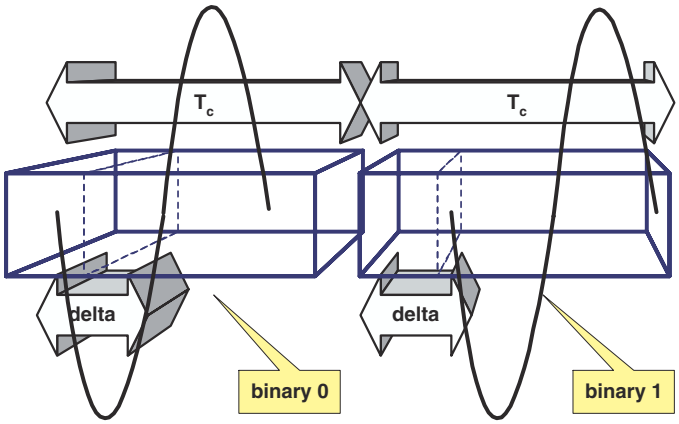

Figure 1.2: Pulse Position data modulation.

As a final result the UWB-IR signal transmitted by the generic user (or link) k who wants to communicate the sequence $\{d_j^{(k)}\}$ is:

$$s_d^{(k)}(t) = \sum_{j=-\infty}^{+\infty} w\left(t - jT_f - c_j^{(k)}T_c - \delta d_{\lfloor j/N_s \rfloor}^{(k)}\right). \tag{1-4}$$

All the parameters involved in it are summarized as follows:

- T_f frame time or pulse repetition time;
- T_c chip time;
- N_h number of hops;
- N_p code period;
- $c_j^{(k)}$ pseudorandom code symbol referring to the k^{th} link and to the j^{th} symbol;
- δ true bit further shift;

- $d^{(k)}_{\lfloor j/N_s \rfloor}$ data symbol associated to the k^{th} link (or k^{th} source) with values in $[0, 1]$;
- N_s data modulation redundancy;
- $R_s = 1/(N_s T_f)$ binary symbol rate.

Some observations are worth greater attention. No strict numeric relation is forced between the parameters N_h and N_p nor it is imposed between both of them and the parameter N_s. They can vary independently, but this cannot be guaranteed if particular project choices are taken, that can impose some constrains. For example, we will see that the choice of some spreading codes can give rise to a numeric relationship between N_h and N_p. Moreover, great values for N_h can reduce the duty cycle of the signal (incrementing T_f if T_c is fixed) thus imposing a reduced value for N_s in order to keep good bit rates R_s. The choice of T_c is technology dependent (clock hardware availability is required) and can be a precise constrain from which to start in order to dimension further parameters. In the table 1.1 (taken from [3]) some values are illustrated, which can be applied to the above mentioned parameters.

1.3 Receiver's signal processing

This section introduces the basic structure of the receiver signal processor as described in [3]. In the transmission event, the information generated by

Table 1.1: Numerical Example of Impulse Radio Basic Operating Parameters

Name	Notation	Typical Values
Impulse width parameter	τ_m	$0.2877ns$
Impulse width	T_m	$0.7ns$
Impulse repetition frequency	T_f^{-1}	10MHz
Frame Time	T_f	$100ns$
Transmission rate	R_s	$19.2Kbps$
Symbol Time	$T_s = R_s^{-1}$	$52.083\mu s$
Number of impulses per symbol	$T_s T_f^{-1}$	520.83

a binary source which is represented with the expression $\{D_i^{(k)}(u)\}_{i=-\infty}^{+\infty}$ is associated to a PPM signal which suffers from a shift δ whenever $D_j^{(k)}(u)$ is a binary 1. The letter u introduced in formulas represents a random variable which explicits the stochastic nature of the source. When redundancy is introduced in the transmitted signal the relationship between the data modulation $d_j^{(k)}$ and the k^{th} source $D^{(k)}(u)$ is expressed by:

$$d_j^{(k)}(u) = D_{\lfloor j/N_s \rfloor}^{(k)}(u).$$

The reception is thus performed thanks to a binary hypotheses revelator. The receiver must decide whether $D_i^{(k)}(u)$ is 0 or 1 based on the observation of the received signal $r(u,t)$ in a time interval of duration $T_s = N_s T_f$. This corresponds to deciding between two hypotheses $\mathcal{H}_0$ and $\mathcal{H}_1$ that can be

expressed as follows:

$$\mathcal{H}_d : r(u,t) = A_k \sum_{j=0}^{N_s-1} w_{rec}\left(t - \tau_k - jT_f - c_j^{(k)}T_c - \delta d\right) + n_{tot}(u,t)$$

in which:

- τ_k represents a possible asynchronism between the receiver clock and the one driving the transmitter k;
- d represents the actual (indexed i in the sequence) value of the source variable $D_i^{(k)}(u)$ and must be in the set $[0,1]$;
- $n_{tot}(u,t)$ sums all the disturbing effects operated by different transmitters interferences (in the population of N_u individuals) and hardware noises, i.e.:

$$n_{tot}(u,t) = \underbrace{\sum_{l=1,l\neq k}^{N_u} A_l s_{rec}^{(l)}(u, t-\tau_l(u))}_{\text{multiple access}} + \underbrace{n(u,t)}_{\text{receiver}}$$

Both hypotheses are explicited as follows:

$$\mathcal{H}_0 : r(u,t) = A_k \sum_{j=0}^{N_s-1} w_{rec}\left(t - \tau_k - jT_f - c_j^{(k)}T_c\right) + n_{tot}(u,t)$$

$$\mathcal{H}_1 : r(u,t) = A_k \sum_{j=0}^{N_s-1} w_{rec}\left(t - \tau_k - jT_f - c_j^{(k)}T_c - \delta\right) + n_{tot}(u,t)$$

As to the particular structure of the receiver, if no other users were present, and if the data $\{D_i^{(k)}(u)\}$ was composed of independent random

variables, then the optimum receiver would be the correlation receiver presented in [5], [6], [7] and [1] which can be reduced to the formula:

$$\text{decide } D_0^{(k)} = 0 \Leftrightarrow \underbrace{\sum_{j=0}^{N_s-1} \overbrace{\int_{\tau_k+jT_f}^{\tau_k+(j+1)T_f} r(u,t)v\left(t-\tau_k-jT_f-c_j^{(k)}T_c\right)dt}^{\text{pulse correlator output } \triangleq\ \alpha_j(u)}}_{\text{test statistic } \triangleq\ \alpha(u)} > 0$$

if applied to the case of the source symbol indexed 0. In the last expression the correlating shape $v(t)$ coincides with:

$$v(t) = w_{rec}(t) - w_{rec}(t-\delta)$$

where δ is chosen as the optimum value indicated in [1]. All this can be otherwise interpreted by using a nomenclature which underlines the common contribute brought about by different pulses redundantly transmitted. Under the assumption that the receiver might be aligned to the clock driving the transmitter, i.e., there is an exact knowledge of the parameter τ_k (at least τ_k mod N_pT_f to allow scrambling code alignment), it is possible to determine a sequence $\{\mathcal{T}_i\}$ of time intervals, with interval $\mathcal{T}_i$ containing the waveform representing data bit $D_i^{(k)}$, i.e.:

$$w_{bit}(t) = \sum_{j=iN_s}^{(i+1)N_s-1} w_{rec}\left(t-jT_f-c_j^{(k)}T_c-\tau_k\right).$$

Inside each interval it is possible to apply the same approach followed in the extensive analysis above with

$$v_{bit} = w_{bit}(t) - w_{bit}(t-\delta) = \sum_{j=iN_s}^{(i+1)N_s-1} v\left(t-jT_f-c_j^{(k)}T_c-\tau_k\right)$$

$$v(t) = w_{rec}(t) - w_{rec}(t - \delta)$$

and the corresponding decision becomes

$$\text{decide } \mathcal{H}_0 \;\leftrightarrow\; \underbrace{\int_{t\in\mathcal{T}_i} r(t)v_{bit}(t)dt}_{\alpha} > 0. \tag{1-5}$$

When more than one link is communicating the optimal processor for receiving the desired signal is not of the form illustrated in 1-5, but it is a complicated processing structure that takes advantage of all of the receiver's knowledge concerning the form of the interfering signals. In the next we will retain the easily implemented decision procedure of 1-5 even when many transmitters are active.

2

Link Acquisition

In this chapter it is considered the most general landscape which can take place in wireless networks, i.e. a fully distributed wireless network. This is done in order to introduce some problematics which come out from such a flexible structure. In the successive chapters, instead, the network will be

limited to an hierarchical structure in which there is a precise distinction between Mobile Terminals (MTs) and Access Points (APs). The simplified approach of assuming a certain level of function centralization in the APs will allow to find easily applicable solutions to those problematics inherited from the wireless environment. The problems illustrated here are very interesting and very difficult to be fronted. At the beginning, the network is even considered to possess directive antennas and only at a later stage omni-directional antennas will be adopted thus cutting away the great deal of problematics illustrated in the next section. The approach which follows is rather didactic and only in the next chapters, with the simplifications announced, it will become a real project. In a mobile environment when a single terminal is turned on for the first time it does not know anything about its surroundings. It lacks fundamental informations such as:

- its absolute position;
- its neighbors' absolute positions (from which to derive the relative distances);
- the clocks' period of other terminals (including the APs if they are foreseen).

All these informations must be collected before any communication can take place and the *link acquisition* is responsible for their acquisition. The necessity for the knowledge of the spatial distribution of different neighbors comes

out from the fully distributed nature which implies that each terminal has to be responsible for the routing of at least a part of the communications performed in its area of localization. As it is underlined in [8] directive antennas bring about different advantages such as an improvement in the network's capacity to resist to jamming/interfering activities, or a better behavior in cross-channel interference. The drawback implied by this choice is the necessity to get communicating equipments perfectly aligned in their beams' directions. The link activation with directive antennas comprises two main functions:

- spatial synchronization;
- temporal synchronization.

These points are going to be analysed by supposing that each terminal is capable of acquiring its own absolute position in space. This is a problem that will be the objective of further studies. In [8], for example, all proceeds considering the Global Positioning System (GPS) as the actual solution, but it is clear that a system based on UWB signals can exploit a better performance in positioning, with a resolution that GPS can never reach. Lets underline a further requirement: the intended communication is bidirectional and both kinds of synchronization have to be performed simultaneously. A link is acquired when round trip is guaranteed.

2.1 Spatial Synchronization

In [8] it is developed the problem of *spatial synchronization* of a pair of equipments in a Line of Sight (LOS) condition. The environment hosting them is contaminated by the presence of an unfriendly interceptor (UI) and the objective is to reach spatial synchronization minimizing the probability that UI could gain control of the informations exchanged. This point of view is rather distant from ours but its introduction is really interesting if it is observed that a low probability of interception can be seen as a capability to produce a signal of minimum impact in the radio systems already developed in the location we are operating. If an UI finds difficulties in scanning the transmitted signal, very realistically the disturbs and interferences procured to other radio systems (for example narrow bandwidth systems) can be considered irrelevant. By being sensible to this problematic the coexistence between the traditional communication systems and the UWB wireless networks becomes feasible. This is the reason for dealing with the implementation of efficient algorithms which allow to reach the spatial synchronization.

2.1.1 Spatial alignment of a pair TX RX.

When a MT must find a neighbor terminal, it has to scan all the area surrounding itself. To exploit this search, the terminal equipment has the necessity to "look at" all the directions in which the presence of its partner is at

least expected. Without a particular knowledge the system must accomplish a research at 360°. All the space around is sampled into M sectors with an angular amplitude determined by the apertures of the antennas. To perform the operations, the time available for transmitting/receiving has to be divided in M time slots each of them dedicated to a particular sector angle. While looking at a certain direction, the MT must have the possibility to complete the following operations:

1. - transmit a discover signal;
   - receive and interpret a discover signal;
2. - receive and interpret a discover signal response;
   - transmit a discover signal response.

To exploit these operations the M^{th} time slot is further divided in two sub-slots which perform respectively the operations in (1) and in (2). These sub-slots are further divided to exploit separately the transmission and reception operations. Given this structure, the interest here is in identifying a procedure useful to find a communication partner. This is done through the introduction of two algorithms which have been first illustrated in [8]. Both algorithms suggest the order in which to perform operations possibly taking advantage of an "a priori" knowledge of the RX and UI previously occupied positions together with their mobility models. In both cases the following nomenclature is adopted:

- two hypotheses are considered, the presence of the signal is indicated with $\mathcal{H}_1$ while $\mathcal{H}_0$ stands for signal absence;
- the event of mono-directional link success at the k^{th} trial is indicated with $\mathcal{L}_{Fk}$ for the forward direction while $\mathcal{L}_{Rk}$ is associated to the reverse direction;
- the event of bi-directional link success at the k^{th} trial, i.e. link acquisition, is indicated with $\mathcal{L}_k = \mathcal{L}_{Fk} \bigcap \mathcal{L}_{Rk}$;
- the event of link unsuccess at the k^{th} trial is indicated with the complementary notations $\mathcal{L}^c_{Fk}, \mathcal{L}^c_{Rk}$ respectively for the forward and reverse direction;
- the event of link unsuccess for the first $k-1$ trials is indicated with $\mathcal{F}_{k-1} \triangleq \mathcal{L}^c_1 \bigcap \mathcal{L}^c_2 \cdots \bigcap \mathcal{L}^c_{k-1}$;
- the event of intercept is indicated with $\mathcal{I}_k$;
- the minimum SNR for which the link is consider acquired is indicated with γ_0.

To both algorithms it is applicable the general scheme which follows:

1. Forward Link (FL):
    - A TX uses a predetermined transmission power to send initiating signals to its intended RX in the first sub-slot of a time slot for

the k^{th} trial.

- Upon receiving the FL signals the RX decides if the FL is successful by performing the following test:

$$\mathcal{L}_{Fk} \triangleq \{y_{Fk} \geq T_F | \mathcal{H}_1\}$$

where T_F is the threshold such that the received SNR is at least γ_0.

2. Reverse Link (RL):

- After deciding the FL is successful and decoding the attached data, the RX adjusts the transmission power to reply the TX in the second time sub-slot such that the TX can receive the RL signals with the SNR achieving minimum threshold. Upon receiving the answer from RX, TX decides if the RL is successful by performing the following test:

$$\mathcal{L}_{Rk} \triangleq \{y_{Rk} \geq T_R\}$$

where T_R is defined to satisfy a preset system false alarm rate P_{FA}, i.e. $P_{FA} = Pr(y_{Rk} \geq T_R | \mathcal{L}_{Fk}^c)$.

- After deciding that the FL is unsuccessful, the RX remains silent and waits for the next initiating signals from TX.

3. Next Search (NS):

- If the RL is successful then the communication link is established and the link acquisition is completed ($\mathcal{L}_k$).
- If the RL is unsuccessful or no answer has been received in the second time sub-slot, the TX uses the NS power to search for the RX in the next available time slot.

The algorithms introduced in [8] are as follows:

- Sequential Ascending Search Power Algorithm (SASPA): making use of "a priori" informations, the first angular sector is chosen to be the one much probably hosting the RX equipment. Once the sector is decided, we select a set of power levels that will be adopted in each of the N trials defined for the procedure. Each successive trial in the same direction will be performed (in case of no response) with an ascending power level until a maximum at the N^{th} try is reached. From that point the actual sector is declared clean and we move to the second most probable direction. Each direction is assigned a set of power levels which can be specific and calculated with the aid of "a priori" informations. Lets have a look at the algorithm itself:
  - Initial values: $j = 0$ and $n = 1$.
  - Iteration:
    1. Search directional sector m_j using power level $P_T(D_{m_j,n})$.

2. If the link succeeds, then stop. If the link does not succeed, then let $n = n + 1$; and
   * if $n < N$, then go to 1.
   * if $n = N$ and $j < M - 1$, then set $j = j + 1$ and go to 1.
   * if $n = N$ and $j = M - 1$, then stop.

For this algorithm to be performed it must be defined the M dimensional vector indicating the order in which to search a sector, together with M vectors of dimension N which must indicate the power levels that must be used sequentially for each sector. If no distinction is made between the angular sectors it suffices a single vector of dimension N.

- Sequential Bayesian Search Algorithm (SBSA): the transmission region centered at the TX is finely discretized into M directional bins (with the same angle aperture, $2\pi/M$) and N range bins (with the same range bin width D_{max}/N). These $M \times N$ cells indexed by the elements in $\Omega = \{(m, n)\ m \in \{0, \cdots, M-1\}, n \in \{0, \cdots, N-1\}\}$ are discretized in such a way that the probability of link acquisition by a RX (or the probability of intercept by a UI located at any point within a cell varies negligibly. The events $\{V_R = (m, n)\}$ and $\{V_I = (i, j)\}$ indicate that the RX and the UI are located in cell $(m, n) \in \Omega$ and $(i, j) \in \Omega$. Define $q_R(m, n, k)$ as the "a posteriori" probability that the RX is located at cell (m, n) after the k^{th} trial, given all previous reverse link tests

variables $\mathbf{y_k} \triangleq (y_{Rk}, y_{Rk-1}, \cdots, y_{R1})$ and all previous search powers indices used by the TX $\mathbf{T_k} \triangleq (T_1, T_2, \cdots, T_k)$ where $T_k \triangleq T(m_{Tk}, n_{Tk})$ and $(m_{Tk}, n_{Tk}) \in \Omega$, i.e.

$$q_R(m, n, k) \triangleq Pr\left(V_R = (m, n) \,|\mathbf{y_k}, \mathbf{T_k}\right)$$

The discretized transmission power index $T(m_{Tk}, n_{Tk})$ for the k^{th} trial means that the TX is pointed at directional bin m_{Tk} with its transmission power level $P_T\left(\frac{D_{max} \cdot n_{Tk}}{N}\right)$. With the SBSA algorithm the TX chooses the search direction and the transmission power level that maximize an objective function which is the conditional probability of link acquisition without intercept by a UI for the k^{th} trial, given $\mathbf{y_k}$ and $\mathbf{T_y}$.

- Initial values: $k = 1, q_R(m, n, 0)$ and $q_I(m, n)$ are given.
- Iteration:
  1. The TX chooses $T^*(m_{Tk}, n_{Tk})$ to search for the RX by computing

     $$T^*(m_{Tk}, n_{Tk}) \text{ t.c. } \max_{T(m_{Tk}, n_{Tk})} Pr(\mathcal{L}_k \bigcap {\mathcal{I}^c}_k | \mathcal{F}_{k-1}, \tilde{y}_{k-1}, \tilde{\mathcal{T}}_k)$$

     from $q_R(m, n, k-1)$ and $q_I(m, n)$.
  2. If the link is successful, then stop.

     If the link is unsuccessful, then
     - the TX computes $q_R(m, n, k)$ from $T^*(m_{Tk}, n_{Tk})$ and y_{Rk} using Bayes' rule (see [9]);

- $k = k + 1$ and go to 1.

The computation of the "a posteriori" probability of the RX's location given y_{Rk} is obtained using the Bayes'rule (see [9], [8]). Thanks to its way of operating, SBSA assures a greater probability of obtaining link acquisition avoiding interference but this is done at the price of a computing complexity sometimes not sustainable. It must even be said, that if the "a priori" knowledge is not close to realty the results of SBSA can be really faulty. If the guessed probability density poses a zero in the place where the RX really stands, the algorithm will never search that direction failing any link activation. These requirements (precise knowledge and hardware complexity) end by making the SASPA algorithm more adapt to the scope.

2.2 Temporal Synchronization

Since in the time hopping UWB signal the information is strictly associated to the shift of the pulses from their nominal position (PPM), it is clear the importance of synchronization as a necessary step towards the reception of a message. Since the T_c width is usually less than a nanosecond and the informative δ shift equals a quarter of that width, the precision required must be of the order of sub-nanoseconds time intervals. When in the project of a wireless network some centralized APs are included, they usually accomplish the synchronization function by the use of beam signals (see chapter 3) capable

of giving the right timing to the listening MTs. In a fully distributed network the problem of synchronization becomes really tough to be solved because there is a number of MTs each possessing a clock phase misalignment with all its neighbors and for this reason incapable to establish a communication with other partners. It is clear that for each communication to take place it is necessary to exploit a *link activation protocol* capable of synchronizing two MTs to a reference clock which can be, for example, the clock belonging to the initiating partner. Each communication can take place through an information exchange where a *master* (the terminal possessing the reference clock) interacts with a *slave* (the terminal which aligns its clock to the master's clock) as follows:

- a bidirectional channel can be mastered by the same clock in both directions, i.e. there is a unique master to which a slave must synchronize both in the up-link and in the down-link communications. When there is at least one AP it becomes the master of all the bidirectional communications with the MTs.

- a bidirectional channel can be mastered by two different clocks in the opposite directions, i.e. the transmitting MT is always the master in the direction TX$\longrightarrow$RX and a bidirectional channel has always the

following structure:

$$\begin{array}{ccc} master & & slave \\ TX & \longrightarrow & RX \\ RX & \longleftarrow & TX \\ slave & & master \end{array}$$

In particular, in chapter 3 the case of an AP mastering all the MTs in its range will be studied taking in great care the point of view of the slave terminal which wants to grasp the master's clock. This process will be referred to as *down-link acquisition* and as a result it will produce the synchronization of the slave MT to the master AP together with other useful informations which will be explained later. In the next section it is introduce the reception problem as it is developed in [10].

2.2.1 The Reception Problem

The UWB systems allow multiple user to share the same resource which coincides with the frequency band occupied uniformly by each transmitted signal. As told in chapter 1 different signals are multiplexed through the use of distinct time hopping codes which distinguish different physical channels. When a great number of transmissions coexist on the same resource, the signal that a MT (or an AP) receives from the environment is the sum of equally contributing transmissions that can in part collide with each other.

To isolate a precise physical channel from the received signal it is necessary that the time hopping codes adopted are orthogonal with each other (this problem will be studied in chapter 4). When the orthogonality between the codes is guaranteed, the receiver can isolate the physical channel of interest by simply performing a correlation between the local reproduction of the code assigned to it and the observed signal. The correlator in the receiver can operate in two different ways:

- Time-Sliding Correlator (TSC): the analog signal received is passed through a series of delay stages. At each stage it is computed the product with the code sequence looked for. The maximum among the results obtained at the different stages indicates the time at which it is observed the much resembling input sequence. If its value trespass a given threshold, than the expected code has been found;

- Time-Integrating Correlator (TIC): the input signal is analysed by a parallel bank of integrators. Each integrator performs the correlation between the input, windowed at a precise time instant (which depends on the particular integrator) and the expected code sequence. This operation is visible in figure 2.1 where the transmission is exploited through the use of *doublets*, i.e. single pulses each followed in transmission by a pulse inverted in polarity (the reasons for this implementation choice are illustrated in [10] and are not being discussed here). In that

figure, the top row shows the received code sequence "110" encoded using impulse doublets, the second row shows the accumulation of charge in the n^{th} integrator, while the third and fourth rows represent the reference code sequences for two successive correlators with a chosen phase misalignment. Once again, the maximum output from the bank of integrators indicates the instant in which the much resembling sequence has been received. Figure 2.2 represents the output from the bank of integrators with the integration period equal to the impulse period and with phases overlapped by an half. The diagram illustrates how the outputs of 17 integrators are generated by different alignments of the reference code $\Phi_i(t)$ with the received signal $\Phi_e(t)$. The peak of the correlation in this case is very clear.

Some words are to be spent in confronting the two approaches. Suffice it to say that the Time-Sliding Correlator suffers from the major problem of delaying an analog signal. The devices capable of this operation are not suitable for the system aimed at and a previous Analog-to-Digital conversion to allow digital memories to operate is not contemplated. The main difference between the two approaches is in the point of view from which the delay operation is seen. In the former solution there is an analog signal that must be shifted against a static digital code sequence. In the latter, it happens the opposite operation in which a digital sequence is easily shifted in time so as

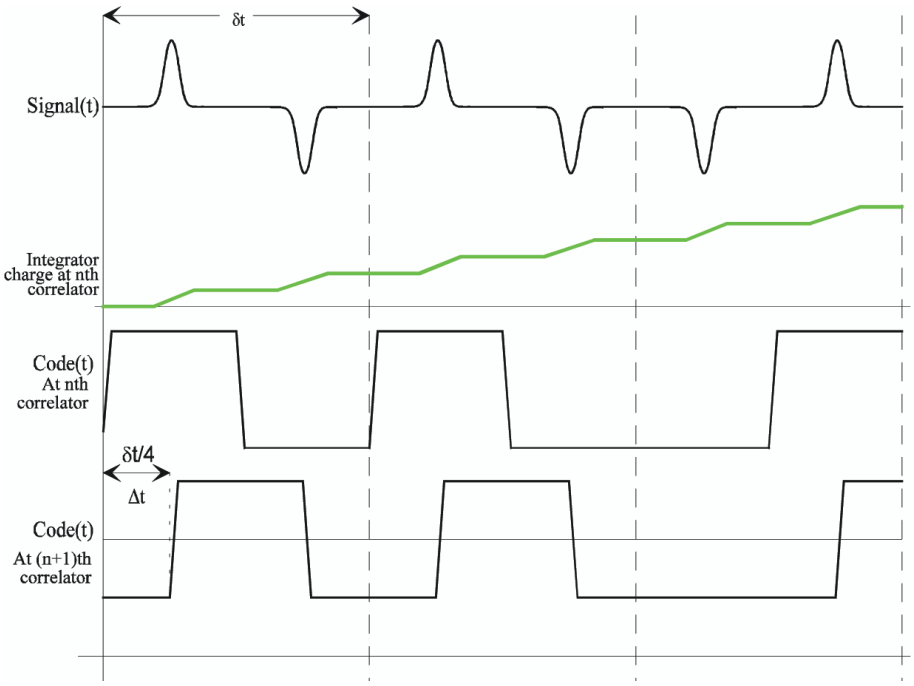

Figure 2.1: Time-Integrating Correlator.

to be confronted with the incoming analog signal. The price which must be paid to obtain this simplification is the necessity of having a single integrator for each temporal window which must be analysed. Practically speaking, the outputs from the series of integrators in TIC represent the sampling of the TSC response as a function of time. Following this interpretation, the correct behavior of the TIC can be proved through the application of the sampling theorem. In the next section some considerations about the TIC are going to be introduced in order to illustrate some implementation dependent features which allow to adapt that kind of correlator to different personal requirements.

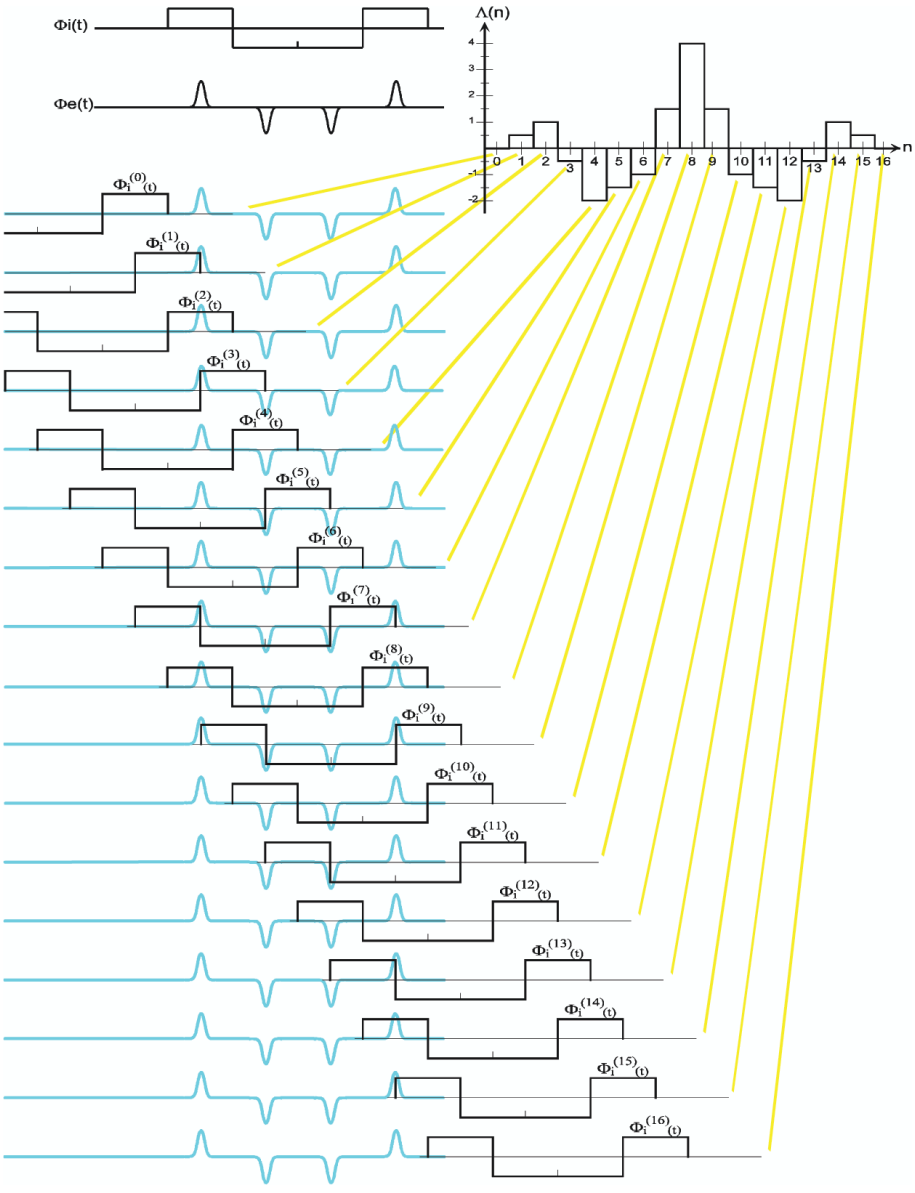

Figure 2.2: Time-Integrating Correlator outputs.

Time-Integrating Correlator and Time Domain Filtering

The rate of the period between two successive pulses in which an integrator effectively operates, can be chosen taking in mind the following drawbacks:

- When the integration period is equal to the total time available between pulses, it follows that no phase at all can position them in a time settle outside the considered time window. If this should happen, the result of the correlation would be erroneously null.

- As a consequence of this conservative choice, it is not possible to take advantage from the *time domain filtering*. In practice the received signal is always added to noise or disturbs. When these interferences collide in time with the period of activity of the transmitted signal, there are no simple means to front them. When the disturbs operates in a silence period of the message source it is sufficient to turn off the receiver in order to improve its performances. This selective choice of the period of activity of the receiver is called time domain filtering. For the time domain filtering to reach efficiency, it is necessary that the synchronization is reached with a good level of precision. If this is the case, the shorter the period of integration is, the better is the signal to noise ratio obtained by the receiver because part of the noise is masked away from the observed signal. In figure 2.3 the time domain filtering operation is exploited by supposing that synchronization

is already achieved. In the first row it is shown the received signal affected by noise, the second row shows the charge accumulation in the n^{th} integrator and the third and fourth rows describe respectively the reference code sequence in the n^{th} and $(n+1)^{th}$ integrator. Thanks to the synchronization, when the integrator is turned off (see the third row) part of the noise is clearly masked out (see the second row).

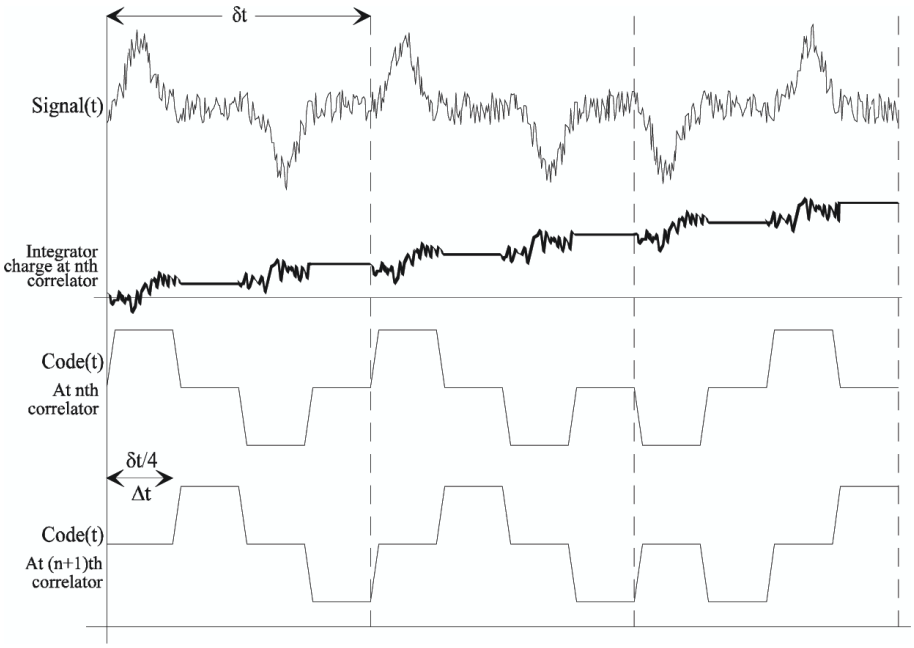

Figure 2.3: Time domain filtering.

Complex TIC patterns to improve time sensitivity

The integration period which influences the efficiency of the time domain filtering is only a part of a wider implementation parameter which is known as *TIC pattern*. A TIC pattern includes the following elements:

- the *time integration period*;
- the *phase overlapping*.

These features can be modified in order to improve the sensitivity of the receiver to time shifts, i.e. phase misalignments. From previous studies, it results that the best behavior (in time shift sensitivity) is obtained when the integration period is equal to the period of repetition of the pulses, and when the phases of the correlators overlap for half that width. This situation is visible in figure 2.1 where the transmission is exploited through the use of doublets, in particular in the third and fourth rows it is visible the overlap of the integration periods of the n^{th} and $(n+1)^{th}$ integrators.

The relation between the TIC and the TSC is underlined in figure 2.4 where the continuous time cross-correlation function $\Lambda(t)$ (left side) is opposed to the results produced by a Time-Integrating Correlator $\Lambda(n)$ (right side). The TIC output clearly coincides with the sampling of the TSC with a period equal to the distance between the phases (in the figure the sampling times are indicated by Dirac functions) which in this particular case ($125\ ps$) produces the optimum condition mentioned afore. Each rectangle on the right side, is the output from a single integrator. The integrators which manifest the greatest sensitivity to time shifts are those coinciding with time instants which rapidly change in the TSC output. For the particular expected code sequence displayed in figure 2.4, the most sensitive integrators

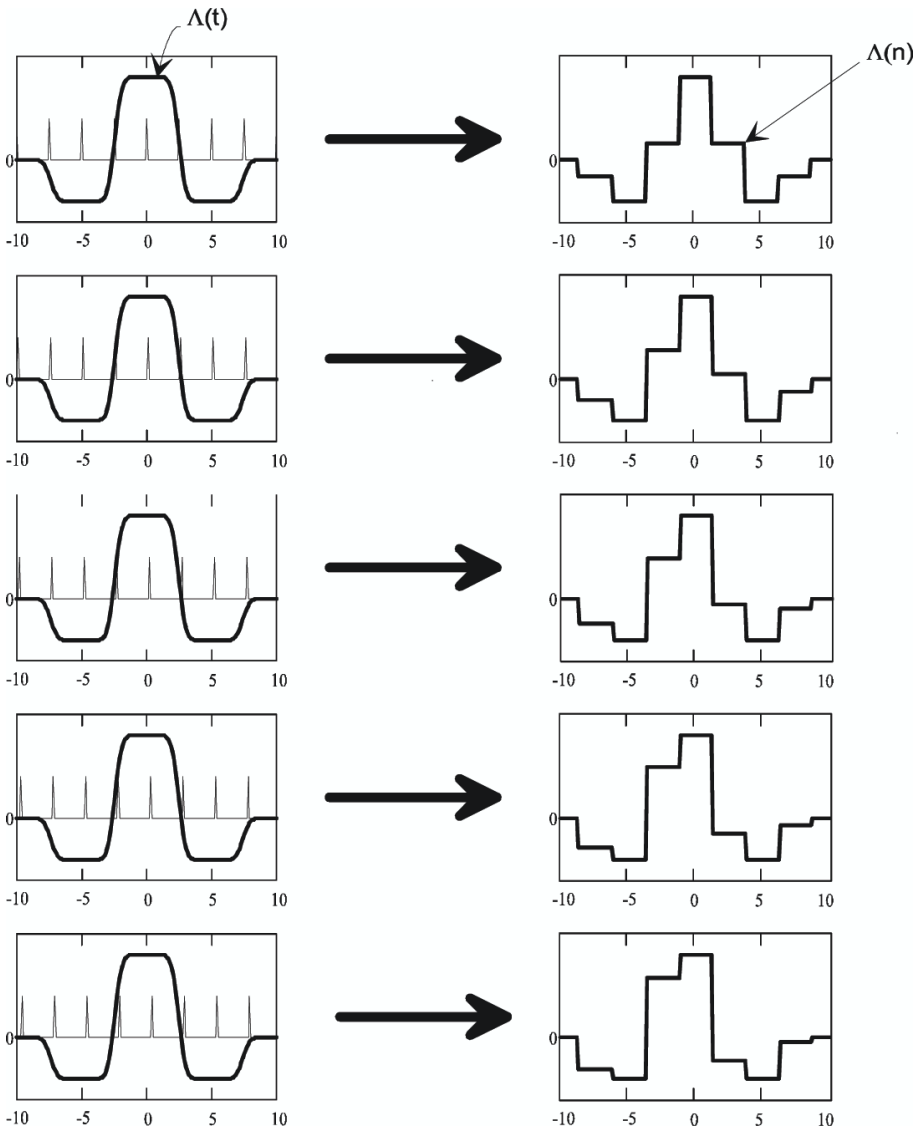

Figure 2.4: Correlation Sampling.

are those associated with odd samples. It is interesting to observe the figure 2.5 where the output of the TIC is shown for successive relative shifts ($\frac{1}{8}$-phase) between the received signal and the reference code. The phase misalignment between the integrators on the bank is of 2.5 ns (which implies that $\frac{1}{8}$-phase= 312 ps) and it is important to note that a time shift of 2.5 ns produces the same pattern in the lower right graph as in the upper left graph, except that it is centered in bin 9 instead of bin 8. It is clear here that the odd integrators are those which undergo the greatest variation in their outputs as a consequence of the shift of the received signal. While the peak value can be confronted with a threshold to decide for the presence or for the absence of the searched code sequence, the odd samples can be used to perform a further tracking in time among the local reproduction of the code sequence and the one received.

2.2.2 Simple Synchronization Protocol Implementation.

The one that follows is a really simple example of a synchronization protocol which can be applied to a fully distributed network (otherwise referred to as a peer-to-peer network) where a bidirectional communication between two MTs is mastered by two different clocks in distinct directions (see section 2.2). The case in which a unique clock is adopted (i.e. an AP is present) will be introduced in chapter 3 and is not of interest here. In a peer-to-peer network

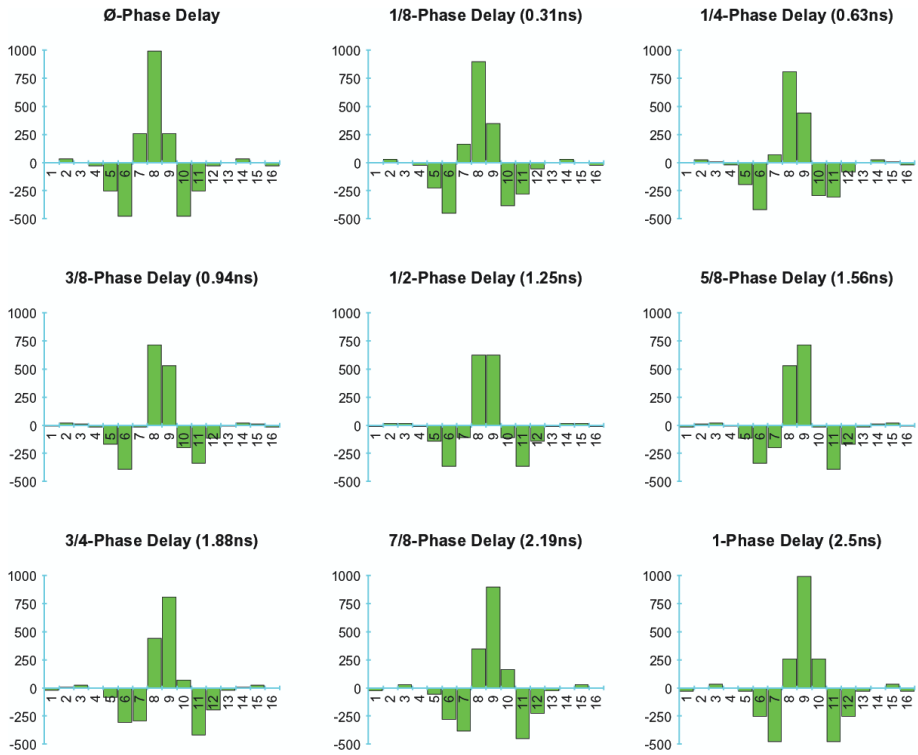

Figure 2.5: Correlation Histogram.

when a communication is going to be established there are two main actors:

- a Call Initiating (CI) Terminal Equipment which requests a communication and as a consequence of it starts the Initiating procedure;
- a Call Terminating (CT) Terminal Equipment which receives the request transmitted by the CI and operates consequently.

Both roles can be accomplished alternatively by each terminal equipment which in the same communication behaves as a master in the up-load direction (when it requests a link, CI, i.e. it polls an up-load link) and as a slave in the down-load direction (when it receives a link request, CT). As an example

suppose that the terminal A is going to activate a link (i.e a bidirectional channel) with terminal B. The device A needs to know the timing produced by clock B. Two realistic hypotheses are advanced here:

- Even if different clocks suffer from time drifting on the long period, they can be considered sufficiently consistent in the short period so as to allow single word reception;
- Propagation delays during the time of the activation procedure are not changing relevantly.

The CI partner, i.e. A at this stage, sends a sequence of words intervalled by a fixed time period. These words are encoded using a reference sequence which is proper of the Link Request Channel referring to terminal B (which at this stage is the CT). Thanks to the short period consistency, B, which knows the link request channel code sequence, is capable of acquiring the sequence of words transmitted by A and, successively, it is able to compute the period of repetition and to confront it to the measure operated by its own clock. After that B can calculate the ratio between the observed frequency of the clock A and the actual frequency of its own clock. At this point, terminal B can synchronize itself with clock A and its down-load link can be activated. Since a bidirectional communication is required, if B decides to accept the communication, it must request an up-load link to the terminal A and the same procedure illustrated so far is activated with an exchange

in roles, i.e. B becomes the master behaving as a CI while A becomes the slave behaving as a CT. This time the transmitted periodic sequence of words is encoded using a reference sequence proper of the Link Request Channel referring to terminal A. Since the transmission can be done in a temporal slot immediately following the one of in which the request from A has been received, A knows approximately when it has to expect an answer and can perform an exhaustive search in that time window. If the answer is immediately sent, A is even capable of evaluating the actual round trip time. When the activation protocol is completed both terminals possess a bidirectional link. This procedure can be easily modified to count for the presence of centralized masters (i.e. APs). This is done in chapter 3 where only the APs' down link acquisition is developed leaving the completion of the bidirectional link acquisition as an objective for future interests.

Part II

Access Points' down-link acquisition

3

Down-Link Acquisition

It is important to introduce a brief description of the adopted receiver. With the hypothesis that the receiving equipment (RXe) knows the code (*spreading sequence*) which vehicles the information it needs, the reception procedure takes place by operating the correlation between the local reproduction of the

known code (*de-spreading sequence*) and the signal received from the environment. The output of the correlator will show some peaks coinciding with the position in time of the searched sequence. If the temporal synchronization with the transmitting equipment (TXe) has been already reached, it is possible to measure the relative shifts of these peaks from expected positions in time. These relative shifts are due to the data modulation that has been introduced in 1.2 (i.e. PPM) and the RXe can guess the transmitted bit by observing the presence or the absence of a δ shift.

The reception of a known time hopping code, operated by such a correlator on a signal disturbed by noise and interference, is known as *code synchronization*. Code synchronization is obtained through:

1. *code acquisition* which is responsible for a coarse time alignment (within a T_c) [11];

2. *code tracking* which guarantees a finer alignment in time and is implemented by *tracking loops* [11].

Its objective is to align the *de-spreading sequence* to the *spreading* one in order to allow pulse position de-modulation.

3.1 Down-Link Acquisition

The adopted network architecture is composed of Mobile Terminals (MTs) and asynchronous Access Points (APs). All the communications between the MTs and the APs are exploited by means of *physical channels (ph-CHs)*. Each ph-CH is meant to be associated with a unique time hopping code and can vehicle more than one *logical channels (l-CHs)*. The l-CHs are classified as Traffic CHannels (TCHs) if they support user data transmissions or Control CHannels (CCHs) if they convey control and maintenance information. Different l-CHs are multiplexed over one ph-CH as follows:
A ph-CH is organized in Logical frames (L-frames) of length T_{Lf}. Each L-frame is divided into M Logical Slots (L-slots) of length T_{Ls} with $M \cdot T_{Ls} = T_{Lf}$. A single l-CH occupies a sequence of not overlapping time intervals distributed in predetermined L-slots at known positions. Moreover, time intervals belonging to different l-CHs are never overlapping. In figure 3.1 three different l-CHs are multiplexed on a single ph-CH. The bit rates of these channels are directly proportional to the allocated time. The l-CHs named a and c occupy one time window in each L-slot (in particular c has the time windows belonging to even L-slots greater than a) while the l-CH named b occupies only odd L-slots. For what said, a allows medium bit rates, b allows slow bit rates while c allows fast bit rates.

In general, the information transmitted in a single l-CH, or even in a

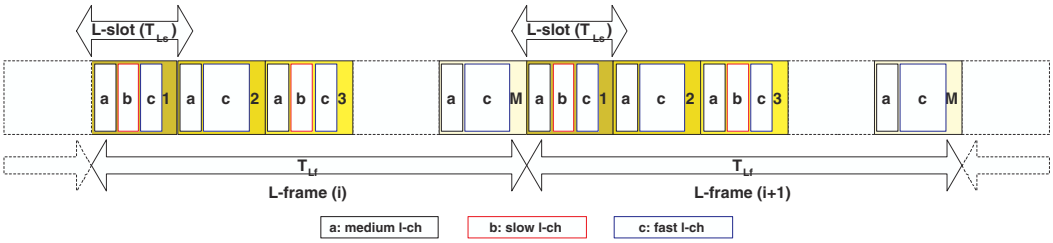

Figure 3.1: Three l-CHs multiplexed on a framed ph-CH.

single ph-CH, apart from changing within the interval T_{Lf}, changes from an L-frame to the next one. There exist some exceptions in which the same information is transmitted with a period T_{Lf}. This is the case of the *Broadcast CHannel (BCH)*. Each AP must possess a ph-CH called BCH in which it periodically transmits all the information needed by the MTs for establishing a connection. The time hopping code associated to the BCH is chosen in a finite set of time hopping codes called *BCHs' codebook*. Each entry in this codebook is indexed by a distinct *identity* and each AP is assigned a BCH-id which is the identity in the BCHs' codebook of the time hopping code associated to its BCH. Different APs cannot share the same BCH-id and consequently the same BCH.

To establish a communication between an MT and the AP it is necessary to acquire the AP's BCH. This process is called *down-link acquisition* and is achieved in two steps:

1. *Temporal synchronization with the AP*, further divided in

   - *L-slot synchronization*, i.e. the search for the beginning of each

L-slot in the L-frame;

- *L-frame synchronization*, i.e. the search for the beginning of each L-frame.

2. *BCH-id selection*, i.e. the search for the assigned BCH-id.

Temporal synchronization is a necessary step for receiving a signal since information is conveyed in the relative position of the pulses from nominal time settles and a precise knowledge of the reference clock is mandatory. Different clocks in different equipments usually present phase misalignments which compromise the intelligibility of the communication. The MTs know "a priori" the T_{Lf}, the T_{Ls}, the T_{SC} and the number M of L-slots in an L-frame as well as the PPM parameters introduced in 1.2. With these hypothesis the temporal synchronization with the AP is reached when the first L-slot in a BCH's L-frame has been precisely isolated in time. From now on, the MT is synchronized with the AP's clock phase and can spot each L-slot in the received L-frames. Temporal synchronization differs from code synchronization. The latter is a low level operation exploited by the correlating receiver, while the former is an high level procedure performed by the MT which uses code synchronization as explained at page 60. Once the temporal synchronization (step 1) has been reached, and the BCH-id has been selected (step 2), the BCH (which is a ph-CH) can be looked up in the BCHs' codebook. Later, the MT starts listening to the CCHs (the class of

l-CHs introduced at the beginning of section 3.1) multiplexed on the BCH thus receiving the information necessary to access the network.

3.2 Cell search

A *cell* is the set of mobile equipments referring to a specific AP. This set can also be called a *MAC domain*. The down-link acquisition procedure considered in the paper is a specific and multi-functional process called *cell search* which accomplishes the steps 1 and 2 described at page 55 in a particular way. Rather than following the scheme reported at page 55, it implements the *two phases procedure* structured as follows:

cell search: phase 1

- Temporal synchronization with the AP: L-slot synchronization

cell search: phase 2

- Temporal synchronization with the AP: L-frame synchronization
- BCH-id selection.

The L-slot synchronization is considered as an independent objective while the L-frame synchronization and the BCH-id selection are merged together in a single phase. The cell search described in the paper takes inspiration from the works illustrated in [12] and [13]. To perform both phases it makes use of a resource shared among all the APs in the network, i.e. the Global

Synchronization CHannel (GSCH). The GCSH is classifiable nor as a ph-CH because it is not associated with a specific time hopping code, nor as a l-CH because it is not multiplexed on any ph-CH. It must be considered as a container for two distinct subchannels:

- the Primary Synchronization CHannel (PSCH): a ph-CH associated with a time hopping code called Primary Synchronization Code (PSC) which is the same for all the APs in the network and which is known "a priori" by any MT in the network;

- the Secondary Synchronization CHannel (SSCH): a particular ph-CH which can be associated to any time hopping code in a dedicated set α of time hopping codes called Secondary Synchronization Codes (SSCs), with $\alpha = [SSC_1, SSC_2, \cdots, SSC_q]$.

  In conclusion:

$$GSCH = PSCH + SSCH.$$

In the GSCH both the PSCH and the SSCH transmit periodically with period T_{Ls} in bursts of length $T_{SC} \ll T_{Ls}$. Between successive bursts their ph-CHs stay silent. PSCH and SSCH's bursts are perfectly superimposed and to avoid interferences the time hopping codes that support them (i.e. PSC and SSCs) must be orthogonal with each other. Each burst of the PSCH and of the SSCH is transmitted at the start of an L-slot of the BCH's L-frame (see

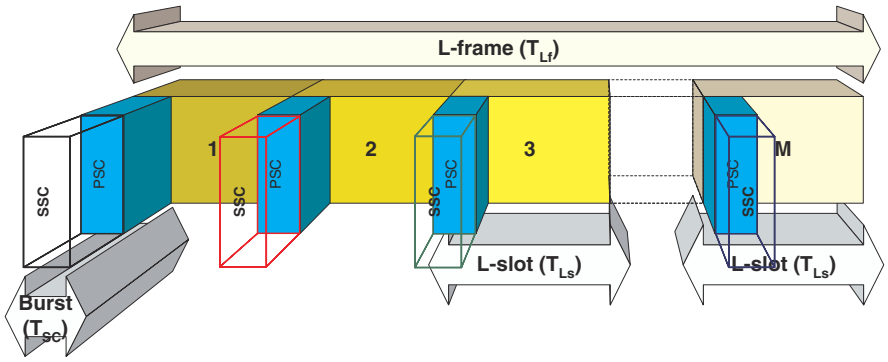

Figure 3.2: L-frame made up of M juxtaposed L-slots of length T_{Ls}. The SSCs are superimposed to a unique PSC.

Figure 3.3: PSC and SSCs' bursts are represented by the inner box containing a time hopping signal. The L-Slots coincide with the outer box.

figure 3.2 and figure 3.3).

While the PSC is the same for each burst and each AP, the SSCs can be different in successive bursts belonging to the same BCH's L-frame. It is mandatory that, once an SSCs sequence[1] $SSC^1 SSC^2 \cdots SSC^M$ has been chosen for a BCH's L-frame, it must be repeated in all the BCH's L-frames. The sequence of SSCs is the BCH-id which must be uniquely assigned to a specific AP (see figure 3.4).

[1]The superscript i refers to the i^{th} burst in the L-frame.

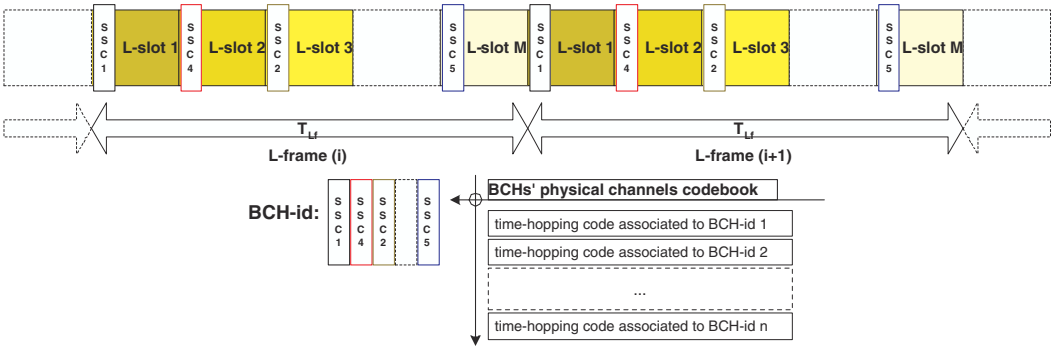

Figure 3.4: Two frames extracted from the periodic transmission of the AP. They belong to the BCH whose BCH-id is the one indicated.

The set of reserved SSCs sequences is called the *BCH-ids' codebook*. The BCH-ids' codebook adopted in the cell search is composed by Cyclically Permutable Codes (CPCs). A CPC is a block code of length n. Its codebook is a set of codewords such that no codeword is a cyclic shift of another and each codeword has n cyclic shifts. Each codeword of length $n = M$ has symbols in the alphabet $\alpha = [SSC_1, SSC_2, \cdots, SSC_q]$ and represents a BCH-id.

Thanks to CPCs, each BCH-id together with all its permutations is assigned to a different BCH. If the symbol SSC^1 is transmitted at the start of the first burst in the BCH's L-frame, the MT can reach L-frame synchronization by recognizing the exact permutation of the received BCH-id.

cell search: phase 1. The objective is to reach L-slot synchronization with the BCH. The MT knows the PSC which is transmitted at the start of each L-slot and turns on the receiver performing the code synchroniza-

tion described in 3 with the PSC locally reproduced. The output of the correlator shows periodic peaks at the beginning of each L-slot. When PSC code tracking is complete, the L-slot synchronization is reached.

cell search: phase 2. The objective is to reach L-frame synchronization and BCH-id selection. The L-slot synchronization obtained in phase 1 implies that the MT knows the time windows in which SSCH's bursts are transmitted. The MT starts searching in each time window for the transmitted SSC by performing code synchronization with each SSC in the set α till success. After analyzing M bursts, the MT has composed a sequence[2] $SSC^{(i+1)_M} SSC^{(i+2)_M} \cdots SSC^{(i+M)_M}$ which coincides with the i^{th} permutation of a generic $SSC^1 SSC^2 \cdots SSC^M$ sequence (which is a BCH-id). By recognizing the sequence $SSC^1 SSC^2 \cdots SSC^M$ the MT acquires the BCH-id (and the relative BCH), moreover, through the picking out of the order i of permutation it locates the start of the L-frame.

3.3 Cell search proposed algorithm

An algorithm that could be a suitable choice for cell search is presented in figure 3.5 and is described in this section. Remember that the MTs know "a priori" the T_{Lf}, the T_{Ls}, the T_{SC}, and the number M of L-slots in an L-frame

[2] $(\cdots)_M$ indicates division mod M.

as well as the PPM parameters introduced in 1.2.

first phase

Temporal synchronization with the AP: L-slot synchronization

- at the beginning, the MT is in the *start state* and moves to the *PSC code acquisition* operation block. The receiver continuously searches for the PSC sequence by performing code acquisition with the PSC locally reproduced. It remains in the code acquisition block until the output of the correlator shows a sufficiently high peak. This peak reveals the position of the beginning of the L-slot and helps to produce a coarse alignment between the beginning of the L-slot and the local clock.

- once that the PSC has been acquired, the MT must perform a further *PSC code tracking* in the dedicated operation block. This block is implemented through a tracking loop (see [11]) controlled by a time counter. This counter is necessary to avoid that false alarms coming out from the analysis operated by the code acquisition block, could produce infinite loops. If time elapses without the completion of the code synchronization the MT goes back to the *reset state*, otherwise, it proceeds to the next phase.

second phase

Temporal synchronization with the AP: L-frame synchronization

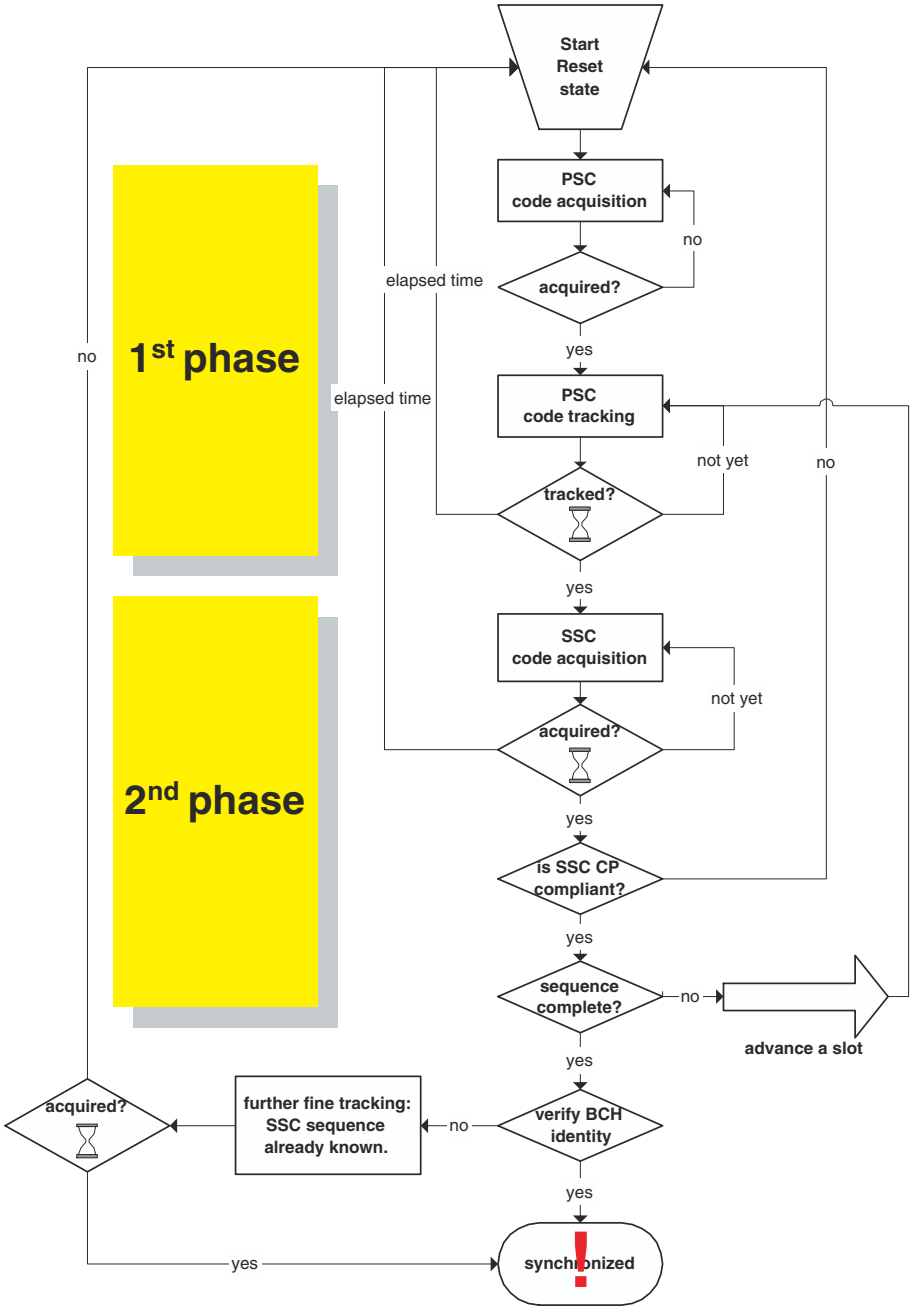

Figure 3.5: Proposed algorithm for the acquisition of the broadcast channel code and timing.

BCH-id selection

- after the completion of phase 1 the MT knows the position in time of the beginning of the L-slot. In the *SSC code acquisition* block the MT searches the SSC that has been transmitted superimposed on the PSC. The SSC must occupy the same time window delimited by the PSC, i.e. an interval of width T_{SC} starting from the beginning of the PSC. The search is exploited through the correlation, only within that time window, between the received signal and the local copy of each SSC that constitutes the CPC code alphabet α. This block is followed by a counter controlled verifier that brings the procedure to the reset state if a false alarm is detected, i.e. no SSC has been recognized so far.

- the *SSC CPC compliant* block determines if the SSC just recognized is compatible with the SSCs already received. In practice, it controls that the BCH-id codeword partially reconstructed (from previously received SSCs) is the initial part of at least one word of the BCH-ids' codebook or of one of its cyclic shifts. If this condition fails, the procedure ends in the reset state, thus starting again.

- if everything went ok, the MT checks if the received sequence is complete, i.e. if all the M SSCs have been received. If this is not the case, the MT proceeds to a further iteration. Knowing the PSC/SSCs repetition period T_{Ls}, the MT directly advances in time by that period,

positioning the correlator on the next expected time window and entering into the *PSC code tracking* block to compensate for misalignment due to transmission delays. After that, it continues the iteration till completion.

- once the sequence of SSCs is complete, the MT is able to recognize the BCH-id together with the order i of the received circular shift. From these informations the MT can find the BCH that is assigned to the AP (the BCH-id addresses the code assigned to the BCH in the BCHs' codebook) and it can accomplish the L-frame synchronization. A last verify is obtained by trying to read the binary translation of the BCH-id sequence which is transmitted at a known position in the L-frame. If something goes wrong, before resetting the procedure, a further tracking is implemented taking advantage of the knowledge of the complete SSCs sequence. If everything went ok, or we have managed to correct the tracking, we reach the *synchronized state*.

4

Code division multiplexing

As told in 1.2, *code division multiplexing* is adopted in order to reduce the cross channel interference. Code division multiplexing implies that different channels are multiplexed over a shared resource, i.e. the available bandwidth, by reserving to each ph-CH a unique code. Since UWB makes use of PPM to

convey information, it is necessary to find orthogonal time hopping codes to exploit multiplexing. The results obtained in the study of frequency hopping codes can be extended to the time domain simply following the approach described in [14]. In this way it is possible to take advantage of some frequency hopping codes' families which show a good behavior even if used as time hopping codes.

4.1 Definitions

Definition 4.1.1. *In a frequency hopping system a single transmitted pulse of length T seconds is subdivided into N_s equal segments of length $\frac{T}{N_s}$. Each segment is transmitted through* 1 *of N_h equally spaced frequencies ω_0 to ω_{N_h-1} and the frequency sequence is determined via a sequence of ordered integers $y(k)$ which is called* the placement operator.

Analogously:

Definition 4.1.2. *In a time hopping system as that introduced in 1.2 a single transmitted bit of length T_b seconds is subdivided into N_s equal segments of length $T_f = \frac{T_b}{N_s}$ called time frames further subdivided into N_h time intervals (i.e.* time chips*) of length T_c. Each segment is transmitted through* 1 *of N_h time chips and the time chips sequence is determined via a sequence of ordered integers $y(k)$ which is called* the placement operator.

Definition 4.1.3. *The placement operator $y(k)$ is a periodic of period N_p integer valued function on the set of integers $J_{N_h} = \{0, 1, 2, \cdots, N_h - 1\}$. If N_h is an odd prime number then J_{N_h} is a finite field. Some code families are defined on the integer set defined as $\dot{J_{N_h}} = J_{N_h} - \{0\}$ called the group. All the placement operators introduced later will have $N_p = N_s = N$.*

Definition 4.1.4. *Let $J(p)$ denote a finite field over a prime p. Denote by J_p the set containing the elements of the field. A* full time/frequency hopping code *is a time/frequency hopping code whose placement operator is a permutation of the set J_p or $\dot{J}_p$ where, $\dot{J}_p = J_p - \{0\}$. A full code has always $N_h = N_p = N$.*

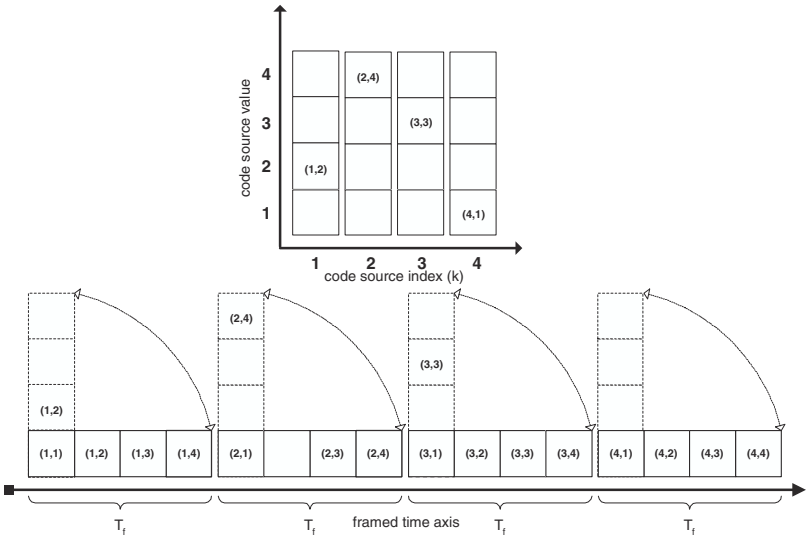

Figure 4.1: Transition from a time hopping code to its matrix representation.

A practical representation of frequency hopping codes is the *code matrix*. Each code of period N_p is associated with an $N_h \times N_p$ matrix whose k^{th} column contains a 1 in the row indexed by the placement operator $y(k)$ and some 0 in the remaining positions. The i^{th} row is associated with the frequency ω_i while the columns are associated with the time segments T/N_s. The same representation can be adopted for time hopping codes. The N_h rows correspond to the N_h possible time chips of length T_c seconds in a frame T_f, and the N_p columns correspond to the N_p time frames each of width T_f seconds in which the code is not repeating itself (code period $T_p = N_p \cdot T_f = N_s \cdot T_f$). The element $a_{l,k}$ of the matrix is equal to 1 if and only if the transmission in the frame k happens in the l^{th} time chip, i.e. $y(k) = l$, otherwise it is 0. The figure 4.1 shows the transition from the time hopping code matrix to the transmitted sequence. A code matrix is biunivocally associated with a placement operator $y(k)$ and it is often called *placement array*.

Definition 4.1.5. *When two placement arrays are superimposed with a relative shift of (x, z) in the horizontal and vertical direction, each position coincidence between a couple of* 1 *belonging to different placement arrays is called an* hit.

Definition 4.1.6. *The* placement difference function *for two placement op-*

erators $y_1(k)$, $y_2(k)$ of period p is defined as:

$$[y_1(k) \triangle y_2(k)]_{x,z} = y_2(k+x) - y_1(k) + z \ mod_p \ with \ |x|, |z| \leq p-1 \quad (4\text{-}1)$$

where x and z denote an integer horizontal and vertical shift, respectively. When an hit is encountered the placement difference function is zero.

Having introduced code matrices (i.e. placement arrays) and having explained the meaning of an hit, it is possible to understand the following definitions:

Definition 4.1.7. *Let C denote the $N \times N$ matrix representing a full time/frequency hopping code, $y(k)$. The auto-hit array $H(x,z)$ is a $(2N-1)\times(2N-1)$ matrix whose elements, $h_{x,z}$, $-N+1 < x, z < N-1$, correspond to the number of hits between the matrix C and the shifted version by (x,z) of itself. The auto-hit array is odd symmetric. The concept of the auto-hit array is easily extended to the cross-hit array.*

Definition 4.1.8. *Let C_1 and C_2 denote two $N \times N$ full time/frequency hopping codes, $y_1(k)$ and $y_2(k)$, respectively. The cross-hit array $C^{1,2}$ is a $(2N-1) \times (2N-1)$ matrix whose elements $c_{x,z}$, $-N+1 < x, z < N-1$, correspond to the number of hits between matrix C_2 shifted by (x,z) and the unshifted version of C_1.*

The number of zeros of the placement difference function calculated in

(x, z) with:

$$k \begin{cases} k + x \in J_N & \text{acyclic horizontal shift} \\ y(k + x) + z \in J_N & \text{acyclic vertical shift.} \end{cases} \tag{4-2}$$

coincide with the element (x, z) of the hit matrix considered.

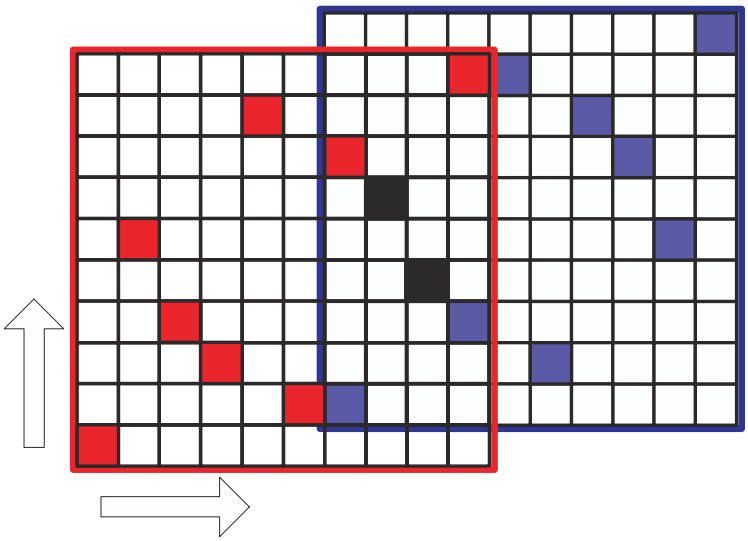

Figure 4.2: Hit-array computation.

4.2 Correlation measures

To evaluate the interference between different ph-CHs (i.e. between distinct time/frequency hopping codes) it is necessary to reckon the number of collisions in time (time hopping codes) or in time and frequency (frequency hopping codes) which may happen after transmission (for a deep insight refer to [15]). The lower is the maximum number of hits between distinct codes,

the higher is their reciprocal orthogonality and the lower is their reciprocal interference.

In asynchronous frequency hopping systems all the interactions between the transmitted hopping codes are obtained by shifting a placement array over the other along the time direction x and, if there is an uncertainty about the received frequencies, even along the frequency direction y. Cross-hit arrays are used to measure the interferences between different ph-CHs as follows. When frequency hopping codes are used and the transmission is

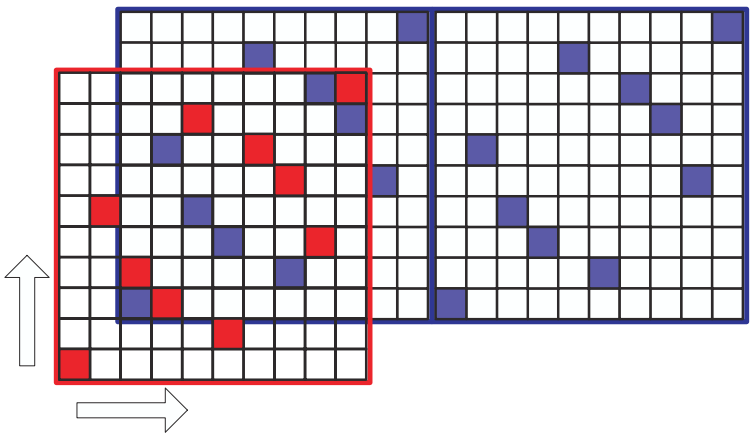

Figure 4.3: Hits computation for frequency hopping codes with continuous retransmission.

intervalled by silences of at least $N_p \cdot T$ seconds (i.e. *bursted transmission*) all the interactions are obtained as displayed in figure 4.2. Since the hit array is always calculated following that scheme (see definitions 4.1.7,4.1.8), it represents a statistic of the collisions detected among all the possible combinations that can be observed. Therefore the hit array can be used directly

as a performance measure against the cross channel interference. If continuous transmission is adopted, all the interactions are obtained as in figure 4.3 where it is outlined the presence of a second placement array, adjacent in time, and interfering with the shift of the considered placement array. The hit array is not sufficient to describe all the possible superimpositions that come out from this condition but it can indirectly define maximum limits: if the hit array has a maximum of M hits, the superimposition of figure 4.3 will show at worst $2 \cdot M$ hits. An analytical expression for the hits of frequency hopping codes transmitted continuously is obtained by defining:

$$c_m^{(i)} = y_i(k = m) \tag{4-3}$$

which is periodic with period N_p for the periodicity of the placement operator, i.e.:

$$c_m^{(i)} = c_{m \ mod_{N_p}}^{(i)}.$$

An hit is obtained for each $m \in J_{N_p}$ which satisfies the equation:

$$c_m^{(i)} + z = c_{m+x}^{(j)}. \tag{4-4}$$

Since in the case of bursted transmission the hits are obtained only for each m which simultaneously satisfies the equation (4-4) and the inequality $0 \leq m + x < N_p$, it follows that the hits number is lower than in continuous transmission.

When time hopping codes are adopted with the hypothesis of bursted transmission (PSC and SSCs are transmitted in bursts) the analysis must

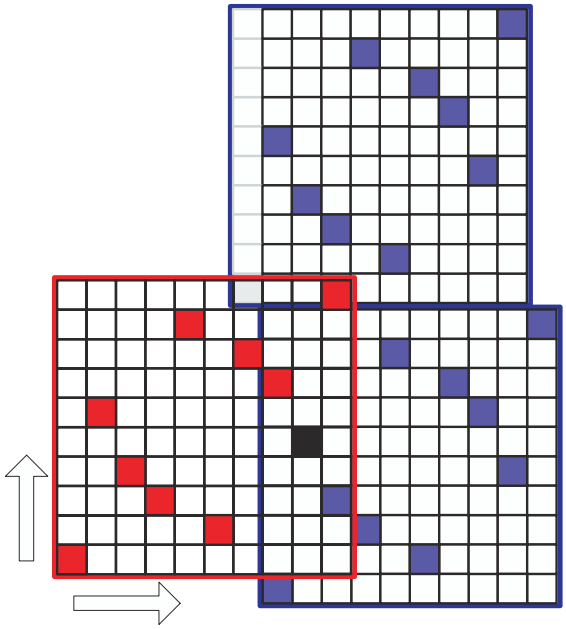

Figure 4.4: Hits computation for time hopping codes with bursted transmission.

be operated carefully. For construction, in time hopping code arrays both axes represent the time domain[1] and adjacent columns represent a continuous sequence in time (see figure 4.1). The situation observed is equivalent to that shown in figure 4.4 where the continuity among adjacent columns is maintained. It is clear that the hit array is not sufficient to perform its evaluation but it can be said that: if the hit array has a maximum of M hits, the superimposition of figure 4.4 will show at worst $2 \cdot M$ hits. When continuous transmission is adopted the situation becomes the one displayed in figure 4.5 and the hit limit doubles from that obtained for bursted transmission and it results in at worst $4 \cdot M$ hits.

[1] With intervals of T_c seconds in the vertical direction and intervals of T_f seconds horizontally.

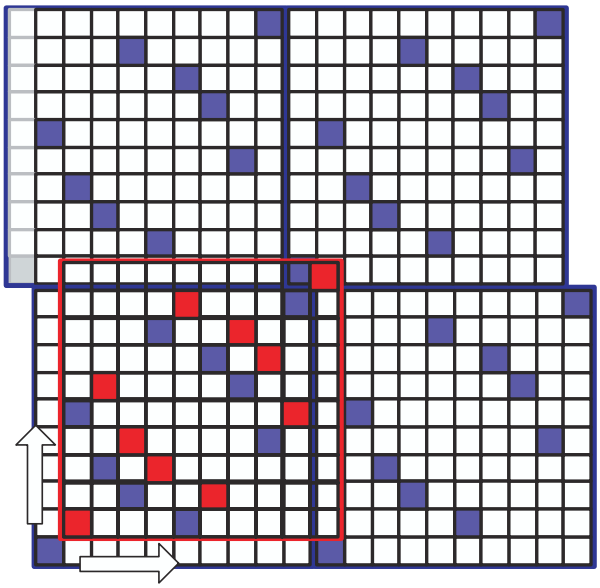

Figure 4.5: Hits computation for time hopping codes with continuous transmission.

Through a comparison between the conclusions obtained graphically for time hopping and for frequency hopping codes it is possible to conclude that:

Conclusion 4.2.1. *Time hopping implies a doubling of the limits computed for frequency hopping codes.*

▸ *proof:* the signal associated to the time hopping (TH) sequence $\{c_m^{(i)}\}$ of period N_p and values in the set J_{N_h} coincides with the binary expression, discrete in time and periodic of period $(N_h \cdot N_p)$ which follows:

$$s_n^{(i)} = \sum_{m=0}^{N_p-1} \delta^{(N_h \cdot N_p)}_{n,m \cdot N_h + c_m^{(i)}} \tag{4-5}$$

where $\delta^{(N_h \cdot N_p)}_{n,m}$ is the Kronecker delta function periodic of period

$N_h \cdot N_p$, i.e

$$\delta_{n,m}^{(N_h \cdot N_p)} = \begin{cases} 1 & if\ n = m\ mod(N_h \cdot N_p) \\ 0 & otherwise \end{cases}$$

Auto and cross-correlation are periodic functions of period $(N_h \cdot N_p)$, in particular the cross-correlation between user i and user j is

$$C_{i,j}(u) = \sum_{n=0}^{N_h \cdot N_p - 1} s_n^{(i)} \cdot s_{n+u}^{(j)}$$

which using (4-5) becomes:

$$C_{i,j}(u) = \sum_{n=0}^{N_h \cdot N_p - 1} \left(\sum_{m=0}^{N_p - 1} \delta_{n, m \cdot N_h + c_m^{(i)}}^{(N_h \cdot N_p)} \cdot \sum_{r=0}^{N_p - 1} \delta_{n+u, r \cdot N_h + c_r^{(j)}}^{(N_h \cdot N_p)} \right)$$

or

$$\sum_{m,r=0}^{N_p - 1} \delta_{m \cdot N_h + c_m^{(i)} + u, r \cdot N_h + c_r^{(j)}}^{(N_h \cdot N_p)} \tag{4-6}$$

It suffices to study the limits for $0 \leq u < N_p$ which if $u = x \cdot N_h + z$ translate into the relations

$$0 \leq x < N_p \qquad 0 \leq z < N_h$$

An hit happens when for $m, r \in [0, N_p - 1]$:

$$m \cdot N_h + c_m^{(i)} + x \cdot N_h + z = r \cdot N_h + c_r^{(j)}\ mod(N_h \cdot N_p)$$

or

$$(m + x)N_h + c_m^{(i)} + z = r \cdot N_h + c_r^{(j)}\ mod(N_h \cdot N_p). \tag{4-7}$$

Since $0 \leq z, c_m^{(i)} < N_h$ it follows that $0 \leq z + c_m^{(i)} < 2(N_h - 1)$ and two cases may happen depending on m:

1. if $z + c_m^{(i)} < N_h$ then for $m, r \in [0, N_p - 1]$ the equation (4-7) means that an hit exists when

$$r = m + x$$

and

$$c_m^{(i)} + z = c_{m+x}^{(j)} \tag{4-8}$$

i.e. the user j occupies the time chip $c_m^{(i)} + z$ in the frame $m + x$

2. if $z + c_m^{(i)} \geq N_h$ then the equation (4-7) becomes

$$(m + x + 1)N_h - N_h + c_m^{(i)} + z = r \cdot N_h + c_r^{(j)} \ mod(N_h \cdot N_p).$$

and it means that for $m, r \in [0, N_p - 1]$ an hit exists when

$$r = m + x + 1$$

and

$$c_m^{(i)} + z - N = c_{m+x+1}^{(j)}$$

or

$$c_m^{(i)} + z = c_{m+x+1}^{(j)} + N \tag{4-9}$$

i.e. the user j occupies the time chip $c_m^{(i)} + z - N$ in the frame $m + x + 1$,

The total number of hits for the time hopping codes is computed from the union of the solutions sets for both equations (4-8),(4-9) i.e

$$c_m^{(i)} + z = c_{m+x}^{(j)} \quad \bigcup \quad c_m^{(i)} + z = c_{m+x+1}^{(j)} + N. \tag{4-10}$$

The equation (4-8) coincides with (4-4) and since equation (4-9) is very similar to (4-8) its solutions' set can be guessed analogous to that of (4-8). All said, from (4-10) it derives that the the maximum hits for time hopping codes are at worst the double of those observed for frequency hopping codes ■

4.3 Frequency and Time Hopping Codes

4.3.1 Hyperbolic Congruence Codes (HCCs)

Table 4.1: Primes less than 500, of the form $N = 3m + 2$

	53	131	227	293	401
5	59	137	233	311	419
11	71	149	239	317	431
17	83	167	251	347	443
23	89	173	257	353	449
29	101	179	263	359	461
41	107	191	269	383	467
47	113	197	281	389	491

Hyperbolic Congruence Codes (HCCs) have been introduced in [16] as a solution to multiplexing in frequency hopping systems. As told at page 69 the code matrix representation used in [16] can be extended to the use of HCCs in time hopping systems thus obtaining an $N_h \times N_p$ placement array. HCCs are full codes, this implies that $N_h = N_p = N$ (see definition 4.1.4), and they are described by $N \times N$ placement arrays. HCCs are defined as follows:

Definition 4.3.1. *A Hyperbolic Hop Code for a finite field, $J(p)$, is defined through the following placement operator*

$$y^{HCC}(k) = \frac{a}{k} \ mod \ p \ with \begin{cases} k & \in \ [1, 2, \cdots, p-1] \equiv \dot{J}_p \\ a & \in \ \dot{J}_p \end{cases} \tag{4-11}$$

Since the operation in 4-11 is performed over a finite field[2]*, every element in the field has a unique inverse (see example in Table 4.2) and it is immediately seen that the hyperbolic codes are full codes with $N = p - 1$ pulses. Also, for each prime there are $p - 1$ different hyperbolic codes.*

Conclusion 4.3.1. *An hyperbolic hop code has at most two hits for any not null time-frequency shift (i.e. for any translation different from (0,0)) in its auto-hit array and at most two hits for any time-frequency shift in its cross-hit array formed with any other hyperbolic frequency hopping code of the same length.*

[2]The inverse of n over a finite field is that element belonging to the finite field whose product with n coincides with the neutral element in the field, with respect to the product operation

Table 4.2: Inversion over a field operation: example in the case $p = 11$ and $a = 1$

Value	Operation	Inverse
1	$1 \cdot 1 = 1\ mod_{11} = 1$	1
2	$2 \cdot 6 = 12\ mod_{11} = 1$	6
3	$3 \cdot 4 = 12\ mod_{11} = 1$	4
4	$4 \cdot 3 = 12\ mod_{11} = 1$	3
5	$5 \cdot 9 = 45\ mod_{11} = 1$	9
6	$6 \cdot 2 = 12\ mod_{11} = 1$	2
7	$7 \cdot 8 = 56\ mod_{11} = 1$	8
8	$8 \cdot 7 = 56\ mod_{11} = 1$	7
9	$9 \cdot 5 = 45\ mod_{11} = 1$	5
10	$10 \cdot 10 = 100\ mod_{11} = 1$	10

▸ *proof:* taking advantage of definition 4.1.6 it is possible to compute the number of hits for each x, z relative shift of two placement arrays. The placement difference function for the HCCs is obtained by substituting (4-11) in (4-1) obtaining:

$$\left[y_i^{HCC}(k) \triangle y_j^{HCC}(k)\right]_{x,z} = \left[\frac{i}{k+x}\right]_p - \left[\frac{j}{k}\right]_p + z\ mod_p$$

where the notation $[...]_p$ indicates division mod(p) (mod_p). The hits are the solutions with respect to k of the equation:

$$\left[\frac{i}{k+x}\right]_p - \left[\frac{j}{k}\right]_p + z = 0\ mod_p.$$

When no cyclic shifts are considered, as in the case of hit arrays (see figure 4.2), a hit exists only when for x, z given

$$k \text{ satisfies } \begin{cases} 0 < k + x < p & \text{acyclic horiz. shift} \\ \left[\frac{i}{k+x}\right]_p + z \in \dot{J}_p & \text{acyclic vert. shift.} \end{cases} \tag{4-12}$$

This implies that:

$$\left[\frac{i}{k+x}\right]_p - \left[\frac{j}{k}\right]_p + z =_p 0 \Leftrightarrow \left[\frac{i}{k+x} + z\right]_p - \left[\frac{j}{k}\right]_p =_p 0$$

and observing that $k \neq 0$ for construction and $k+x \neq 0$ for (4-12) the latter becomes:

$$\frac{i}{k+x} + z - \frac{j}{k} =_p 0.$$

and the zeros are found from the equation:

$$\frac{k \cdot i + k(k+x)z - (k+x)j}{k(k+x)} =_p 0$$

or

$$z \cdot k^2 + (i - j + x \cdot z)k + x \cdot j =_p 0. \tag{4-13}$$

When $z \neq 0$ and $i = j$ or $i \neq j$ equation (4-13) is of the second order in k and for the Lagrange theorem it has at most two non congruent solutions which imply at most two hits in the auto/cross-hit arrays. In particular, for $x = 0, i = j$ no solutions are allowed, while for $x = 0, i \neq j$ both solutions are coincident

and produce a single hit.

When $z = 0, i = j$ equation (4-13) becomes:

$$x \cdot j =_p 0$$

and, since p is a prime number, the solutions exist only for $x = 0$. The equation (4-13) results in the identity $0 = 0$ which implies that all the values for $k \in \dot{J}_p$ are a zero[3]. The total number of hits is $p - 1$; this value is the peak of the auto hit array in the origin $(0, 0)$.

When $z = 0, i \neq j$ equation (4-13) becomes:

$$(i - j)k + x \cdot j =_p 0$$

and this is a first order equation which admits at most one solution and in particular no solutions for $x = 0$.

[3]Remember from (4-12) that $0 < k + x < p$ and substituting $x = 0$ the range of k is [1,p-1].

The results are summarized as follows:

$$\begin{array}{cc} \text{auto-hit max limits} & \text{cross-hit max limits} \\ \left[\begin{array}{c|c|c} \vdots & \vdots & \vdots \\ \cdots 2 \cdots & 0 & \cdots 2 \cdots \\ \vdots & \vdots & \vdots \\ \hline \cdots 0 \cdots & p-1 & \cdots 0 \cdots \\ \hline \vdots & \vdots & \vdots \\ \cdots 2 \cdots & 0 & \cdots 2 \cdots \\ \vdots & \vdots & \vdots \end{array}\right] & \left[\begin{array}{c|c|c} \vdots & \vdots & \vdots \\ \cdots 2 \cdots & 1 & \cdots 2 \cdots \\ \vdots & \vdots & \vdots \\ \hline \cdots 1 \cdots & 0 & \cdots 1 \cdots \\ \hline \vdots & \vdots & \vdots \\ \cdots 2 \cdots & 1 & \cdots 2 \cdots \\ \vdots & \vdots & \vdots \end{array}\right] \end{array} \tag{4-14}$$

These results demonstrate conclusion 4.3.1■

Experimental results confirm conclusion 4.3.1. The figure 4.6 shows the auto-hit array of an HCC derived from $p = 101$, i.e. $N_p = N_h = N = 100$. The x-axis and the y-axis represent relative shifts in the reciprocal directions while the z-axis shows the hit-matrix values normalized to the peak (i.e. 100). As expected the sub-maximum is 2 and represents only a 2% of the peak, this percentage is inversely proportional to p. A further confirmation comes from figure 4.7 which displays the cross-hit matrix between a couple of different HCCs with $p = 101$. The values on the z-axis are normalized to the auto-hit array peak value (i.e. 100) and once again they are always beneath the value 2 (which in this case is the 2% of the auto-hit peak).

Having computed the maximum limits in the hit arrays, it is possible to apply conclusion 4.2.1 obtaining that:

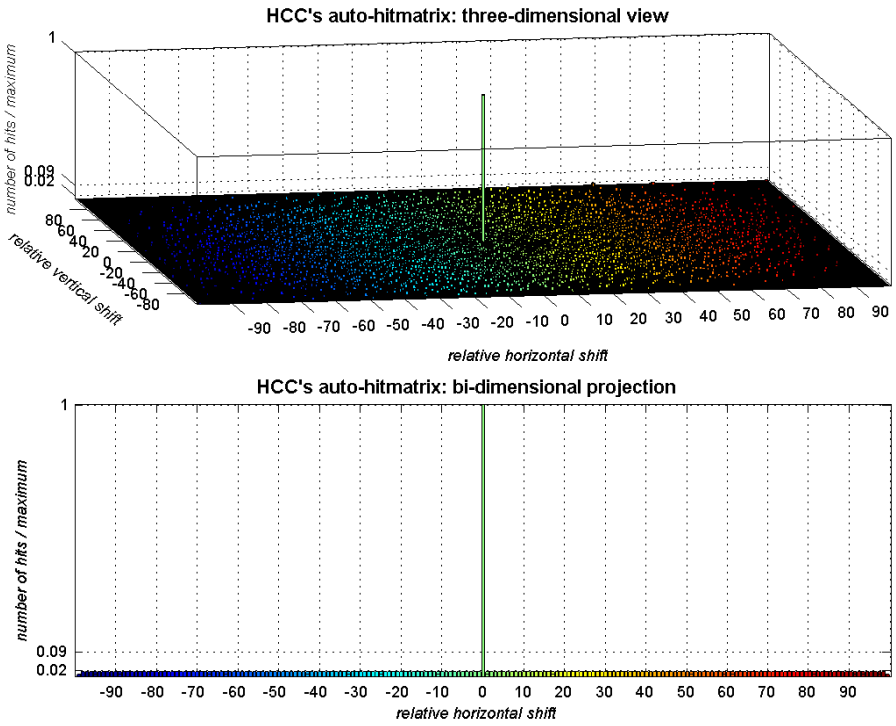

Figure 4.6: Auto-hit matrix of an Hyperbolic Congruence Code generated with $p = 101$

Conclusion 4.3.2. *An hyperbolic time hopping code of any period* $N_p = p-1$ *when it is transmitted in bursts has at most four hits against a time shifted replica of itself (not including null shifts, i.e. (0,0) translations) and four hits against a different, however shifted hyperbolic congruence code of the same period. Its auto-hit maximum is in the origin and grows with* $p - 1$.

Experimental results confirm conclusion 4.3.2. In figure 4.8 it is displayed the auto-correlation of an HCC time hopping sequence derived from $p = 101$, i.e. $N_p = N_h = N_s = 100$. As expected, the peak of the correlation is 100 and must be confronted with a sub-maximum of 4 independent from p.

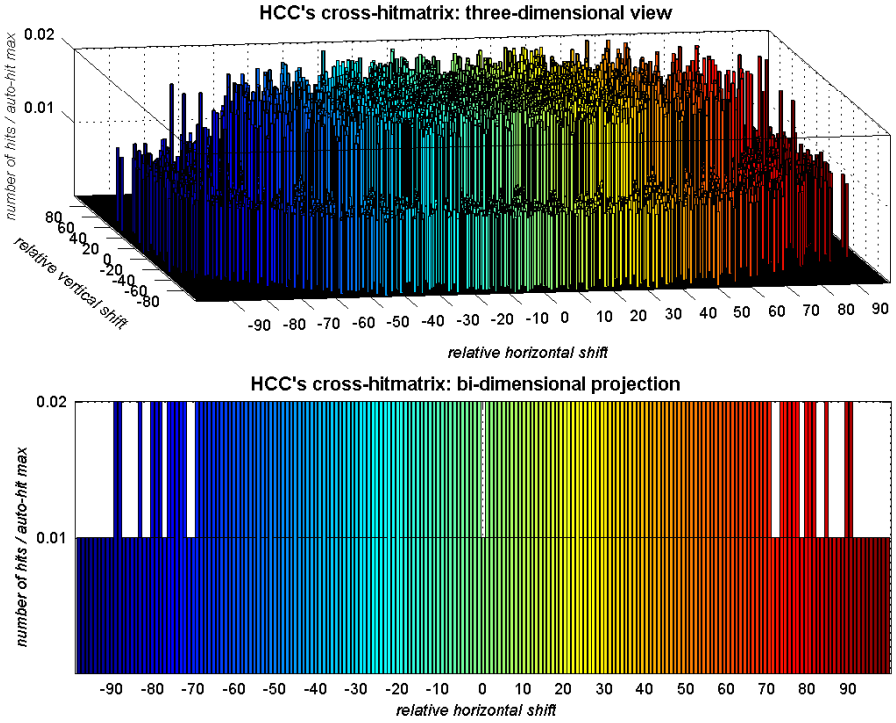

Figure 4.7: Cross-hit matrix of two different Hyperbolic Congruence Codes generated with $p = 101$.

From conclusion 4.3.2 it is clear that by augmenting $p - 1$, which in the case of time hopping codes is equal to $N_p = N_h = N_s$, the performance against cross channel interference improves. In fact, cross-hit arrays' maximum and auto-hit array's sub-maximum represent a disturb which remains limited to the value 4 when p increases while the peak of the auto correlation grows linearly. As a drawback, since $N_p = N_h$, an increment in N_p implies an increment in N_h which brings to an increase in T_f, i.e. by improving the response to interferences the duty cycle of the PPM signal is diminished and the transmitted bit rate further decreases. All this is reported in figure 4.9

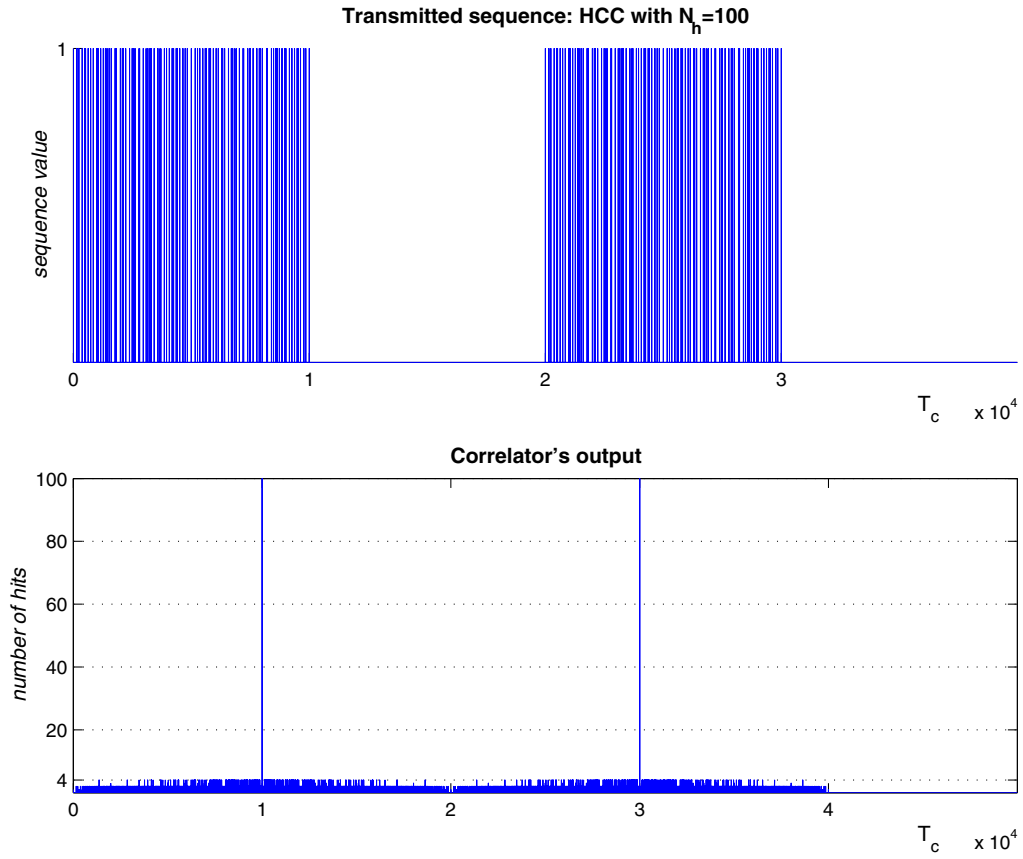

Figure 4.8: Auto correlation of a time hopping HCC sequence with $p = 101$.

where some statistics over hit arrays have been displayed. In that figure the x-axis refers to the independent variable N_h and the left y-axis refers to the PPM signal's frame length $T_f = T_c \cdot N_h$. Instead the right y-axis refers to the mean number of hits between two code arrays[4] further averaged between all the possible couples of HCCs that can be generated with a given N_h and further normalized to the peak value. To compute the T_f expressed in ns it

[4]This mean is the average among all the relative shifts which produce a hit number not null (peak included).

is considered a time chip $T_c = 1ns$; it can be seen that the T_f grows linearly with N_h thus reducing the duty cycle of the transmitted signal. Instead performances are improving monotonically because the normalized hit average diminishes with the increase of N_h.

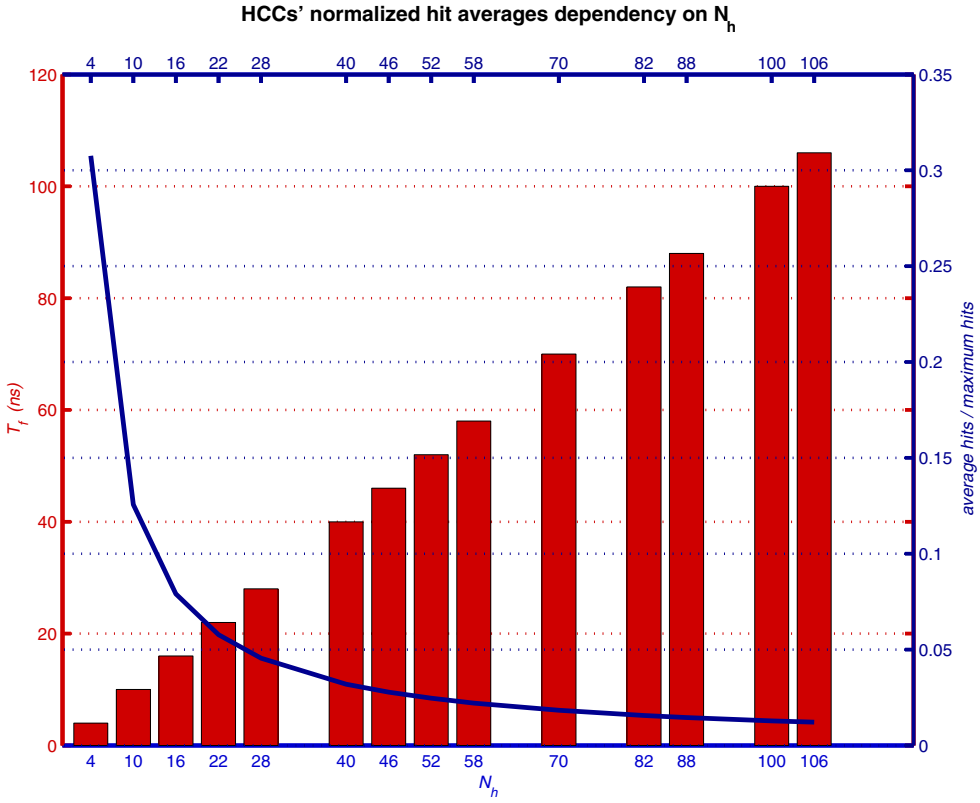

Figure 4.9: Statistics about HCCs' hit arrays as a function of N_h ($T_c = 1ns$).

4.3.2 Extended Hyperbolic Congruence Codes (HCCXs)

As told at page 79 HCCs suffer from having the bond $N_h = N_p$ which limits their flexibility. A simple way to overcome this problem is to build longer

sequences by juxtaposing a number m of HCCs, where m is called *multiplicity*. This operation alters the orthogonality properties, but the consequences are controlled by taking care that a single HCC should be used by one, and only one, long sequence. When a code is built up in this way it is called *extended hyperbolic congruence code (HCCXs)*. If the number of available HCCs is $N = N_h = N_p$, the respective set of HCCXs has a reduced cardinality equal to $\left\lfloor \frac{N}{m} \right\rfloor$, where $\lfloor ... \rfloor$ is the low integer part operator. The HCCXs considered here have been constructed by partitioning $\dot{J}_p$ in m contiguous subsets each with $\left\lfloor \frac{N}{m} \right\rfloor$ distinct coefficients a. An HCCX is obtained juxtaposing the m HCCs which are generated substituting the homologous coefficients[5] in the placement operator (4-11) (see table 4.3).

Table 4.3: HCCXs construction: $N = p - 1 = 10, m = 2$

code	HCCs' sequence	subset	I	II	III	IV	V
$HCCX_1$	$HCC_1\ HCC_6$	$\mathcal{A}$	1	2	3	4	5
$HCCX_2$	$HCC_2\ HCC_7$						
$HCCX_3$	$HCC_3\ HCC_8$	$\mathcal{B}$	6	7	8	9	10
$HCCX_4$	$HCC_4\ HCC_9$						
$HCCX_5$	$HCC_5\ HCC_{10}$						

Definition 4.3.2. *In the auto/cross hit arrays all the hits apart from those reckoned in the origin (0,0) of the auto-hit array (i.e. the peak value) are*

[5] One for each subset.

considered undesired hits (UHs).

It has been said that the maximum number of UHs for hyperbolic congruence frequency hopping codes (HCfCs) transmitted in burst is 2 (see conclusion 4.3.1) while for hyperbolic congruence time hopping codes (HCtCs) transmitted in bursts it is 4 (see conclusion 4.3.2). The maximum number of UHs for extended hyperbolic congruence frequency hopping codes (HCfCXs) is computed observing that when bursted transmission is performed the first $m-1$ HCCs which constitute the HCCX behave as explained in figure 4.3 (they are followed by another HCC transmitted continuously), while the last one behaves as in figure 4.2 (it is followed by a silence). Consequently for what said about those figures:

$$frequency\ hopping \Rightarrow UH^{max}_{HCfCX} = 4(m-1)+2. \tag{4-15}$$

When HCCXs are transmitted through time hopping (HCtCXs) each HCC which is part of the sequence must be substituted by the configuration displayed in figure 4.3 and the maximum limit becomes the double of (4-15) as it was predictable through conclusion 4.2.1:

$$time\ hopping \Rightarrow UH^{max}_{HCtCX} = 8(m-1)+4. \tag{4-16}$$

In both cases the UH^{max} are greater than in the case of HCCs. A performance measure for the perceived orthogonality of the family of codes is expressed

by the ratio (which must be as low as possible):

$$R = \frac{UH^{max}}{\mathcal{P}}$$

where $\mathcal{P}$ is the auto-hit peak value. A comparison is exploited by analyzing the expressions:

$$\frac{R_{HCCfX}}{R_{HCfC}} = \frac{4(m-1)+2}{m(p-1)} \cdot \frac{p-1}{2} = \frac{2m-1}{m} \tag{4-17}$$

and

$$\frac{R_{HCCtX}}{R_{HCtC}} = \frac{8(m-1)+4}{m(p-1)} \cdot \frac{p-1}{4} = \frac{2m-1}{m}. \tag{4-18}$$

The equations (4-17) and (4-18) represent quantities which are always greater than 1 for $m > 1$ and imply that in the HCCXs the perceived orthogonality has been reduced from that of the HCCs. In practice, the limits (4-15) and (4-16) are too pessimistic since with a bit of attention it is possible to prevent the worst case (of 4/8 hits respectively for frequency/time hopping codes) from happening simultaneously in all the m HCCs of the considered HCCX. The observed expressions for UH^{max} assume the more realistic forms

$$UH^{max}_{HCfCX} = \alpha(m-1)+2$$

$$UH^{max}_{HCtCX} = 2 \cdot \alpha(m-1)+4$$

where $1 \leq \alpha \leq 4$; (4-17) and (4-18) become:

$$\frac{R_{HCCX}}{R_{HCC}} = \frac{\alpha \cdot m - \alpha + 2}{2m} = \frac{\alpha \cdot (m-1)+2}{2m}$$

which is a quantity monotonically decreasing with α (since $m > 1$) and greater than 1 only if $\alpha > 2$. The parameter α is the the maximum number of hits due to a single HCC averaged between all the HCCs which build up the HCCX sequence. When m increases this mean diminishes because the probability that the worst case may happen in all the composing HCCs decreases with m: *α manifests an inverse proportionality against multiplicity.* When p is increased, that probability grows because it is greater the number of combinations which bring to the maximum number of hits: *α manifests a direct proportionality with p.* The coefficient α presents a further dependency on the type of partitioning which is chosen for $\dot{J}_p$. Different permutations of the same HCCs in a single HCCX behave differently since each hit array has the hits distributed in a unique way, and some combinations might be more efficient than others. With the correct choices it is possible to obtain an $\alpha < 2$ which results in a better behavior of the HCCXs if confronted with the HCCs with the same p. It is difficult to explicit the analytical dependency of α on the components mentioned afore and this is not done here, instead some statistical results are presented to show the applicability of what said.

The maximum values for the undesired hits between HCtCs with $N_h = 10$ are visible in table 4.4 where the entry (i, j) is the maximum number of UHs between the codes i and j. It is clear that the limit 4 of conclusion 4.3.2 is reached between more than one couples of codes. When HCtCXs are constructed from the HCCs with $N_h = 10$ the maximum number of UHs

Table 4.4: Undesired hits limits for HCtCs with $N_h = 10$

code	1	2	3	4	5	6	7	8	9	10
1	2	3	2	2	2	3	2	3	2	4
2	3	4	3	2	3	2	3	2	2	3
3	2	3	4	3	2	3	2	3	3	3
4	2	2	3	4	3	3	4	2	3	3
5	2	3	2	3	2	2	3	2	3	3
6	3	2	3	3	2	2	3	3	3	2
7	2	3	2	4	3	3	3	2	3	2
8	3	2	3	2	2	3	2	4	3	2
9	2	2	3	3	3	3	3	3	2	2
10	4	3	3	3	3	2	2	2	2	4

observed are those in the tables 4.5. For each multiplicity m it is presented the value of the observed coefficient α together with the UHs'array. Since in all cases $\alpha < 2$, the expression (4-18) is lower than unity and it can be said that HCC extension brings to an increase in the perceived orthogonality. While m grows α diminishes thus improving the observed behavior. Similar

Table 4.5: Undesired hits limits for HCtCXs with $N_h = 10$

$m = 2 \Rightarrow \alpha = 1$

code	1	2	3	4	5
1	5	4	4	5	4
2	4	6	5	4	5
3	4	5	6	4	4
4	5	4	4	5	5
5	4	5	4	5	5

$m = 3 \Rightarrow \alpha = 1$

code	1	2	3
1	8	6	7
2	6	6	6
3	7	6	7

$m = 4 \Rightarrow \alpha = 0.83$

code	1	2
1	9	8
2	8	8

$m = 5 \Rightarrow \alpha = 0.625$

code	1	2
1	9	9
2	9	9

statistics are computed with $N_h = 22$: HCtCs (see table 4.6) manifest the maximum 4 (once again see conclusion 4.3.2), while the UHs for the derived HCtCXs are those in table 4.7. Apart from the case where $m = 2$ which implies $\alpha = 2$ all the other multiplicities produce $\alpha < 2$ which result in a better perceived orthogonality than that which is obtained with the starting HCCs. The coefficient α reveals its tendency to diminish with the growth of m with the exception of the passages $m : 7 \longrightarrow 8$ and $m : 10 \longrightarrow 11$. The reason that justifies this behavior is that α has a dependency on $\dot{J}_p$ which can favor or corrupt the performance. When the HCCXs are constructed in the way introduced at page 88 the $\dot{J}_p$ varies with m and its effect can prevail over the beneficial action of m thus producing the slight increment which has been observed. It is interesting to notice that with m being equal, the respective values of α are greater for $N_h = 22$ than for $N_h = 10$ as a result of the announced direct proportionality with p. In table 4.20 some values for α are presented as they observed with different determinations of the multiplicity and of the number of hops. Together with $N_h = 10, N_h = 22$ the cases $N_h = 52, N_h = 100$ are in part studied. The numeric solutions present some further oddities deriving once again from the varying of J_{N_h} with m but the general tendency is to have rows with results decreasing towards the right border and columns increasing downward. Once that N_h has been chosen the multiplicity must be selected balancing the tradeoff of reducing the number of symbols available for the CPC alphabet. This is the only limit

to the value of m apart from the constrain of not trespassing the quantity $N_h/2$ which guarantees at least two codes applicable to binary CPCs and of not creating too long sequences.

Table 4.6: Undesired hits limits for HCtCs with $N_h = 22$

code	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	4	3	3	3	3	3	3	3	3	3	3	3	3	4	3	4	3	4	3	3	3	4
2	3	4	3	3	4	3	3	3	3	4	3	3	4	3	3	3	3	3	4	4	4	3
3	3	3	4	4	4	4	3	3	3	3	3	3	3	4	3	3	3	4	4	3	3	4
4	3	3	4	2	3	3	3	3	4	3	3	3	4	3	4	3	3	4	4	3	4	4
5	3	4	4	3	4	3	4	3	3	4	4	3	3	4	3	3	3	4	3	3	3	3
6	3	3	4	3	3	4	4	4	4	4	4	4	3	4	3	3	3	3	3	4	4	4
7	3	3	3	3	4	4	4	3	4	3	3	4	3	3	3	3	3	3	3	4	3	3
8	3	3	3	3	3	4	3	4	4	4	4	4	3	4	4	3	3	3	4	3	4	3
9	3	3	3	4	3	4	4	4	4	4	4	3	3	3	4	3	4	3	3	3	3	3
10	3	4	3	3	4	4	3	4	4	4	4	4	3	4	4	3	4	4	3	3	4	4
11	3	3	3	3	4	4	3	4	4	4	4	4	4	4	3	3	3	3	3	3	3	3
12	3	3	3	3	3	4	4	4	3	4	4	4	3	4	4	3	4	3	3	3	4	4
13	3	4	3	4	3	3	3	3	3	3	4	3	4	3	4	3	3	3	4	3	3	3
14	4	3	4	3	4	4	3	4	3	4	4	4	3	4	3	3	3	3	4	4	3	3
15	3	3	3	4	3	3	3	4	4	4	3	4	4	3	4	3	3	3	3	3	3	3
16	4	3	3	3	3	3	3	3	3	3	3	3	3	3	3	4	3	3	4	3	3	4
17	3	3	3	3	3	3	3	3	4	4	3	4	3	3	3	3	4	3	3	3	3	3
18	4	3	4	4	4	3	3	3	3	4	3	3	3	3	3	3	3	4	3	4	3	4
19	3	4	4	4	3	3	3	4	3	3	3	3	4	4	3	4	3	3	4	3	3	3
20	3	4	3	3	3	4	4	3	3	3	3	3	3	4	3	3	3	4	3	4	3	3
21	3	4	3	4	3	4	3	4	3	4	3	4	3	3	3	3	3	3	3	3	4	4
22	4	3	4	4	3	4	3	3	3	4	3	4	3	3	3	4	3	4	3	3	4	4

Table 4.7: Undesired hits limits for HCtCXs with $N_h = 22$

$m = 2 \Rightarrow \alpha = 2$

code	1	2	3	4	5	6	7	8	9	10	11
1	6	6	6	5	5	6	7	5	6	6	6
2	6	6	6	5	7	5	6	7	6	5	5
3	6	6	8	6	6	5	6	6	5	6	6
4	5	5	6	6	6	6	7	6	6	6	5
5	5	7	6	6	6	6	6	6	6	6	7
6	6	5	5	6	6	8	5	5	5	6	5
7	7	6	6	7	6	5	8	7	7	7	6
8	5	7	6	6	6	5	7	7	5	6	6
9	6	6	5	6	6	5	7	5	6	5	6
10	6	5	6	6	6	6	7	6	5	6	6
11	6	5	6	5	7	5	6	6	6	6	6

$m = 3 \Rightarrow \alpha = 1.5$

code	1	2	3	4	5	6	7
1	8	8	10	8	8	8	7
2	8	9	7	7	8	7	8
3	10	7	8	8	7	7	7
4	8	7	8	10	7	9	8
5	8	8	7	7	8	8	7
6	8	7	7	9	8	9	8
7	7	8	7	8	7	8	8

$m = 4 \Rightarrow \alpha = 1.33$

code	1	2	3	4	5
1	11	12	11	8	10
2	12	10	9	10	10
3	11	9	9	10	10
4	8	10	10	10	10
5	10	10	10	10	10

$m = 5 \Rightarrow \alpha = 1.125$

code	1	2	3	4
1	12	11	10	11
2	11	13	13	11
3	10	13	11	11
4	11	11	11	12

$m = 6 \Rightarrow \alpha = 1.1$

code	1	2	3
1	14	12	15
2	12	15	12
3	15	12	13

$m = 7 \Rightarrow \alpha = 0.92$

code	1	2	3
1	14	14	15
2	14	15	14
3	15	14	15

$m = 8 \Rightarrow \alpha = 0.93$

code	1	2
1	17	16
2	16	16

$m = 9 \Rightarrow \alpha = 0.875$

code	1	2
1	17	18
2	18	18

$m = 10 \Rightarrow \alpha = 0.83$

code	1	2
1	18	19
2	19	19

$m = 11 \Rightarrow \alpha = 0.85$

code	1	2
1	19	21
2	21	19

Table 4.8: Undesired hits limits for HCtCXs with $N_h = 52, m = 3$

$m = 3 \Rightarrow \alpha = 2.25$

code	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	10	9	11	10	10	8	8	9	10	10	10	9	9	10	9	13	9
2	9	12	9	12	8	9	9	11	11	9	8	10	9	11	10	9	10
3	11	9	10	9	10	9	10	10	10	10	8	8	9	9	8	8	10
4	10	12	9	10	10	9	9	10	10	9	9	10	9	9	10	9	11
5	10	8	10	10	10	9	10	9	10	9	10	12	9	10	9	10	9
6	8	9	9	9	9	12	10	9	9	8	9	10	9	9	11	9	10
7	8	9	10	9	10	10	10	9	8	10	8	9	8	10	10	8	9
8	9	11	10	10	9	9	9	11	9	10	9	9	10	9	9	9	9
9	10	11	10	10	10	9	8	9	11	13	9	10	8	10	10	9	11
10	10	9	10	9	9	8	10	10	13	12	9	9	10	9	10	11	10
11	10	8	8	9	10	9	8	9	9	9	12	9	9	8	10	9	10
12	9	10	8	10	12	10	9	9	10	9	9	9	10	9	8	10	9
13	9	9	9	9	9	9	8	10	8	10	9	10	10	10	9	10	10
14	10	11	9	9	10	9	10	9	10	9	8	9	10	10	10	9	9
15	9	10	8	10	9	11	10	9	10	10	10	8	9	10	10	9	10
16	13	9	8	9	10	9	8	9	9	11	9	10	10	9	9	10	10
17	9	10	10	11	9	10	9	9	11	10	10	9	10	9	10	10	9

$m = 4 \Rightarrow \alpha = 1.833$

code	1	2	3	4	5	6	7	8	9	10	11	12	13
1	12	12	12	11	11	10	11	11	12	11	13	11	11
2	12	11	11	12	11	11	12	11	12	11	13	10	13
3	12	11	12	13	11	11	12	12	13	12	10	11	10
4	11	12	13	12	11	11	10	13	10	11	11	13	11
5	11	11	11	11	12	12	11	12	11	12	11	12	10
6	10	11	11	11	12	11	13	12	11	13	12	12	10
7	11	12	12	10	11	13	11	13	11	11	13	12	12
8	11	11	12	13	12	12	13	12	11	12	11	13	10
9	12	12	13	10	11	11	11	11	13	11	10	12	11
10	11	11	12	11	12	13	11	12	11	12	12	11	10
11	13	13	10	11	11	12	13	11	10	12	13	12	12
12	11	10	11	13	12	12	12	13	12	11	12	15	15
13	11	13	10	11	10	10	12	10	11	10	12	15	12

Table 4.9: Undesired hits limits for HCtCXs with $N_h = 52, m = 5\ 6\ 7$

$m = 5 \Rightarrow \alpha = 1.5$

code	1	2	3	4	5	6	7	8	9	10
1	14	15	12	14	14	13	13	14	14	13
2	15	15	13	13	13	13	13	14	13	13
3	12	13	14	13	13	12	14	12	14	13
4	14	13	13	14	13	15	13	13	13	15
5	14	13	13	13	14	12	12	12	13	14
6	13	13	12	15	12	15	13	15	14	13
7	13	13	14	13	12	13	13	13	14	14
8	14	14	12	13	12	15	13	15	14	14
9	14	13	14	13	13	14	14	14	14	13
10	13	13	13	15	14	13	14	14	13	16

$m = 6 \Rightarrow \alpha = 1.5$

code	1	2	3	4	5	6	7	8
1	15	16	16	14	15	16	14	14
2	16	19	14	15	14	14	15	16
3	16	14	18	17	15	14	14	16
4	14	15	17	17	16	14	14	16
5	15	14	15	16	14	15	15	14
6	16	14	14	14	15	14	14	15
7	14	15	14	14	15	14	16	13
8	14	16	16	16	14	15	13	15

$m = 7 \Rightarrow \alpha = 1.25$

code	1	2	3	4	5	6	7
1	19	16	17	15	19	15	15
2	16	17	16	15	17	17	15
3	17	16	18	16	17	15	16
4	15	15	16	16	16	16	17
5	19	17	17	16	16	19	15
6	15	17	15	16	19	16	16
7	15	15	16	17	15	16	16

Table 4.10: Undesired hits limits for HCtCXs with $N_h = 52, m = 8\ 9$

$m = 8 \Rightarrow \alpha = 1.142$

code	1	2	3	4	5	6
1	20	19	19	18	18	20
2	19	19	19	18	18	17
3	19	19	20	19	18	19
4	18	18	19	19	19	19
5	18	18	18	19	18	17
6	20	17	19	19	17	18

$m = 9 \Rightarrow \alpha = 1.1875$

code	1	2	3	4	5
1	20	23	20	20	18
2	23	21	18	19	20
3	20	18	21	19	19
4	20	19	19	20	21
5	18	20	19	21	22

Table 4.11: Undesired hits limits for HCtCXs with $N_h = 52, m = 10\ 15$

$m = 10 \Rightarrow \alpha = 1.055$

code	1	2	3	4	5
1	21	23	20	22	23
2	23	22	20	20	20
3	20	20	22	20	21
4	22	20	20	22	22
5	23	20	21	22	23

$m = 15 \Rightarrow \alpha = 1.035$

code	1	2	3	4	5
1	21	23	20	22	23
2	23	22	20	20	20
3	20	20	22	20	21
4	22	20	20	22	22
5	23	20	21	22	23

Table 4.12: Undesired hits limits for HCtCXs with $N_h = 52, m = 17\ 20\ 25\ 26$

$m = 17 \Rightarrow \alpha = 0.937$

code	1	2	3
1	34	29	29
2	29	34	33
3	29	33	30

$m = 20 \Rightarrow \alpha = 0.89$

code	1	2
1	38	36
2	36	36

$m = 25 \Rightarrow \alpha = 0.813$

code	1	2
1	42	43
2	43	40

$m = 26 \Rightarrow \alpha = 0.80$

code	1	2
1	42	44
2	44	42

Table 4.13: Undesired hits limits for HCCXs with $N_h = 100, m = 5$

$m = 5 \Rightarrow \alpha = 1.875$

HCCXs	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	17	14	15	14	16	15	14	13	15	15	14	14	15	15	15	14	14	14	14	14
2	14	14	15	14	15	15	17	14	15	13	16	15	13	15	16	14	15	17	15	17
3	15	15	17	14	16	14	15	14	14	14	14	13	15	17	14	14	14	13	15	14
4	14	14	14	17	15	15	14	14	14	14	15	14	14	14	13	14	13	15	15	14
5	16	15	16	15	16	15	14	15	14	15	15	15	14	14	14	15	13	13	14	15
6	15	15	14	15	15	17	16	14	14	13	15	14	14	16	14	15	14	15	15	16
7	14	17	15	14	14	16	15	16	14	15	15	14	14	15	15	14	13	15	15	14
8	13	14	14	14	15	14	16	15	15	16	14	14	13	14	15	14	14	14	16	14
9	15	15	14	14	14	14	14	15	17	16	14	14	14	15	15	14	14	14	14	14
10	15	13	14	14	15	13	15	16	16	17	15	14	15	16	14	15	14	14	15	14
11	14	16	14	15	15	15	15	14	14	15	14	14	14	14	14	14	14	14	14	15
12	14	15	13	14	15	14	14	14	14	14	14	17	16	16	15	14	14	13	14	15
13	15	13	15	14	14	14	14	13	14	15	14	16	15	15	14	14	15	14	14	15
14	15	15	17	14	14	16	15	14	15	16	14	16	15	16	15	14	15	14	15	14
15	15	16	14	13	14	14	15	15	15	14	14	15	14	15	16	14	15	15	17	15
16	14	14	14	14	15	15	14	14	14	15	14	14	14	14	14	15	14	15	14	15
17	14	15	14	13	13	14	13	14	14	14	14	14	15	15	15	14	17	14	13	15
18	14	17	13	15	13	15	15	14	14	14	14	13	14	14	15	15	14	17	14	15
19	14	15	15	15	14	15	15	16	14	15	14	14	14	15	17	14	13	14	14	14
20	14	17	14	14	15	16	14	14	14	14	15	15	15	14	15	15	15	15	14	19

Table 4.14: Undesired hits limits for HCCXs with $N_h = 100, m = 6$

	$m = 6 \Rightarrow \alpha = 1.6$															
$HCCXs$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	18	16	16	16	17	18	15	18	17	16	15	16	17	16	16	19
2	16	16	17	15	16	16	15	15	18	16	15	16	17	16	15	17
3	16	17	17	18	17	17	16	16	16	15	17	17	15	16	17	18
4	16	15	18	18	18	17	16	16	17	16	15	16	16	16	16	18
5	17	16	17	18	17	16	17	15	15	17	17	16	17	16	16	17
6	18	16	17	17	16	18	18	17	15	16	19	16	16	17	17	16
7	15	15	16	16	17	18	16	15	16	16	16	15	17	17	17	17
8	18	15	16	16	15	17	15	17	16	17	20	17	18	16	16	16
9	17	18	16	17	15	15	16	16	18	16	17	16	16	16	16	16
10	16	16	15	16	17	16	16	17	16	18	17	17	20	18	17	15
11	15	15	17	15	17	19	16	20	17	17	17	18	16	16	16	16
12	16	16	17	16	16	16	15	17	16	17	18	17	16	16	17	17
13	17	17	15	16	17	16	17	18	16	20	16	16	17	15	16	15
14	16	16	16	16	16	17	17	16	16	18	16	16	15	20	17	16
15	16	15	17	16	16	17	17	16	16	17	16	17	16	17	18	18
16	19	17	18	18	17	16	17	16	16	15	16	17	15	16	18	17

Table 4.15: Undesired hits limits for HCCXs with $N_h = 100, m = 7$

	$m = 7 \Rightarrow \alpha = 1.583$													
$HCCXs$	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	23	19	17	19	16	18	18	18	20	18	18	18	19	19
2	19	20	18	16	17	18	18	17	18	18	18	17	17	18
3	17	18	18	18	17	17	17	18	17	17	18	17	18	19
4	19	16	18	17	18	17	18	17	17	17	17	16	17	18
5	16	17	17	18	20	17	18	17	18	17	18	19	17	18
6	18	18	17	17	17	19	21	18	16	18	18	18	18	20
7	18	18	17	18	18	21	18	18	18	19	21	18	17	18
8	18	17	18	17	17	18	18	18	16	19	18	18	18	18
9	20	18	17	17	18	16	18	16	19	19	20	17	17	16
10	18	18	17	17	17	18	19	19	19	21	18	18	17	18
11	18	18	18	17	18	18	21	18	20	18	22	17	18	17
12	18	17	17	16	19	18	18	18	17	18	17	20	18	18
13	19	17	18	17	17	18	17	18	17	17	18	18	18	18
14	19	18	19	18	18	20	18	18	16	18	17	18	18	18

Table 4.16: Undesired hits limits for HCCXs with $N_h = 100, m = 8$

$m = 8 \Rightarrow \alpha = 1.428$

$HCCXs$	1	2	3	4	5	6	7	8	9	10	11	12
1	20	20	21	19	22	20	20	22	19	22	20	20
2	20	22	20	19	19	19	20	21	19	19	19	18
3	21	20	23	19	19	19	20	18	21	20	20	21
4	19	19	19	21	21	21	20	19	19	19	20	20
5	22	19	19	21	22	19	23	19	20	21	21	19
6	20	19	19	21	19	22	20	19	20	20	20	22
7	20	20	20	20	23	20	20	21	20	21	19	23
8	22	21	18	19	19	19	21	20	21	20	20	19
9	19	19	21	19	20	20	20	21	20	23	19	19
10	22	19	20	19	21	20	21	20	23	24	20	20
11	20	19	20	20	21	20	19	20	19	20	20	19
12	20	18	21	20	19	22	23	19	19	20	19	20

Table 4.17: Undesired hits limits for HCtCXs with $N_h = 100, m = 9$

$m = 9 \Rightarrow \alpha = 1.375$

$code$	1	2	3	4	5	6	7	8	9	10	11
1	26	21	22	22	21	20	20	21	23	24	20
2	21	20	21	23	22	25	21	23	21	21	21
3	22	21	25	21	23	21	21	20	22	20	22
4	22	23	21	22	21	22	21	23	21	21	22
5	21	22	23	21	24	21	22	24	21	20	20
6	20	25	21	22	21	21	21	24	20	20	24
7	20	21	21	21	22	21	21	20	22	21	23
8	21	23	20	23	24	24	20	23	22	20	22
9	23	21	22	21	21	20	22	22	21	20	21
10	24	21	20	21	20	20	21	20	20	21	21
11	20	21	22	22	20	24	23	22	21	21	21

Table 4.18: Undesired hits limits for HCtCXs with $N_h = 100, m = 10$

	$m = 10 \Rightarrow \alpha = 1.222$									
code	1	2	3	4	5	6	7	8	9	10
1	23	24	23	23	22	23	23	24	23	22
2	24	26	22	23	23	25	23	23	23	25
3	23	22	24	24	24	24	23	24	23	23
4	23	23	24	25	23	25	22	23	23	22
5	22	23	24	23	25	25	24	23	25	25
6	23	25	24	25	25	26	23	24	26	21
7	23	23	23	22	24	23	26	22	23	22
8	24	23	24	23	23	24	22	24	24	24
9	23	23	23	23	25	26	23	24	24	23
10	22	25	23	22	25	21	22	24	23	24

Table 4.19: Undesired hits limits for HCtCXs with $N_h = 100, m = 11$

	$m = 11 \Rightarrow \alpha = 1.15$								
code	1	2	3	4	5	6	7	8	9
1	26	24	24	25	24	23	25	23	26
2	24	26	26	24	24	25	24	23	25
3	24	26	27	23	27	25	23	26	26
4	25	24	23	26	24	27	23	24	25
5	24	24	27	24	24	23	25	26	25
6	23	25	25	27	23	26	25	24	25
7	25	24	23	23	25	25	26	23	26
8	23	23	26	24	26	24	23	27	25
9	26	25	26	25	25	25	26	25	26

Table 4.20: Observed α for different N_h, m

$N_h \searrow m$	2	3	4	5	6	7	8	9	10	11
10	1	1	0.83	0.625	∅	∅	∅	∅	∅	∅
22	2	1.5	1.33	1.125	1.1	0.92	0.93	0.875	0.83	0.85
52	2.5	2.25	1.833	1.5	1.5	1.25	1.142	1.1875	1.055	1.05
100	3.5	2.5	2.167	1.875	1.6	1.583	1.428	1.375	1.222	1.15

4.3.3 Pseudorandom Codes (PNCs)

These codes are constructed as follows: for a given N_h, the N_p code symbols are obtained by casually extracting N_p numbers between 0 and $N_h - 1$. If we indicate the random extraction of a number in the range $[a, b]$ with the function $rand(a, b)$, the placement operator is:

$$y_{PNC}(k) = rand(0, N_h - 1).$$

For construction, these codes possess the maximum randomness nature, thus emerging as an optimum choice as UWB power spectrum whiteners. The crucial negative aspect is that they are not addressable at all and for this reason they can only be used as a paragon family for other codes artificially engineered. They will have a great role in the performance measures of codes, like the HCCs and the HCCXs, against which we will study cross-correlation properties. PNCs retain the maximum flexibility because N_h and N_p are not associated by any formula.

A possible variant can be implemented with the goal of obtaining a full code. To generate these codes it suffices to perform an extraction experiment without reinsertion of the extracted element in the set of those available for the successive extractions. Since the extractions are repeated until the available population is the null set they have $N_h = N_p$ (as expected from full codes) and the flexibility is reduced.

4.3.4 Informatically generated PN Codes (iPNCs)

This section introduces a possible structured generation of codes which have a behavior similar to the PNCs presented afore. The algorithm illustrated[6] has been developed in [17].

The input elements of a computational block implementing the algorithm are constituted by:

- a binary string called the *seed* of length L_s;
- an integer k in the range $[0; L_s - 1]$.

The output produced is:

- a code with a pseudorandom behavior.

The algorithm proceeds along the following steps:

- Initial values: $L_s = \mathsf{strlen}(\mathit{seed})$, $\mathsf{k{=}rand}(0, L_s - 1)$, $N_b = \log_2 N_h$, $i = 0$.
- Iteration:
  1. we point at the $(k+i)^{th}$ bit of the binary string *seed*. Starting from this position we consider the adjacent N_b bits. Their value read in a decimal form coincides with $y(i)$ where $y(...)$ is the placement operator.[7]

[6] The algorithm is covered by an Italian patent

[7] See page 67 for a definition of *placement operator*.

2. while $i < N_p$ we increment i as follows $i{+}{+}$, and repeat the iteration.

In this way the placement operator produces values in the range $[0, (N_h - 1)]$. This algorithm allows to exactly reproduce the code from the knowledge of its inputs. As much as a telecommunication system with two partners exchanging informations (for example a MT and an AP) is concerned, the *seed* could be a string composed by the juxtaposition of two binary strings associated to the identities of the source and the destination devices. These strings could be the IPv4 or the IPv6 or simply the MAC addresses of the communicating terminals.

A possible variation of the previous algorithm can be useful in the case in which the addresses of the devices considered are very similar with each other.[8] In the algorithm described afore a degree of randomness is guaranteed only by the starting point k in the first step; if the bit sequences are similar this might not be sufficient. The algorithm could be modified as follows in order to improve its randomness:

- Initial values: $L_s = \mathsf{strlen}(\mathit{seed})$, $\mathsf{k{=}rand}(0, L_s - 1)$, $N_b = \log_2 N_h$, $i = 0$, $N_u = L_s$[9].

- Iteration:

  1. we point at the not used yet $((k + i * k) \mathsf{mod}_{N_u})^{th}$ bit from the origin

[8] This happens, for example, if the devices belong to a small IPv4 LAN. These addresses share the same Net-Mask and differ only in the last digits.

[9] N_u is the number of unused binary digits of the *seed*.

of the binary string *seed*. Starting from this position we consider the adjacent N_b bits. Their value read in a decimal form coincides with $y(i)$ where $y(...)$ is the placement operator.

2. we decrement the number of unused binary digits in this way[10]:

$$N_u = (N_u - 1)\mathsf{mod}_{L_s}$$

and if $N_u = L_s$ then consider all the bits unused.

3. while $i < N_p$ we increment i as follows $i{+}{+}$, and repeat the iteration.

Supposing that the *seed* is obtained with the following operation[11]:

$$seed = source\ address + destination\ address$$

and observing that a bidirectional channel can be thought about as a couple of opposite unidirectional channels in which the roles of the source and the destination are reciprocally exchanged, it can be concluded that:

Conclusion 4.3.3. *when two communicating partners A and B exchange informations in the directions*

$$A \longrightarrow B \ and \ A \longleftarrow B,$$

by associating to one verse of the communication the:

$$seedAB = address\ A + address\ B$$

[10]The following operation maintains the value of N_u always in the range $[0, L_s - 1]$ starting again the count when all the bits have been already used at least one time.

[11]Here the + symbol stands for the juxtaposition operator.

and to the other the:

$$seedBA = address\ B + address\ A$$

with the introduced algorithms[12] *it is possible to generate univocally two codes, by the further*[13] *exchange of a single integer k that must be transmitted together with the partners' addresses.*

4.4 HCCs and HCCXs' interactions vs PNCs

The figure 4.10 illustrates the maximum hit number between all the HCCs which can be generated with $N_h = 100$ and thirty different PNCs randomly generated. Each HCC is distinguished through the value of the parameter a in the placement operator described in definition 4.1.3. This value is reported on the x-axis while the y-axis addresses the different PNCs. The z-axis shows the maximum number of hits (normalized to the peak of the HCCs auto-hit arrays) in the respective cross-hit arrays. In the worst case the interference reaches a magnitude which is about the 9% of the peak. In figure 4.6 this value (0.09) is confronted against the peak of the HCC's auto-hit array and against the mutual interference among HCCs. It can be seen that the 9%

[12] In this situation the *seedAB* and the *seedBA* are pretty similar because they are composed by different juxtapositions of the same strings and perhaps the variant can be more efficient.

[13] In each telecommunication application the source and destination addresses are already foreseen and the incremental information required by this method is only the key k.

is a low percentage which allow good performances when a single PNC is superimposed on the HCC.

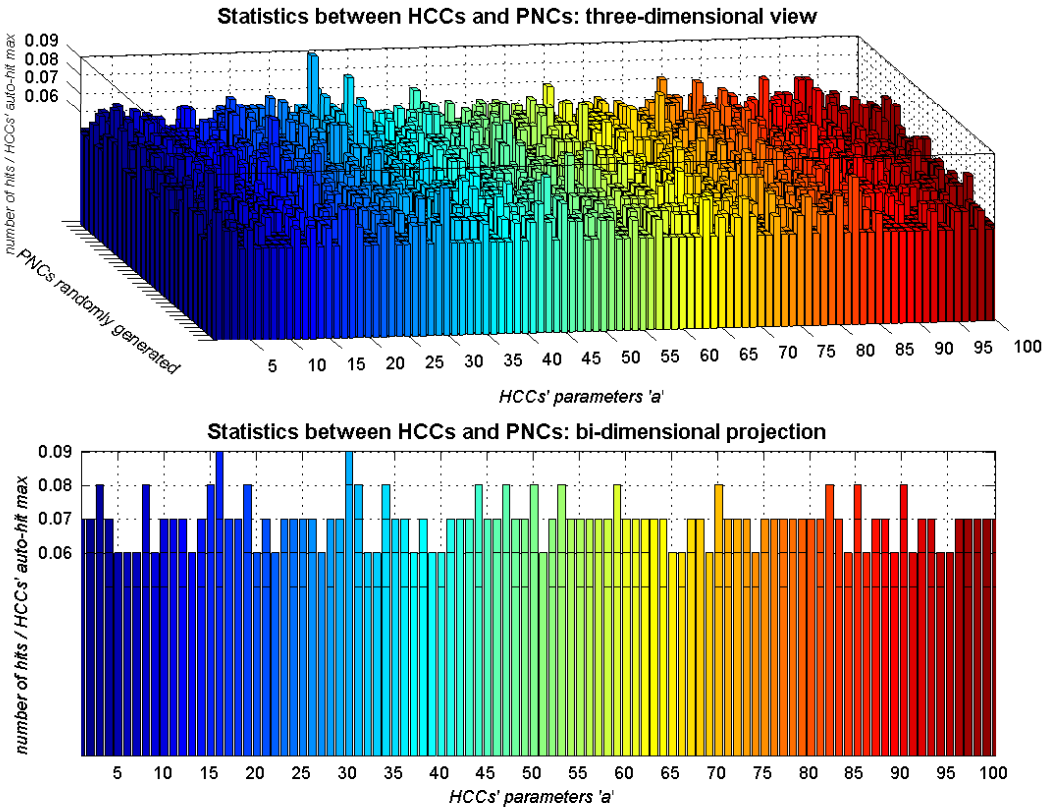

Figure 4.10: HCCs hit limits against 30 PNCs randomly generated.

Everything changes when more than one PNC coexist in the environment in which the HCCXs are transmitted as time hopping codes (HCtCXs). Once again it is important that the peak obtained by correlating the local reproduction of the transmitted HCtCX with the corrupted sequence might be easily resolvable. In figure 4.11 three different plots are displayed:

1. *high plot:* it represents the transmitted sequence HCtCX with $N_h =$

$100, m = 5$ and a period of repetition equal to $2 \cdot N_p$. The horizontal axis is expressed in T_c units and a value of 1 is reported on the vertical axis only if the relative T_c is occupied by a pulse. Since the number of pulses transmitted is very high they are not singularly discernable.

2. *mid plot:* it represents the transmitted sequence when it is corrupted by the superimposition of fifty different PNCs. This time the transmitted HCtCX seems to be completely hidden by the action of the interferers.

3. *low plot:* it represents the correlation between the transmitted HCtCX and the corrupted sequence above.

In plot 3 two peaks are observed at the trailing edges of the transmitted HCtCXs ($5 \cdot 10^4\ T_c = 0.5 \cdot 10^5\ T_c$). Since both peaks trespass the mean value by a difference of about 500 pulses, they can be isolated from the disturbed sequence. If the HCtCX is used as a PSC, then the positions of the peaks allow to reach the L-slot synchronization. The same experiment has been conducted with $100, 200, 300$ and 400 interferers. The results are displayed in the figures 4.12, 4.13, 4.14 and 4.15. The figure 4.16 is useful to perform a comparison between these results. To simplify the analysis the three plots reported here use the same scale on the vertical axis and it is clear that the peaks observed at the trailing edges of the transmitted HCtCX trespass the respective means by a difference which has a similar order of magnitude in all the presented cases. It can be concluded that the interference provoked by

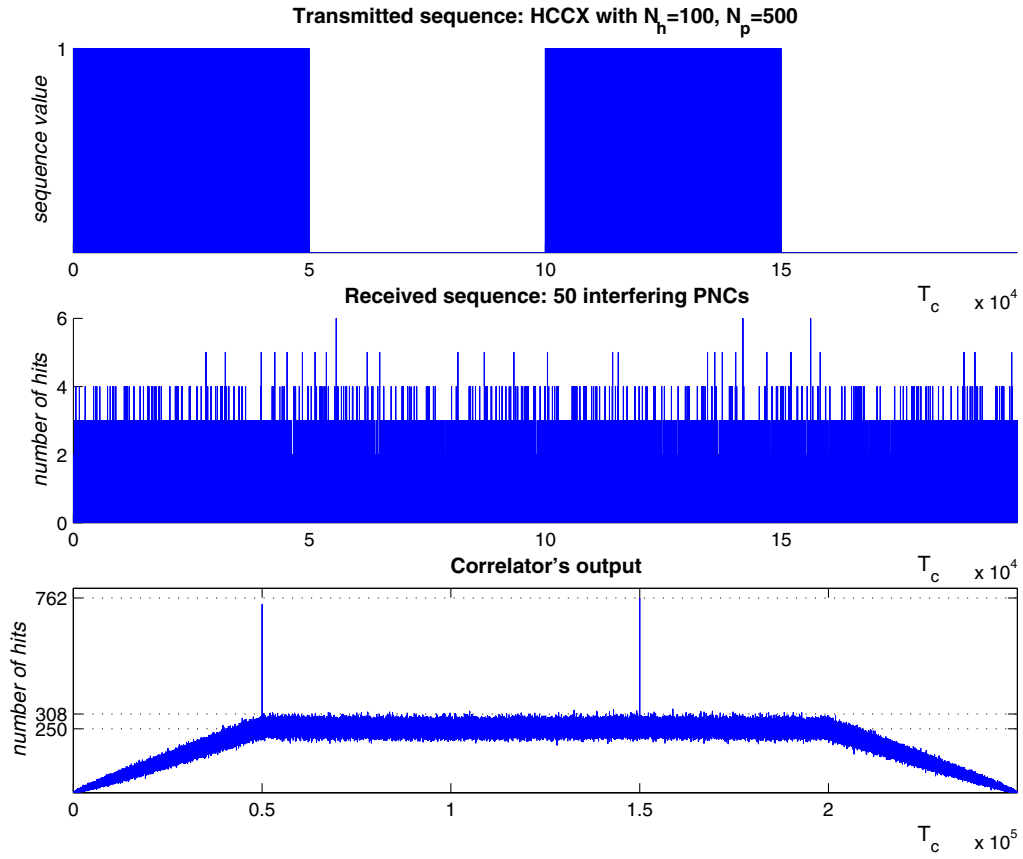

Figure 4.11: HCtCX transmitted in an environment disturbed with 50 PN interferers.

the PNCs is twofold. It is destructive in the way that it introduces pulses in undesired positions, and it is constructive because it may add pulses in the time chips already occupied by the transmitted HCtCX. Since the pulses sum themselves in energy, collisions with the transmitted HCtCX are beneficial to reception. As a result, the main effect of PNCs' interference is to bias the energy received to an higher mean value (which is directly proportional to the number of interferers), this consequence can be compensated with the aid of an adaptive threshold decisor which operates behind the receiving correlator.

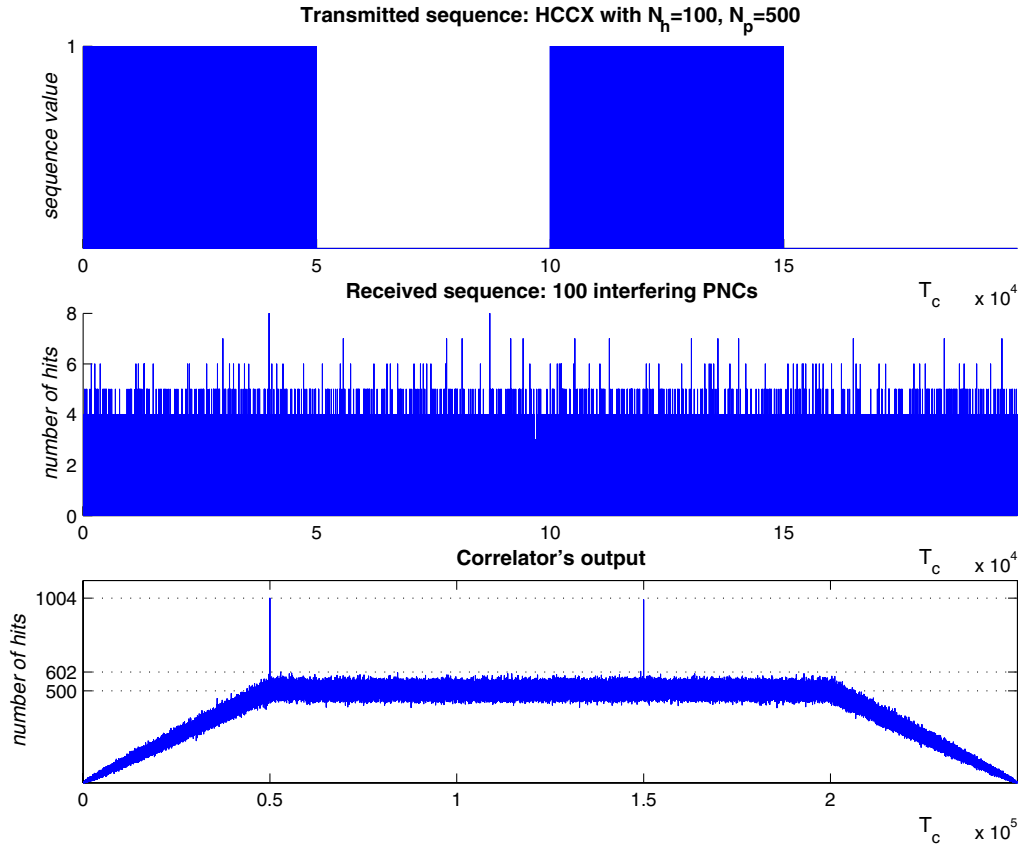

Figure 4.12: HCtCX transmitted in an environment disturbed with 100 PN interferers.

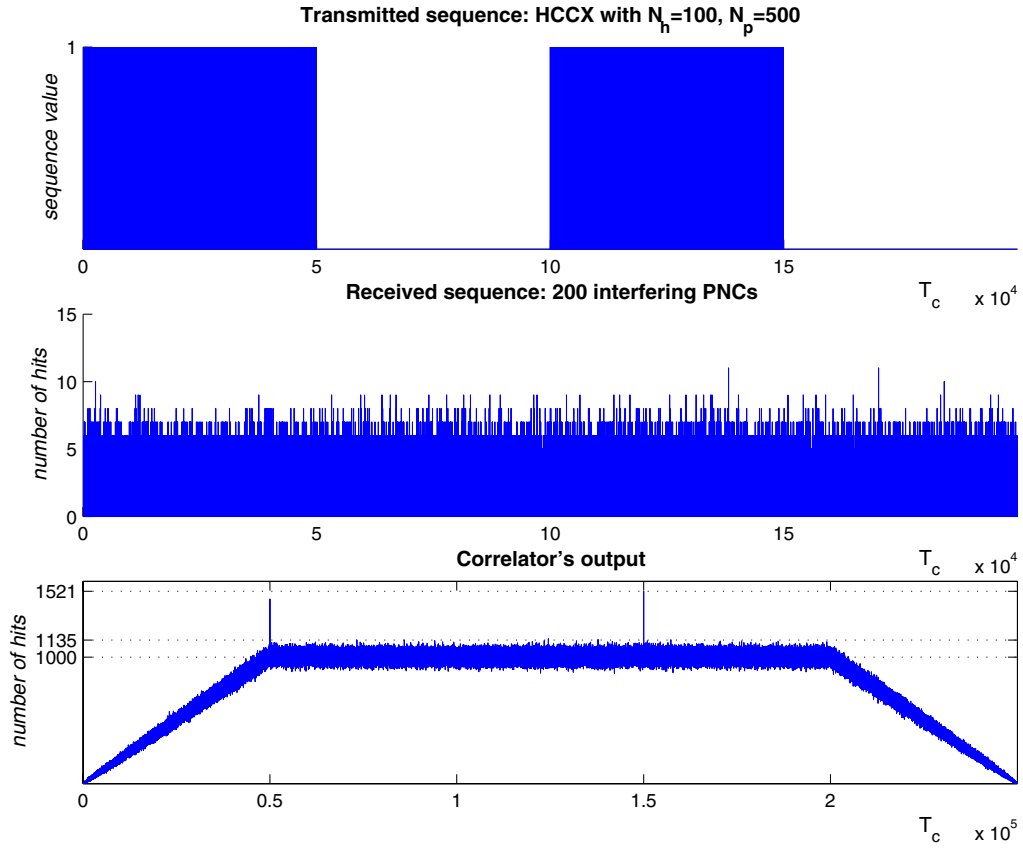

Figure 4.13: HCtCX transmitted in an environment disturbed with 200 PN interferers.

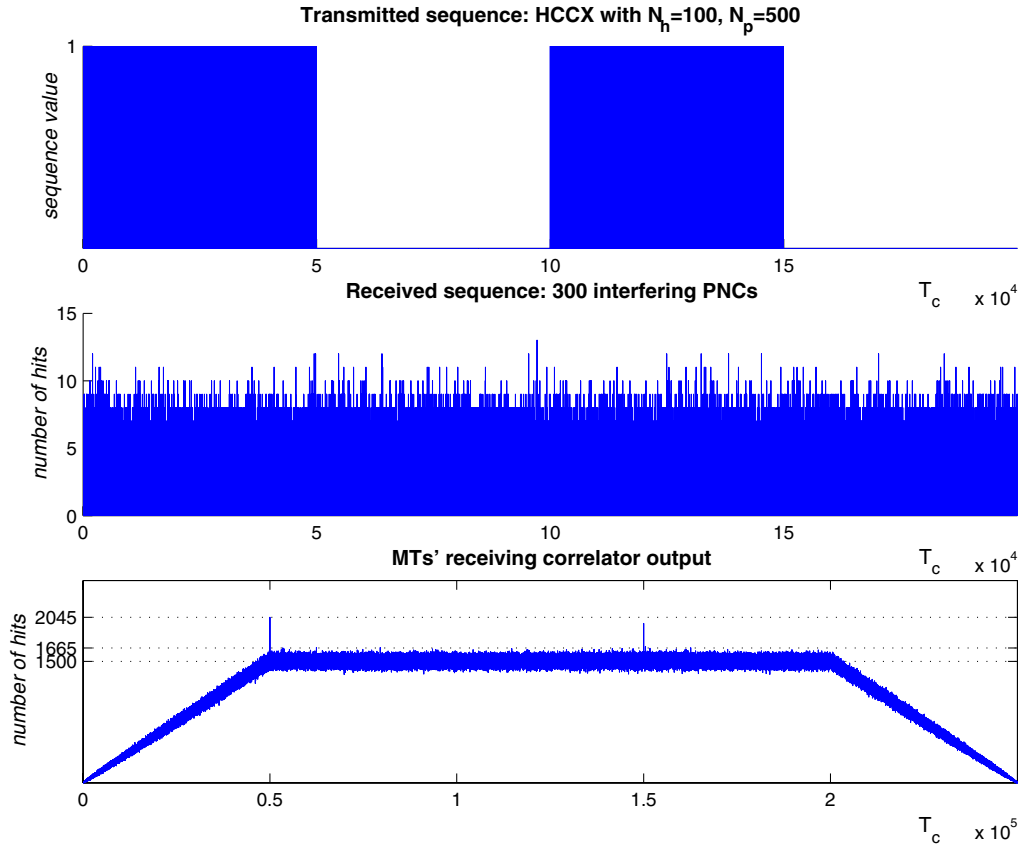

Figure 4.14: HCtCX transmitted in an environment disturbed with 300 PN interferers.

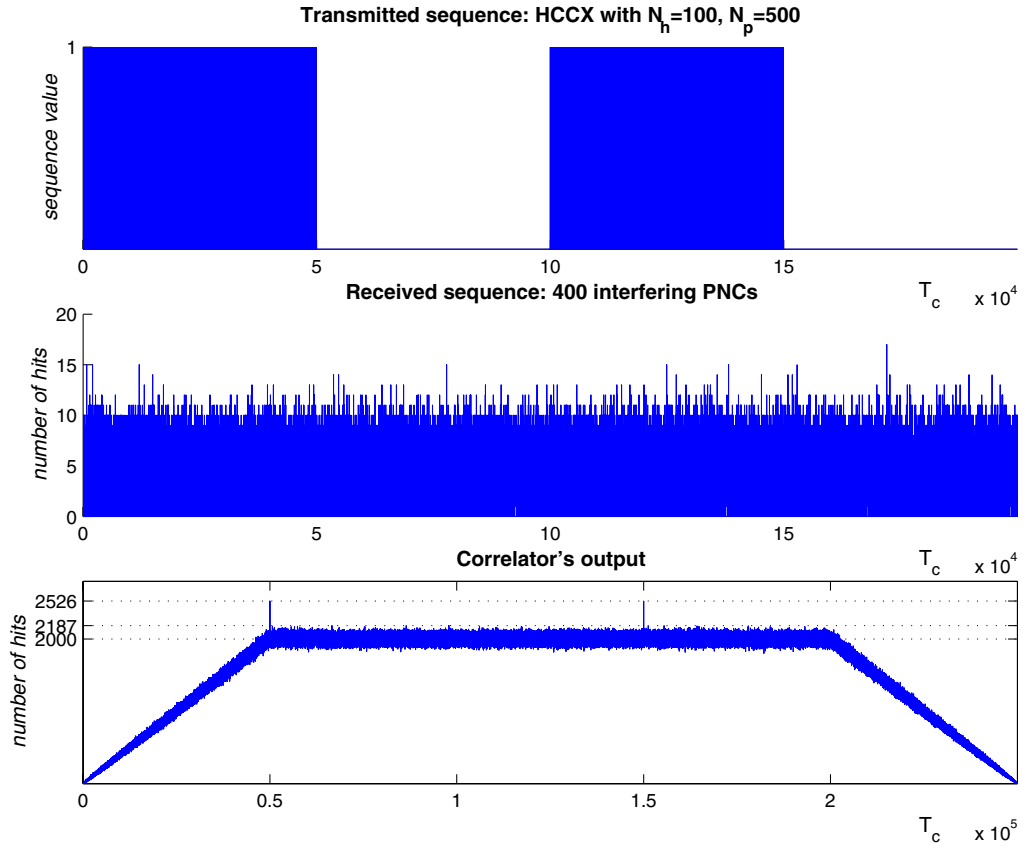

Figure 4.15: HCtCX transmitted in an environment disturbed with 400 PN interferers.

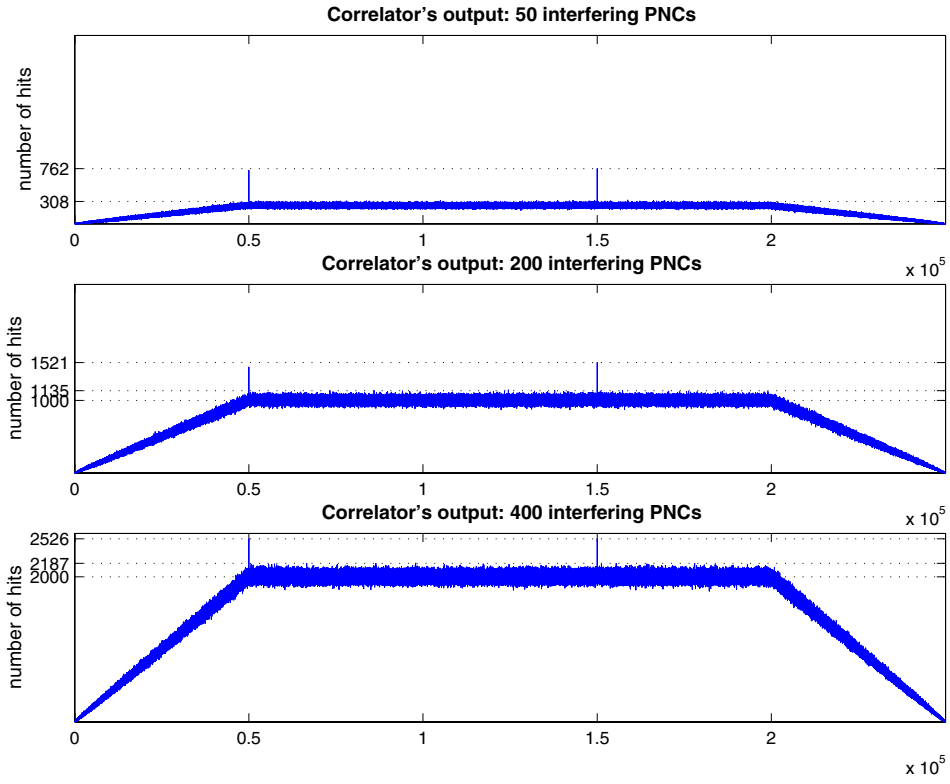

Figure 4.16: HCtCX transmission: comparison between the peaks obtained at the receiver with different levels of interference.

Part III

Conclusions

5

Codes

The auto and cross-correlation properties illustrated in sections 4.3.1, 4.3.2, 4.4, promote the HCCs has good candidates for PSC and SSCs. The main disadvantage that these codes present is the bond $N_h = N_p$ which is overcome by the extension suggested with the HCCXs. The HCCXs maintain a suffi-

cient level of orthogonality which is increased by augmenting the multiplicity m. It has been demonstrated that the parameter m can relevantly improve the perceived orthogonality since longer sequences result in higher peaks in the origin of the auto-correlation and in lower UHs levels, i.e they produce lower ratios R. When the down-link acquisition is projected a decision is taken about the length M of the CPC block and about the dimension q of the CPCs' alphabet adopted. The parameter m must be chosen only after having decided the number of hops N_h in a frame T_f, which determines the cardinality of the set of available codes, and in accordance with the constrain of producing at least $q + 1$ HCCXs. Of these HCCXs, q codes are employed as SSCs while the last one is adopted as a PSC. A further possibility is to prepare the network to the coexistence of different sub-networks by reserving a number β of HCCXs for distinct PSCs. All the APs belonging to the same sub-network must adopt a unique PSC among those available. In this case the parameter m is chosen with the constrain of producing at least $q + \beta$ HCCXs where q is decided as a function of the number of BCHs required. If correct dimensioning is performed, the HCCXs are almost perfect for PSC and SSCs. The good level of orthogonality within the same family allows to obtain a precise temporal synchronization. Since traffic channels can be considered as PNCs, the good behavior manifested by the HCCXs against PNCs allows to limit the inter-channel interference.

6

Link Acquisition protocol

In chapter 3 it has been developed the first step to be performed in order to complete the link acquisition in a mobile wireless network with several APs mastering the communications between the MTs. That procedure has been called down link acquisition and allows to reach the temporal synchro-

nization between a MT and an AP as well as the acquisition of the BCH. Both objectives are of fundamental importance but are not sufficient if the expected goal is to establish a bidirectional channel. Since a bidirectional communication implies that both partners are aware of each other, the down link acquisition through cell search must be completed with further actions aimed at the identification of the MT by the AP. Only after completion of this procedure it is obtained the access to the network, and a link is considered acquired.

As told in chapter 3 the BCH contains all the information about the common channels reserved to the AP which is mastering the cell. When a MT is able to listen to the BCH, it can collect all the references to the ph-CHs adopted by the AP and has the possibility to use some of them in order to transfer the control information necessary to identify itself. Among the others, the AP will surely possess a common up-link channel called Random Access CHannel (RACH). This channel will have the same framed structure of the BCH, it will have the same temporization, and will multiplex a number of equivalent l-CHs which are not reserved to any particular communication. All these l-CHs can be randomly occupied by any MT which needs to register itself in the AP. When a MT wants to establish a communication it transmits in the first free l-CH of the RACH a package which contains at least the following data:

- its identity (MT-id), which can be a MAC address for example;
- an Equipment Class Identifier (EC-id), which can be useful in heterogeneous networks where the AP behavior might depend on the class of the MT;
- the Security and Accounting information whose function is to avoid the unauthorized access to the network or the use of some services which are not allowed.

The previous list is not complete but it is useful in order to offer an idea of the contents of the transmitted package. When the AP receives the request on the RACH, it activates the identification procedures and, if the authorization is obtained, it sends a response package through a common down-link channel called Access Grant CHannel (AGCH) which has the same framed structure of the RACH. The response package contains at least the following data:

- the recipient MT-id;
- the addresses of some ph-CHs or l-CHs which the AP has reserved for the dedicated communications with the MT.

As usual, the previous list is not complete since more and more data could be inserted in the response package. This aspect of the project is not considered here as it would deserve a dedicated report over the structures of the packages transmitted between the MTs and the APs as well as over the types, and

the required transmission rates of the l-CHs adopted. When the procedure suggested in the link acquisition protocol is completed, the MT and the AP have established the bidirectional communication which was the objective of Link Acquisition.

Bibliography

[1] Robert A. Scholtz. Multiple access with time-hopping impulse modulation. In *MILCOM93*. IEEE, October 1993.

[2] Robert A. Scholtz. Impulse radio: How it works. In *IEEE Communications Letters*, volume 2, pages 36–38. IEEE, February 1998.

[3] Moe Z. Win and Robert A. Scholtz. Ultra-wide bandwidth time-hopping spread-spectrum impulse radio for wireless multiple-access communications. In *Transactions on communications*, volume 48, pages 679–691. IEEE, April 2000.

[4] Moe Z. Win. Spectral density of random time-hopping spread-spectrum UWB signals with uniform timing jitter. In *Military Communications Conference Proceedings*, volume 2, pages 1196–1200. MILCOM IEEE, 1999.

[5] J.M. Wozencraft and I.M. Jacobs. *Principles of Communication Engineering*. London, U.K.: Wiley, 1st edition, 1965.

[6] M.K. Simon, S.M. Hinedi, and W.C. Lindsey. *Digital Communication Techniques: Signal Design and detection*. Englewood Cliffs, NJ: Prentice Hall, 1st edition, 1995.

[7] J.G. Proakis. *Digital Communications*. New York: McGraw Hill, 3rd edition, 1995.

[8] Tien-Yow Liu and Robert A. Scholtz. Link activation protocols for a mobile communication network with directive/adaptive antennas. In *Transactions on communications*, volume 48, pages 60–74. IEEE, January 2000.

[9] E.C. Posner and J.H. Rumsey. Continuous sequential decision in the presence of a finite number of hypotheses. In *Transactions on Information Theory*, volume IT-12, pages 248–255. IEEE, 1966.

[10] Robert Fleming and Cherie Kushner. Low-power, miniature, distributed position location and communication devices using ultra-wideband, non-sinusoidal communication technology. Technical report, Aether Wire & Location Inc., 1995.

[11] Andreas Polydoros and Charles L.Weber. A unified approach to serial search spread-spectrum code acquisition - part I: general theory. In

Transactions on communications, volume 32, pages 542–549. IEEE, May 1984.

[12] Kenichi Higuchi, Mamoru Sawahashi, and Fumiyuki Adachi. Fast cell search algorithm in inter-cell asynchronous ds-cdma mobile radio. In *IEICE Transactions on Communication*, volume E81-B, pages 1527–1534. IEICE, July 1998.

[13] Sundararajan Sriram and Srinath Hosur. Cyclically permutable codes for rapid acquisition in ds-cdma systems with asynchronous base stations. *IEEE Journal on selected areas in Communications*, 19:83–94, January 2001.

[14] Edward L.Titlebaum. Frequency- and time-hop coded signals for use in radar and sonar systems and multiple access communication systems. In *Conference record of the Twenty-Seventh Asilomar conference on Signals, systems and computers*, pages 1096–1100. IEEE, 1993.

[15] Tomaso Erseghe. Ultra wide band pulse communications. Master's thesis, Universitá degli studi di Padova, 1998/2001.

[16] Svetislav V.Marić and Edward L.Titlebaum. A class of frequency hop codes with nearly ideal characteristics for use in multiple-access spread-spectrum communications and radar and sonar systems. In *Transactions on communications*, volume 40, pages 1442–1447. IEEE, September 1992.

[17] Maria Gabriella Di Benedetto and Maria Stella Iacobucci. Metodo informatizzato di generazione di codici pseudo-ortogonali. deposited italian patent.