

Synchronization for Subsampling Digital UWB Receiver: a Holistic Approach

Yves Vanderperren*, Geert Leus[†], Wim Dehaene*

*EE Dept. (ESAT-MICAS), Katholieke Universiteit Leuven, Belgium

[†]Faculty of EE, Mathematics and CS, Delft University of Technology, The Netherlands

Email: {yves.vanderperren,wim.dehaene}@esat.kuleuven.be, g.leus@tudelft.nl

Abstract—This paper evaluates in AWGN and dense multipath environments several time acquisition algorithms for digital pulsed UWB receivers sampling below Nyquist Rate for high speed communication. As UWB systems imply severe implementation challenges, we perform an interdisciplinary analysis of the design space by associating the system level assessment with an implementation perspective. Several architectures of the proposed algorithms have been rapidly prototyped on FPGA using a state-of-the-art design flow which can be extended to ASIC design.

I. INTRODUCTION

The design of pulsed UWB receivers presents unique challenges along the complete design flow, ranging from system (algorithmic) level to practical implementation aspects. The need to capture efficiently the energy of the transmitted pulses which are spread by the UWB channel, accurate synchronization, interference mitigation, are few examples of system design concerns which are exacerbated in the UWB context. Limited ADC sampling rates and ADC bit widths, dynamic and static (leakage) power consumption constraints, constitute typical implementation aspects which require dedicated attention to guarantee the successful deployment of UWB systems, in particular for power-constrained mobile devices.

Wireless systems have been traditionally developed following design flows where DSP engineers, system architects, hardware and software designers, develop the system with a single focus on their speciality. However, the peculiarities of UWB technology motivate innovative holistic design approaches which consider the system being developed as a whole, and take global optimizations opportunities into account. We integrate therefore the disciplines of system level and detailed design into a combined effort with the support of advanced prototyping techniques, and apply in this paper our approach to the study of the synchronization of a subsampling UWB receiver.

Subsampling techniques [1][2] provide a particularly attractive alternative to classical receivers in the context of time acquisition, as their search space is dramatically reduced and fast synchronization time can be achieved. In [2], a subsampling receiver was proposed which can achieve 100 Mbit/s at 10 m distance for signals occupying the 3.1–10.6 GHz band, with an ADC sampling below 1 GSamples/s. This paper presents the assessment of several signal synchronization acquisition algorithms for such receiver from a combined system- and implementation-level perspective.

This paper is structured as follows. After a quick overview of the types of synchronization (Section II) and the working principle of the subsampling receiver (Section III), several algorithms are presented in Section IV. Finally, their performance is evaluated (Section V) before the conclusions.

II. SYNCHRONIZATION TYPES

Time acquisition has been extensively studied for classical UWB receivers and includes *signal detection* and synchronization at *pulse*, *symbol* (or *code*), and *preamble* level [3][4].

The task of signal detection consists of the decision on the presence or absence of signal within a limit of time specified at a higher level in the communication protocol. The synchronization at preamble level is specific to data-aided acquisition systems and estimates where the preamble ends.

The synchronization at code or symbol level detects the symbol boundaries. By virtue of their autocorrelation properties, PN codes used in time-hopping (TH) or direct-sequence (DS) UWB systems enable the estimation of symbol delimitation, as only one code permutation maximizes the energy of the despread signal.

The synchronization at pulse level identifies the optimal boundaries of the periods constituting the symbols. In coherent receivers, where the incoming signal is correlated (matched filtered) with a locally generated template, pulse level synchronization algorithms determine the optimal alignment between the template and the received signal, typically by looking for the template position which maximizes the correlator output. These receivers can reach in theory the highest performance if perfect matched filtering is achieved and complete channel knowledge (including path- and antenna-dependent pulse distortion) is known before the synchronization. Such approach suffers from severe implementation difficulties associated with RAKE structures in the analog domain with extremely large number of fingers, or high sampling rates for digital-based receivers. On the other hand, non-coherent receivers, such as transmitted reference systems [5], are typically based on the maximization of the received signal energy for different time window candidates, and do not require a-priori estimation of the channel. In any case, coherent and non-coherent receivers face the same fundamental issue that the search space is infinite for analog architectures, which require accurate digitally controlled analog delay lines [6] and are difficult to implement,

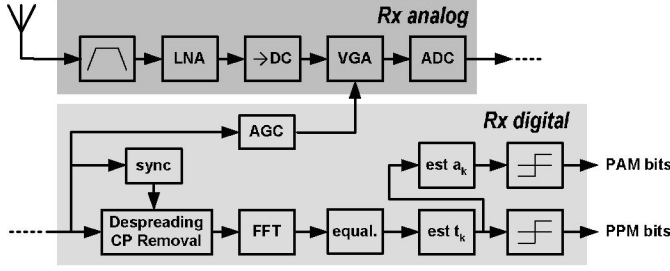


Fig. 1. Subsampling Receiver Architecture.

and remains prohibitively huge for digital architectures sampling at Nyquist rate. What is worse, synchronization at code level increases the search space size proportionally to the code length. This issue has motivated many search optimization techniques and 2-steps synchronization algorithms [7].

Subsampling receivers provide a suboptimal but fully digital and flexible alternative which allows several signal processing techniques to be applied. Combined channel estimation and synchronization methods were proposed in [8] and [9]. However, compared to the solutions investigated in this paper, they require higher ADC sampling rates, a-priori knowledge of the number of multipaths, longer preambles, and frequency domain processing. As further developed in Section IV, we rely instead on time domain algorithms which do not require any knowledge of the channel characteristics.

III. BASIC PRINCIPLES OF THE SUBSAMPLING RECEIVER

This section summarizes how the proposed subsampling receiver (fig. 1) demodulates the incoming signal, to ease the identification of synchronization algorithms which take benefit of its peculiarities.

A received pulsed UWB signal $r(t)$ can be modelled as the convolution between a stream of Diracs sent at frame rate $1/T_f$ and the compound channel $h_c(t) = \sum_{i=1}^{N_p} \alpha_i p_i(t - \tau_i)$, which includes the pulse distortion caused by the antennas and the dispersive behavior of the building materials in the propagation channel:

$$r(t) = h_c(t) \otimes \sum_{n=-\infty}^{+\infty} \sum_{k=1}^K a_{n,k} \delta(t - nT_f - t_{n,k}) + n(t) \quad (1)$$

where $a_{n,k} \in \{0, \pm A, \pm 3A, \dots\}$ and $t_{n,k} \in \{0, \Delta, 2\Delta, \dots\}$ are the data streams modulating K pulse amplitudes and positions per period T_f , respectively, and $n(t)$ is the received additive white gaussian noise (AWGN). For the sake of simplicity, we omit in (1) the TH or DS spreading code as the working principles of the subsampling receiver are independent of the chosen spreading technique.

The received signal $r(t)$ is filtered, subsampled according to [10], and despread appropriately. Following the work on sampling signals with finite rate of innovation [10], we suggested in [2] to apply a line spectrum PSD estimation method, such as ESPRIT, in the frequency domain to estimate the position of the Diracs, after compensation of the channel effects by a simplified MMSE equalizer. The application of

an equalizer is motivated by the substantial simplifications it introduces in the subsequent blocks: a line spectrum method of order K only is required, instead of KN_p .

A line spectrum estimation method such as ESPRIT requires expanding a data matrix using a Singular Value Decomposition (SVD), solving a least-squares system and computing the roots of a polynomial. In the particular case of $K = 1$ (i.e. a single pulse per period T_f), which requires the lowest sampling rate and will be assumed, the implementation of ESPRIT is reduced to the SVD, as the other operations become trivial, and can be efficiently realized with a systolic array [11][12].

IV. SYNCHRONIZATION OF THE SUBSAMPLING RECEIVER

A. Preamble Signal Model

Similarly to common existing wireless systems, we assume data-aided synchronization: the transmitter sends at the beginning of each packet a sequence of symbols known at the receiver, to aid synchronization acquisition and training of the equalizer. This preamble consists, for example, of a repetition of identical symbols terminated by a sign inverted symbol. This sign change allows the synchronization at preamble level by means of a running autocorrelation or a simple sign inversion detection, without having to perform any kind of demodulation to recognize the preamble pattern.

As motivated in section II, we assume the transmitter is spreading the symbols in the preamble using PN codes, although other types of code may be used. For the sake of simplicity, we assume that the code is used to modulate the amplitude of the pulses (DS-UWB), not their position. This decision simplifies the implementation since it allows for pipelined operation, as shown in the following sections, and eases the routing of the wires from the shift registers to the combinational logic computing the despread symbols.

As an example, the shortest sequence of symbols for a receiver relying on the autocorrelation of two consecutive symbols is $\{A, A, -A\}$. For a pulse repetition rate of 20 MHz, which prevents inter frame interference (IFI) with common UWB channel models [13], and a PN code of length $N_c = 31$, the preamble length for the synchronization is equal to $3 \times 31 \times 50\text{ns} = 4.65\mu\text{s}$. Adding few symbols for stabilization of the Automatic Gain Control (AGC) still keeps the preamble length shorter than current high data rate wireless systems such as IEEE 802.11a (16 μs).

The received preamble can therefore be expressed as

$$r_p(t) = h_c(t) \otimes \sum_{i=0}^{N_p-1} \sum_{j=0}^{N_c-1} a_i c_j \delta(t - iN_c T_f - jT_f) + n(t) \quad (2)$$

where N_p is the number of symbols in the preamble, $a_i \in \{-A, A\}$ the pulse amplitudes, and $c_j \in \{-1, 1\}$ the PN code. This signal is subsampled at rate $1/T_s$, and $N_f = T_f/T_s$ complex samples per pulse period are available.

B. Synchronization Algorithms

For a time window of $N_s = N_c N_f$ samples, the receiver can despread the received signal by N_s different permutations of

the PN code upsampled by a factor N_f . The synchronization algorithm must detect which permutation maximizes the SNR, i.e. the signal energy collected after despreading.

Given a sequence of samples $x[k] = r_p(kT_s + \epsilon)$, where $\epsilon < T_s$ is the initial clock offset between the transceivers and $0 \leq k \leq N_p N_s - 1$, different synchronization algorithms can be investigated.

1) *Maximum autocorrelation estimated with N_s samples:*

From a window of N_s consecutive samples of $x[k] = s[k] + n[k]$, where $s[k]$ and $n[k]$ are the signal and noise contribution at the output of the sampling device, the autocorrelation of despread samples for different positions of the pulse period boundaries and permutations of the PN code can be computed.

The despread symbol $y^{(i)}[k]$, $0 \leq k \leq N_f - 1$, corresponding to the code permutation $0 \leq i \leq N_s - 1$ is given by

$$y^{(i)}[k] = \frac{1}{N_c} \sum_{j=0}^{N_c-1} c[j] x[(k + jN_f + i) \pmod{N_s}] \quad (3)$$

$$= y_s^{(i)}[k] + y_n^{(i)}[k] \quad (4)$$

where $y_s^{(i)}[k]$ and $y_n^{(i)}[k]$ correspond to the despread signal and noise contribution for the i^{th} permutation. By approximating $n[k]$ with an AWGN random variable with variance σ_n^2 , we have $y^{(i)}[k] \sim N(y_s^{(i)}[k], \frac{\sigma_n^2}{N_c})$. The estimated power of the despread noiseless signal corresponding to the i^{th} code permutation is given by $P^{(i)} = \frac{1}{N_f} \sum_{k=0}^{N_f-1} |y^{(i)}[k]|^2$, and

$$P^{(i)} = \frac{1}{N_f} \sum_{k=0}^{N_f-1} |y_s^{(i)}[k]|^2 + \frac{1}{N_f} \sum_{k=0}^{N_f-1} |y_n^{(i)}[k]|^2 + \frac{1}{N_f} \Re \left\{ 2 \sum_{k=0}^{N_f-1} y_s^{(i)}[k] y_n^{*(i)}[k] \right\} \quad (5)$$

$$= P_s^{(i)} + P_n^{(i)} + P_{sn}^{(i)} \quad (6)$$

The synchronization point is given by $i_{sync} = \arg \max_{0 \leq i \leq N_s-1} P^{(i)}$ which, in absence of noise, corresponds to the synchronization point maximizing the signal power $i_{sync,opt} = \arg \max_{0 \leq i \leq N_s-1} P_s^{(i)}$.

The noise corrupts the deterministic part of $P^{(i)}$, $P_s^{(i)}$, by adding 2 random contributions:

- $P_n^{(i)} \sim \frac{\sigma_n^2}{N_c N_f} \chi_{2N_f}^2$, a term proportional to the noise power of the received signal
- $P_{sn}^{(i)}$, a signal-cross-noise term converging to zero as N_f increases, since $E[y_s^{(i)}[k]] = E[y_n^{(i)}[k]] = 0$.

As a results, $P^{(i)}$ follows a non-central chi-square distribution for large N_f .

Clearly, the drawback of this algorithm is the bias introduced in the estimated signal power caused by the noise. Another negative point is the correlation between the noise affecting the different $P^{(i)}$. It can indeed be shown that

$$E[(P^{(i)} - P_s^{(i)})(P^{(j)} - P_s^{(j)})] \neq 0 \quad \forall i, j \quad (7)$$

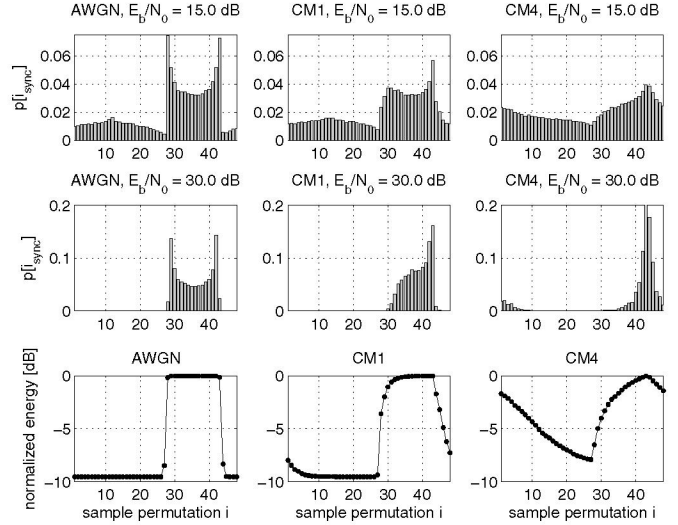


Fig. 2. Distribution of position of the synchronization point ($N_c = 3$, $N_f = 16$, $1/T_s = 320\text{MHz}$, $T_f = 50\text{ns}$) estimated from N_s samples, and corresponding collected energy.

If the noise affecting $P^{(i)}$ was not correlated, the cdf of $\max_{0 \leq i \leq N_s-1} P^{(i)}$ would be simply given by the product of the cdfs of $P^{(i)}$ and the pdf of i_{sync} analytically deduced. Simulations indicate that the noise correlation has as consequence that $E[i_{sync}] \neq i_{sync,opt}$ at low SNR (fig. 2). In case short of impulse response of the compound channel, N_f different alternatives give similar autocorrelation. The synchronization algorithm should therefore choose any of these with same probability, i.e. $P_s^{(i)} = P_s^{(j)} \Leftrightarrow p[i_{sync} = i] = p[i_{sync} = j]$. Instead, it favours permutations where the boundaries of the code are close to the location of the pulse, decreasing therefore the robustness of the system against clock offset and increasing the risk of choosing a sub-optimal permutation in presence of a channel with an impulse response much shorter than the pulse period. However, $\lim_{E_b/N_0 \rightarrow \infty} E[i_{sync}] = i_{sync,opt}$. Moreover, the SNR degradation remains negligible, as shown in Section V. Finally, the issue is quickly alleviated in presence of longer channel response.

The chief advantage of this algorithm is its moderate implementation cost and fast synchronization time. Indeed, the shortest preamble for synchronization is $\{A, -A\}$. Instead of working on a single symbol stored into memory, an unpractical and power hungry solution at the sampling rate of this system, incoming samples can be stored in a shift register and a running estimation of the despread symbol and autocorrelation can be performed (fig. 3). Under the assumption of an unrolled architecture required by the fast clock speed, and before fixed-point optimization, the algorithm needs

- in case the autocorrelation is computed every cycle from all available samples: $2N_s b$ flip-flops (FFs), $2(N_c - 1)$ adders for the despreading, and $4N_f$ multipliers of $(b + \lceil \log_2(N_c) \rceil)$ bits, where b is the ADC bit width. This approach makes the chip layout particularly difficult, requires a prohibitive number of multipliers, and will

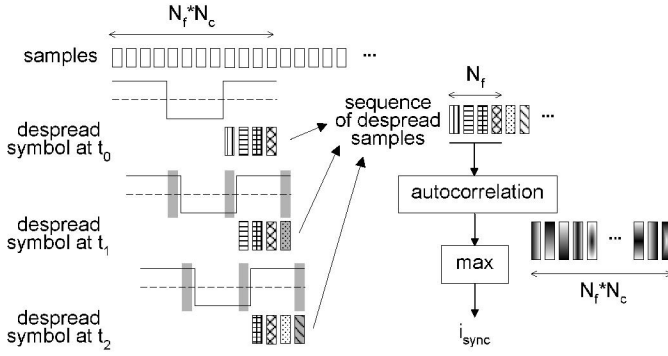


Fig. 3. Architecture for autocorrelation based algorithm. The despread symbol at t_{i+1} only differs from t_i by 1 sample, which can be computed with a tree of $2(N_c - 1)$ adders.

therefore not be further considered.

- in case the autocorrelation is computed at cycle i from the value at previous cycle $i-1$: $2N_s b + 2N_f(b + \lceil \log_2 N_c \rceil)$ FFs, $2(N_c - 1)$ adders for the despreading, and 8 multipliers of $(b + \lceil \log_2(N_c) \rceil)$ bits.

2) *Maximum cross-correlation estimated with $2N_s$ samples:*

Compared to previous solution, the correlation between two consecutive despread windows of N_s samples is computed instead of the autocorrelation of a single window. The obvious benefit is an increased robustness against noise as the squared noise term in eq. (5) is avoided.

In this case, $P^{(i)} = \frac{1}{N_f} \sum_{k=0}^{N_f-1} y_1^{(i)}[k] y_2^{*(i)}[k]$, where

$$\begin{aligned} y_1^{(i)}[k] &= \frac{1}{N_c} \sum_{j=0}^{N_c-1} c[j] x[(k + jN_f + i) \pmod{N_s}] \\ &= y_{s,1}^{(i)}[k] + y_{n,1}^{(i)}[k] \end{aligned} \quad (8)$$

$$\begin{aligned} y_2^{(i)}[k] &= \frac{1}{N_c} \sum_{j=0}^{N_c-1} c[j] x[(N_c N_f + k + jN_f + i) \pmod{N_s}] \\ &= y_{s,2}^{(i)}[k] + y_{n,2}^{(i)}[k] \end{aligned} \quad (9)$$

As $y_{s,1}^{(i)}[k] = y_{s,2}^{(i)}[k] \triangleq y_s^{(i)}[k]$, $y_1^{(i)}[k]$ and $y_2^{(i)}[k]$ both follow a $N\left(y_s^{(i)}[k], \frac{\sigma_n^2}{N_c}\right)$ distribution, where we approximate as before $n[k]$ by an AWGN with variance σ_n^2 . As a result,

$$\begin{aligned} P^{(i)} &= \frac{1}{N_f} \sum_{k=0}^{N_f-1} |y_s^{(i)}[k]|^2 + \frac{1}{N_f} \sum_{k=0}^{N_f-1} y_s^{(i)}[k] y_{n,2}^{*(i)}[k] + \\ &\quad \frac{1}{N_f} \sum_{k=0}^{N_f-1} y_s^{*(i)}[k] y_{n,1}^{(i)}[k] + \frac{1}{N_f} \sum_{k=0}^{N_f-1} y_{n,1}^{(i)}[k] y_{n,2}^{*(i)}[k] \\ &= P_s^{(i)} + P_{sn}^{(i)} + P_{ns}^{(i)} + P_n^{(i)} \end{aligned} \quad (10)$$

and the synchronization point is given by $i_{sync} = \arg \max_{0 \leq i \leq N_f-1} P^{(i)}$. The noise corrupts $P_s^{(i)}$, the deterministic part of $P^{(i)}$, by adding 3 random contributions:

- $P_n^{(i)}$, noise-cross-noise term: each product $y_{nn} \triangleq y_{n,1}^{(i)}[k] y_{n,2}^{*(i)}[k]$ has a probability density function given by

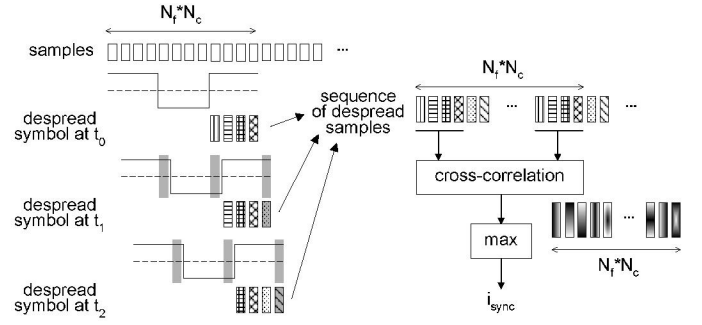


Fig. 4. Architecture for cross-correlation method, single despreading logic.

$p[y_{nn}] = \frac{1}{\pi \sigma_n^2} K_0 \left(\frac{|y_{nn}|}{\sigma_n^2} \right)$. By virtue of the central limit theorem, $P_n^{(i)}$ can be approximated by $N\left(0, \frac{\sigma_n^4}{N_c^2 N_f}\right)$.

- $P_{sn}^{(i)}$ and $P_{ns}^{(i)}$, which follow $N\left(0, \frac{\sigma_n^2}{N_c N_f}\right)$.

Compared to previous solution, the noise on $P^{(i)}$ has reduced variance but remains correlated, and the shortest preamble becomes now $\{A, A, -A\}$. From an implementation viewpoint, and assuming directly a pipelined architecture (i.e. the cross-correlation at cycle i is computed from the previous cycle $i-1$), it requires

- in case the despread samples are stored into a shift register (fig. 4): $2N_s b + 2N_s(b + \lceil \log_2 N_c \rceil)$ FFs, $2(N_c - 1)$ adders for the despreading, and 8 multipliers of $(b + \lceil \log_2(N_c) \rceil)$ bits.
- in case the logic to compute the despread symbols is duplicated: $4N_s b + 4N_f(b + \lceil \log_2 N_c \rceil)$ FFs, $4(N_c - 1)$ adders for the despreading, and 16 multipliers of $(b + \lceil \log_2(N_c) \rceil)$ bits. This solution trades logic against FFs.

3) *Maximum singular value of SVD from N_s samples:* The SVD present in the system for demodulation purposes can be also used to acquire synchronization and save significant chip area. Indeed, there are N_f ways of chopping up a window of N_s samples into N_c vectors of N_f consecutive samples $\mathbf{x}_j^{(i)} = [x[(i + jN_f) \pmod{N_s}], \dots, x[(N_f - 1) + i + jN_f) \pmod{N_s}]]$, $0 \leq i \leq N_f - 1$, $0 \leq j \leq N_c - 1$. These vectors are then stacked column-wise into a matrix of size $N_f \times N_c$. We obtain N_f different matrices $\mathbf{X}^{(i)} = [\mathbf{x}_0^{(i)} \dots \mathbf{x}_{N_c-1}^{(i)}] = \mathbf{S}^{(i)} + \mathbf{N}^{(i)}$, where $\mathbf{S}^{(i)}$ and $\mathbf{N}^{(i)}$ correspond to the signal and noise contributions in $\mathbf{X}^{(i)}$.

In absence of noise, the optimum synchronization point corresponds to the permutation $i_{sync,opt}$ which yields to $\mathbf{X}^{(i_{sync,opt})} = \mathbf{S}^{(i_{sync,opt})}$ with the greatest ratio $\sigma_{s,0}^{(i)}/\sigma_{s,1}^{(i)}$ between its first and second singular values. Indeed, in presence of IFI, only $\mathbf{S}^{(i_{sync,opt})}$ will have a rank equal to 1, and corresponds to the optimum alignment of the PN code. All other permutations give $\text{rank}(\mathbf{S}^{(i)}) \neq 1$, as the singular values $[\sigma_{s,1}^{(i)}, \dots, \sigma_{s,N_c-1}^{(i)}]$ are nonzero.

In presence of noise, the singular values of $\mathbf{S}^{(i)}$ will be corrupted by noise, and the SVD of $\mathbf{X}^{(i)} = \mathbf{U}^{(i)} \mathbf{\Sigma}^{(i)} \mathbf{V}^{(i)H}$ gives the singular values $\sigma_k^{(i)} = \sigma_{s,k}^{(i)} + \sigma_{n,k}^{(i)}$, $0 \leq k \leq N_c - 1$. The synchronization point corresponds to the permutation

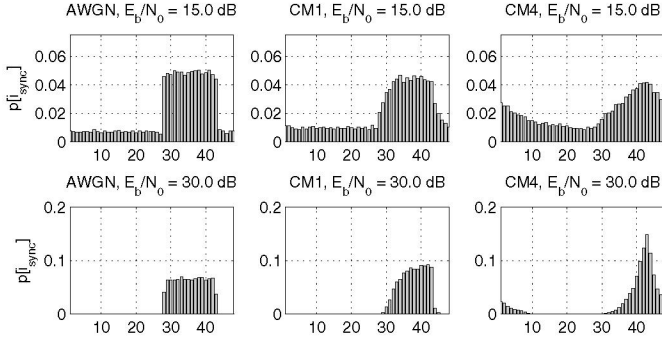


Fig. 5. Distribution of the position of the synchronization point ($N_c = 3$, $N_f = 16$, $1/T_s = 320\text{MHz}$, $T_f = 50\text{ns}$) estimated from N_s^2 samples.

$i_{sync} = \arg \max_{0 \leq i \leq N_c-1} \sigma_0^{(i)} / \sigma_1^{(i)}$. As the noise power increases, $\sigma_{n,k}^{(i)}$ will gradually become higher than $\sigma_{s,k}^{(i)}$, and $\sigma_{s,1}^{(i)}$ is the last singular value in $[\sigma_{s,1}^{(i)}, \dots, \sigma_{s,N_c-1}^{(i)}]$ to become negligible with respect to the noise, and motivates the decision to choose $\sigma_0^{(i)} / \sigma_1^{(i)}$ as a decision variable.

This solution has the advantage of providing directly the PN code in the first column of $\mathbf{V}^{(i_{sync})}$. However, the correlation of the noise affecting the decision variable $\sigma_0^{(i)} / \sigma_1^{(i)}$ can lead to suboptimal results, as for algorithms 1 and 2. From an implementation viewpoint, and assuming a systolic array based on unrolled CORDICs of $b_{cor} \geq b$ stages to achieve maximal speed, where b_{cor} is the CORDIC bit width, this algorithm requires $2N_s b$ FFs to store the samples, and N_s CORDICs requiring each $3b_{cor}^2$ FFs and $3b_{cor}$ adders of b_{cor} bits. However, this implementation cost is virtually zero if the same hardware is used for this SVD and the one in ESPRIT, which is possible by choosing carefully the code length and sampling rate of the receiver.

4) *Maximum autocorrelation/cross-correlation/singular value ratio estimated with N_s^2 samples:* If successive windows of N_s samples are taken from N_s^2 samples instead of N_s , and the i^{th} window is used to estimate the decision variable at index i , the noise affecting these variables becomes uncorrelated, and $p[i_{sync}]$ follows the energy profile (fig. 5). We will not consider further this solution as the preamble length and synchronization time become unacceptably long, whereas the performance gain remains negligible.

C. Signal Detection

Signal Detection can be carried out together with the synchronization procedure. A simple and classical solution is to compare the synchronization decision variable to a threshold which must be determined according to a given criterion. The threshold is static in our system as the signal power at the input of the synchronization block is fixed at a level specified by the AGC [14]. In this paper, we will use the Neyman-Pearson criterion [15], which maximizes the probability of detection P_D while minimizing the probability of false alarm P_{FA} . By representing the events of signal detection, presence and absence of a signal by D , S and $\bar{S}$ respectively, $P_D = P(D|S)$ and $P_{FA} = P(D|\bar{S})$.

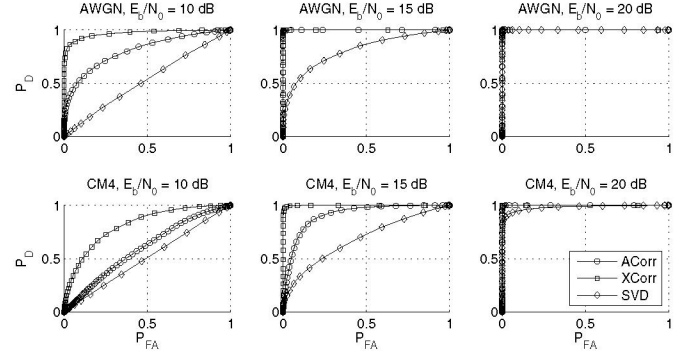


Fig. 6. P_D vs. P_{FA} in AWGN and multipath channel conditions, $N_c = 31$.

V. PERFORMANCE AND IMPLEMENTATION COST OF THE SYNCHRONIZATION ALGORITHMS

A. Synchronization Metric

In order to have a metric independent of the demodulation technique, we will rely on the SNR degradation at the chosen synchronization point instead of BER curves. Synchronization algorithms will be assessed in terms of *probability of synchronization* P_{sync} , i.e. the likelihood that the a valid synchronization point is chosen and a signal is correctly detected. The validity of the synchronization decision will be determined by comparing the SNR at the chosen synchronization point SNR_{sync} with respect to the SNR achieved at the optimal synchronization point, SNR_{opt} . Correct synchronization is declared if the SNR degradation $SNR_{opt} - SNR_{sync}$ caused by missynchronization is below a tolerable degradation Δ_{SNR} , i.e., $P_{sync} = P((SNR_{opt} - SNR_{sync} \leq \Delta_{SNR}) \cap D|S)$. In this paper, we will consider $\Delta_{SNR} = 1\text{dB}$.

B. Comparison of the Algorithms

Figure 6 illustrates few examples of receiver operating characteristics (ROCs) for the presented solutions, and shows the degradation caused by IFI as well as the superiority of the algorithm based on cross-correlation. A closer look at the SNR degradation for various code lengths (fig. 7) indicates that the performance of the SVD does not significantly improve with longer codes. Indeed, the SVD method does not need to compute the despread signal but the risk of wrongly estimating the PN code alignment does not decrease with longer codes. Instead, the despreading operation provides stronger robustness for the auto- and cross-correlation based methods as the code length increases, although the IFI ultimately sets a lower bound on the SNR degradation. A similar conclusion can be drawn for P_{sync} (fig. 8).

The different architectures proposed for the auto- and cross-correlation in section IV-B have been prototyped on Xilinx Virtex 4 LX200 FPGA with a clock rate slower by a factor 3 compared to the target ADC and ASIC clock speed of 320 MHz. Advanced tools have been used which allow direct implementation from Simulink towards FPGA [16]. This design flow can be extended towards ASIC development [17]. At the

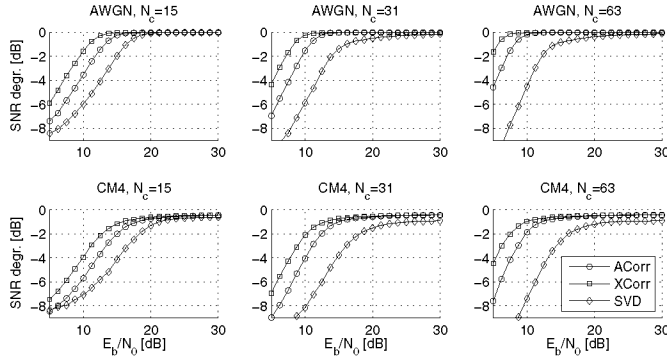


Fig. 7. SNR degradation at the estimated synchronization point in AWGN and multipath channel conditions, for various code lengths.

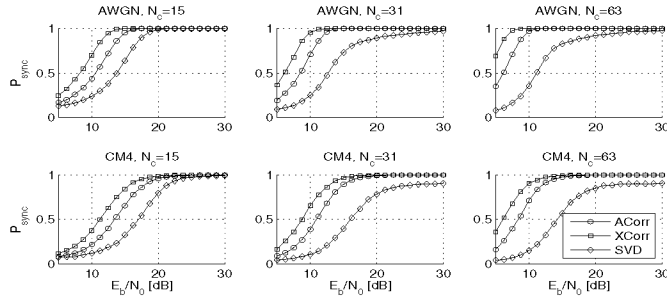


Fig. 8. P_{sync} in AWGN and multipath channel conditions, for various code lengths.

time of writing, the implementation of the SVD is on-going and we present here a pre-synthesis estimation.

The results for different bit widths and architectures are shown on figure 9 and validate the estimation carried out in section IV-B. Fixed-point optimization allows to further optimize the area. As expected from our earlier estimation, the algorithm based on cross-correlation should be implemented with duplicated despreading logic, as it allows saving significant FFs area at moderate extra cost of logic area. However, fixed-point optimization can reduce significantly the difference between both architectures. The autocorrelation method has obviously the lowest total implementation cost. The SVD cost is indicated for information, as this functionality comes actually with minor overhead in the complete system since it is already present inside ESPRIT and only requires extra logic to select the operating mode (synchronization or demodulation). Although the presented results correspond to an FPGA target, they provide an interesting estimation for an ASIC implementation [18].

VI. CONCLUSIONS

Different synchronization alternatives for a digital based subsampling receiver in the 3.1-10.6 GHz frequency band have been evaluated and prototyped on FPGA. The optimal choice of the algorithm depends on the specification of the system. While cross-correlation based synchronization provides the highest robustness among the proposed approaches, the SVD

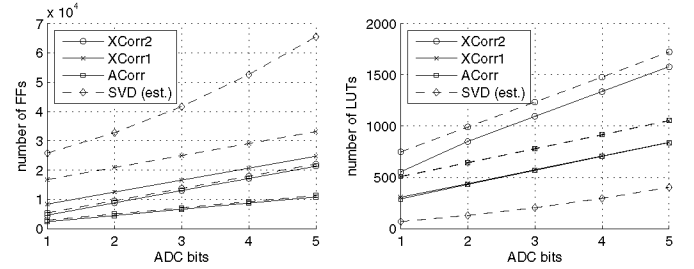


Fig. 9. FPGA area for the different algorithms before (dashed lines) and after (continuous lines) fixed-point optimization.

method yields to significant savings for an implementation of our subsampling receiver where area and power consumption are the dominant criterion. Future work will concentrate on the complete implementation of the system for a detailed evaluation of its complexity.

ACKNOWLEDGEMENT

This work has been developed in the context of the MEDEA+ UPPERMOST project. The authors gratefully thank Jeff Weintraub from Xilinx for prompt tool support.

REFERENCES

- [1] M. Chen and R. Brodersen, "A Subsampling UWB Impulse Radio Architecture Utilizing Analytic Signaling," *IEICE Trans. Electronics*, vol. E88-C, no. 6, pp. 1114–1121, 2005.
- [2] Y. Vanderperren, W. Dehaene, and G. Leus, "A Flexible Low Power Subsampling UWB Receiver Based on Line Spectrum Estimation Methods," in *Proc. IEEE Int. Conf. on Comm.*, 2006.
- [3] S. Aedudodla, S. Vijayakumaran, and T. Wong, "Timing Acquisition in Ultra-wideband Communication Systems," *IEEE Trans. Vehic. Tech.*, vol. 54, no. 5, pp. 1570–1583, 2005.
- [4] J. Chen and Z. Zhou, "Overview of Synchronization in DS-UWB," in *Proc. IEEE Symp. Comm. and Inf. Tech.*, 2005.
- [5] R. Djapic, G. Leus, A.-J. van der Veen, and A. Trindade, "Blind Synchronization in Asynchronous UWB Networks Based on the Transmit-Reference Scheme," *EURASIP J. Wireless Comm. Netw.*, 2006.
- [6] L. Yang and G. Giannakis, "Timing Ultra-Wideband Signals with Dirty Templates," *IEEE Trans. Comm.*, vol. 53, no. 11, pp. 1952 – 1963, 2005.
- [7] E. Homier, "Synchronization of Ultra-Wideband Signals in the Dense Multipath Channel," Ph.D. dissertation, Univ. Southern Calif., 2004.
- [8] I. Maravić and M. Vetterli, "Low-Complexity Subspace Methods for Channel Estimation and Synchronization in Ultra-Wideband Systems," in *Int. Workshop on UWB Systems*, 2003.
- [9] J. Zhang et al., "Principal Components Tracking Algorithms for Synchronization and Channel Identification in UWB Systems," in *IEEE 8th Int. Symp. Spread Spectrum Tech. and Appl.*, 2004, pp. 369–373.
- [10] M. Vetterli, P. Marziliano, and T. Blu, "Sampling Signals w. Finite Rate of Innovation," *IEEE Trans. Signal Proc.*, vol. 50, pp. 1417–1428, 2002.
- [11] R. P. Brent, F. T. Luk, and C. V. Loan, "Computation of the Singular Value Decomposition Using Mesh-Connected Processors," *J. VLSI Computer Systems*, vol. 1, no. 3, pp. 243–270, 1985.
- [12] N. D. Hemkumar, "A Systolic VLSI Architecture for Complex SVD," Master's thesis, Rice University, 1991.
- [13] J. Foerster, "Channel Modeling Sub-Committee Report Final (IEEE P802.15-02/490r1-SG3a)," 2003.
- [14] Y. Vanderperren, G. Leus, and W. Dehaene, "An Approach for Specifying the ADC and AGC Requirements for UWB Digital Receivers," in *IET Seminar on UWB Syst., Technologies and Applic.*, 2006.
- [15] V. Poor, *An Introd. to Signal Detection and Estimation*. Springer, 1994.
- [16] Xilinx, *System Generator for DSP v9.1 User Guide*, 2007.
- [17] B. Richards, C. Chang, and R. Brodersen, "DSP System Design using the BEE Hardware Emulation Environment," in *Proc. 37th Asilomar Conf. Sig., Syst. and Comp.*, 2003.
- [18] Xilinx App. Note 059, "Gate Count Capacity Metrics for FPGAs."