Performance Evaluation of Message-Passing-Based Algorithms for Fast Acquisition of Spreading Codes with Application to Satellite Positioning

Fabio Principe [(1)], Keith M. Chugg [(2)], and Marco Luise [(1)]

[(1)] *Dipartimento di Ingegneria dell'Informazione – University of Pisa*
Via G. Caruso 16, 56122 Pisa, Italy
Phone: +39 050 2217511
E-mail: {fabio.principe, marco.luise}@iet.unipi.it

[(2)] *Communication Sciences Institute – University of Southern California*
3740 McClintock Ave., Los Angeles, CA 90089-2565, USA
Phone: +1 213 740 7294
E-mail: chugg@usc.edu

ABSTRACT

Exploiting the experience of iterative decoding of modern codes (LDPC and turbo codes), recent studies have proposed new acquisition methods based on iterative message passing algorithms (iMPAs) to be run on loopy graphs to detect linear feedback shift register (LFSR) sequences with low-complexity at low signal-to-noise ratio (SNR). In this context, our contribution addresses the issue to the evaluation of the mean and the variance of the acquisition time of iMPA detectors, using Markov chain theory. This theoretical analysis leads to a comparison between the performance of iMPA detectors and fully parallel and serial search algorithms in terms of computational complexity and average acquisition times.

I. INTRODUCTION

DIRECT-SEQUENCE/SPREAD-SPECTRUM (DS/SS) communications are widely used in wireless military and civil communications as well as in satellite positioning systems, because they provide low probability of interception, strong anti-jam protection, and low co-channel interferences. All these proprieties are basically due to use of long high-rate binary pseudo-noise (PN) sequences that *spread* the spectrum bandwidth and make it difficult to be detected and corrupted by jammers. At the receiver side, to correctly demodulate SS signals, a *despreading* operation is accomplished by correlating the incoming signal with its PN sequence, generated locally. Because of this, a precise code timing synchronization is necessary. This result is generally obtained in two receiver stages ([1], [2], and [3]): the *acquisition* stage, that provides a preliminary coarse alignment between the received PN sequence and its local replica, and the *tracking* stage, in which a fine synchronization is realized and maintained by a delay locked-loop (DLL) unit, exploiting the previous rough alignment. Therefore, the PN acquisition is the critical point to have a rapid and correct synchronization between DS/SS transmitters and receivers.

The standard and well-known acquisition techniques used to detect such sequences are ([1], [4]): full parallel search, simple serial search and hybrid search. The common denominator of all these techniques is that the received and local SS sequences are correlated and then processed by a suitable detector/decision rule to decide whether the two codes are in synchronism. The first method implements a *maximum-likelihood* (ML) estimation algorithm and needs a fully parallel search. Hence, it provides fast detection at price of a high implementation complexity, especially in case of long SS sequences. The simple serial search has lower complexity, but its acquisition time is prohibitively long. The hybrid search is a trade-off between these two algorithms.

Nevertheless, recent studies ([5], [6], and [7]), conducted independently, have presented a new technique to acquire *linear feedback shift register* (LSFR) sequences, demonstrating the perfect equivalence of a decoding and detection problem. This method is based on the paradigm of *message passing* (MP) on graphical models, and more specifically, on *iterative message passing algorithms* (iMPAs) to be run on loopy graphs ([8], [9], [10], [11], and [12]). In other words, instead of correlating the incoming signal with a local PN replica, this algorithm uses all the soft information provided by the received signals, as messages to be run on a predetermined graph, modelled on the structures of the LSFR sequence to be acquired, thus approximating the ML method. This results a sub-optimal algorithm, that searches all code phases in parallel with a complexity lower than the fully parallel implementation, and an acquisition time that is shorter than that of the simple serial algorithm.

This work addresses the problem to the evaluation of the mean and the variance of the acquisition time of iMPA detectors, exploiting Markov chain theory [13]. Starting from the Holmes's seminal work, presented in [14], to compute the *moment generating function* (MGF) of the acquisition time in case of a *non-coherent* serial detector, we reuse this method for our goal. In particular, after the description of an iMPA detector architecture, we present all the standard procedures and stages that an acquisition unit should follow to correctly acquire an incoming SS signal. Then, we build the Markov chain flow graph of the detector stages, that is used to evaluate the MGF of its acquisition time. From the MGF, we can easily compute the mean and the variance of the acquisition time ([2], [13], and [14]), that are compared to the corresponding parameters of the standard detection techniques (fully parallel and serial search). Furthermore, to complete this analysis, all these algorithms are also compared in terms of computational complexity and (missed/wrong) acquisition probability derived by simulations.

The reminder of this paper is structured as follows: section II introduces the DS/SS signal model used to measure detector performance, and gives an overview on iMPAs. A detailed description on the iMPA detector, its main stages, and the theoretical analysis to build the Markov chain and evaluate the main moments of the acquisition time are presented in section III. Section IV contains simulation results and comparisons between the new algorithm and the standard detection algorithms (fully parallel and simple serial search). Finally, the conclusions and suggestions for future works are reported in section V.

II. SIGNAL MODEL AND DETECTION ALGORITHM

A basic and standard base-band (BB) representation of a DS/SS communication system during the acquisition stage is reported in Fig. 1. It is basically made up of: a *base-band transmitter*, that produces a predetermined spreading sequence (LFSR sequences are considered), a *channel* that introduces a propagation delay ($\Delta \geq 0$) and an additive white gaussian noise (AWGN), and finally a *detection unit* that acquires received signals, estimating their code delay. The following subsections give more details for each stage of this communication system model.

A. Base-Band Transmitter

The BB transmitter is basically made up of a PN sequence generator that produces random binary sequences, **c** (each element is $c_k \in \{0,1\}$) and a *BPSK mapper* that outputs the correspondent antipodal sequences **y**, where each component is $y_k = (-1)^{c_k}$. Only *linear feedback shift register* (LFSR) generators are taken into account in this paper.

LFSR sequences are implemented by a r-stage shift register (SR) with a linear combination of its elements in a feedback path [1]. A general LFSR generator is shown in Fig. 2. As shown in the picture, at the generic time k, assuming that c_k is the SR output and c_{k+i} (with $0 \leq i \leq r$) is the content of the i^{th} register, the following parity equation is verified

$$g_r \cdot c_k \oplus g_{r-1} \cdot c_{k+1} \oplus \ldots \oplus g_1 \cdot c_{k+r-1} \oplus g_0 \cdot c_{k+r} = \bigoplus_{i=0}^{r} g_{r-i} \cdot c_{k+i} = 0$$

where $\oplus$ is modulo-2 addition and $g_i \in \{0,1\}$, $0 \leq i \leq r$, are the feedback coefficients (also referred to as *taps*). The most common way to represent an r-stage LFSR is providing its generating polynomial (that also gives the tap configuration of the code) as

$$P(D) = g_0 + g_1 \cdot D + \ldots + g_{r-1} \cdot D^{r-1} + g_r \cdot D^r = \sum_{i=0}^{r} g_i \cdot D^i$$

where D is the unit delay operator, and r is the polynomial degree. For a given degree r, g_0 and g_r are always 1.

M-Sequences are a subset of LFSR sequences, because their generating polynomials are *primitive* (see [1]). This characteristic implies that their period, $N = 2^r - 1$, is the maximum achievable with an r-stage LFSR generator (so they are also referred to as *maximal* LFSR sequences). Another consequence of this property is that these sequences are univocally identified by their generating polynomials, because the initial word of their SRs (except for the "forbidden" zero word) only produce a cyclical shift of the code.

We remark that in this work only m-sequences are considered to compare the iMPA detector performance to all standard acquisition algorithms, but, from the conclusions reported in [15], it is also possible to extend this analysis to Gold code detection.

B. Communication Channel

The incoming BB spreading signal at the detection unit is found to be

$$z_k = \sqrt{E_c} \cdot y_k + n_k = \sqrt{E_c} \cdot (-1)^{c_k} + n_k \tag{1}$$

where z_k is a noisy sample received by detection unit, y_k is the antipodal modulation of the spreading sequence chip c_k (N is the sequence period), and n_k is an additive white gaussian noise (AWGN) with mean value 0 and variance $N_0 / 2$. No data modulation is shown, since we are assuming to acquire a *pilot* signal with *coherent* detection. This is admittedly a simplified representation, that we use here to "isolate" the issue we are concerned with as is customary done in the spread-spectrum literature (as also reported in [1], [5], [6], [7], and [15]).

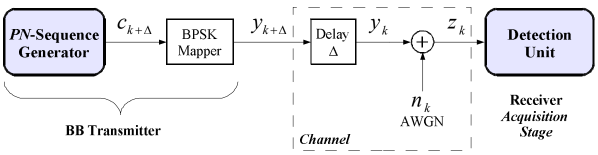

Fig. 1. Model of a DS/SS communication system.

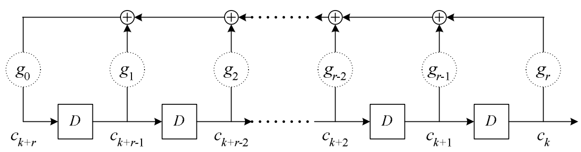

Fig. 2. Scheme of a r-stage LFSR generator.

C. Detection Unit

All standard detection algorithms operate by correlating the received signal with a shifted local replica of the code, until the right alignment is obtained ([1], [3], and [4]).

A new approach is proposed in [5], [6], [7], [15], and [16]. It is basically a generalization of the standard sequence decoding problem. Consider the M-dimensional received vector, $\mathbf{z} = [z_0, z_1, \ldots, z_{M-1}]$, the ML detection algorithm can be formulated as

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y}_i}\left[p\left(\mathbf{z}|\mathbf{y}_i\right)\right], \text{ with } i = 0,1,\ldots,N-1$$

where $\mathbf{y}_i$ is a vector that contains M shifted chips of the transmitted PN sequence, and $p(\mathbf{z}|\mathbf{y}_i)$ is the *likelihood function* of $\mathbf{y}_i$. Instead of casting our problem into one of delay estimation, we stick to a detection approach. We have to search into a set of N different sequences corresponding to all possible shifts of the spreading code. This problem is definitely similar to performing decoding of a block code, in which the ML decoder selects the codeword $\mathbf{y}_i$ (among a set of N different codewords that could have been transmitted) that maximizes $p(\mathbf{z}|\mathbf{y}_i)$, [2]. So, the equivalence between acquisition and decoding is clearly proved.

Exploiting the experience of iterative decoding of modern codes ([17], [18]), an ML algorithm can be implemented by a MPA run on graphical model without cycles (a *tree graph*). Unluckily, these optimal algorithms are often too complex to implement, so graphical models with cycles (e.g. *Tanner graphs*, TGs), [8] and [9], are commonly used. Indeed, these cyclic models yield sub-optimal solutions with lower complexity and it has been experimentally observed that, with a proper model design, the performance can be close to that of the MLA. These graphical models are, basically, made up of sets of *variable nodes*, directly associated to incoming soft information, and *check nodes*, that identify the parity equations (local constrains) verified by the transmitted code. Nevertheless, a systematic method for designing the best graphical model for a given specified code is not known.

Complete treatments on standard MPAs are reported in [8], [10], [11], and [12]. Roughly speaking, an iMPA, passes soft information between nodes in its graph, and each iteration ends when all nodes are activated. Hence, in order to correctly implement an iMPA, one must define its *activation schedule*, which is the order in which all variable and check nodes are activated, including when the algorithm is terminated. Typically, these algorithms end either when their estimated vectors verify all parity checks or when the max number of iteration, I_{MAX}, is obtained.

The last step is to define the processing used to perform the message updating. As reported in [8], [10], and [12], there are two main algorithms: *Sum-Product*, and *Min-Sum* algorithm. We only consider the Min-Sum algorithm (MSA) version because it is simpler and does not require an estimate of the operating signal-to-noise ratio (SNR).

The next section initially gives a simplified design of the iMPA detector and describes its main phases, then continues with a theoretical analysis to evaluate the mean and the variance of its acquisition time, exploiting Markov chain theory.

III. ACQUISITION TIME ANALYSIS

A basic architectural design of a coherent iMPA detector (also referred to as *iterative detection unit*, iDU) is shown in Fig. 3. As the picture shows, it is made up of an *input buffer* (IB), to store the received vector $\mathbf{z}$, an *iterative processing unit* (iPU), that runs the iMPA, which is the core of the iDU, and a *parity control unit* (PCU), that stops the acquisition procedure when the estimated binary vector verifies all the parity equations, otherwise a new iteration is carried out until the maximum number of iterations, I_{MAX}, is achieved.

More in detail, considering the communication system shown in Fig. 1, a predetermined LFSR sequence is transmitted through an AWGN channel. At the receiver side, an incoming vector of M observations, $\mathbf{z}$, (each element, z_k, is characterized in (1)) is stored in the IB. So, let T_c be the chip time of a generic z_k element (the chip rate is $R_c = 1 / T_c$), the required time to fill the buffer is $M\ T_c$. The iPU runs an iMPA on a predetermined loopy graph (see [8], [10], [11], and [12]). Typically, at the end of each iteration, the iMPA provides a *soft-output information* vector that is used to estimate the transmitted vector by hard decision, and, soon after, a parity control is executed on it ([15], and [16]). Because of this, the acquisition algorithm can end earlier than I_{MAX} iterations, as soon as the check is positive. Nevertheless, to simplify our analysis, we will over-bound the acquisition time, assuming that the iMPA always ends when all I_{MAX} iterations are run. Therefore, the time required by iPU, τ_{iPU}, to output a soft-output information vector (produced by the iMPA) and perform a hard-decision on it, is

$$\tau_{iPU} \triangleq \frac{I_{MAX}}{\rho} \cdot T_c$$

where I_{MAX} is the maximum number of iterations of the iMPA, T_c is the chip time, T_{it} is the time per iteration ($R_{it} = 1 / T_{it}$ is the iteration rate), and $\rho = T_c / T_{it} = R_{it} / R_c$ is the iMPA *time-factor*. Of course, the ρ–factor depends on the implementation technology of the receiver.

When all I_{MAX} iterations are run, the estimate vector, $\hat{\mathbf{c}}$, is handed over to the PCU that checks the parity. If the parity control fails, a *missed detection* is fed back and a new received vector is processed by the iDU. In this case, the missed detection time, τ_{MD}, is

$$\tau_{MD} \triangleq m \cdot T_c, \text{ with } m = \max\left(M, \frac{I_{MAX}}{\rho} \right). \qquad (2)$$

In other words, τ_{MD} is the longest time between $\tau_{RB} \triangleq M \cdot T_c$ (the time to refill the IB) and τ_{iPU}. If the check is passed, we may have either a *correct* or a *wrong acquisition*. Therefore, the receiver goes into a verification mode, which may include a long correlation test. At the end of the verification, it is reasonable to assume that the probability of wrong decision is close to 0. Hence, the case of wrong detection is characterized by a *penalty-time*, $\tau_{pt} \triangleq k \cdot M \cdot T_c$ (see also [14]). In the case of correct acquisition (often referred to as a *hit*) the signal is detected and tracking is started. Fig. 4 summarizes all the main stages of this detection strategy.

As illustrated, one acquisition attempt can end with just one of three mutually exclusive possible outcomes: missed detection (MD), or wrong detection (WD), or correct detection (CD). So, referring to the CD probability as P_{CD}, to the WD probability as P_{WD}, and to the MD probability as P_{MD}, the following equality is verified

$$P_{CD} + P_{MD} + P_{WD} = 1 \;\Rightarrow\; \frac{P_{CD}}{1 - P_{MD}} + \frac{P_{WD}}{1 - P_{MD}} = 1. \qquad (3)$$

We can now associate a Markov chain to the iDU, and use it to compute the *moment generating function* (MGF) that is necessary to evaluate the mean and the variance of the acquisition time ([2], [13], and [14]).

Assuming T_c as our *time unit*, the ad-hoc Markov chain in the *z*-transform domain is depicted in Fig. 5, where each node represents one of the stages of the iDU and each edge is labelled by a transition probability multiplied by its time delay. More specifically, starting from the *start* node, the *A-B* edge is the IB filling stage, labelled z^M because its time interval is $M\, T_c$. After that, the iPU provides an estimate vector on which parity checks are carried out by the PCU, so the missed detection is represented by the *B-B* edge, labelled $P_{MD}\, z^m$ (where *m* is (2)), while the right parity is the *B-C* edge, labelled $(1 - P_{MD})\, z^{I_{MAX}/\rho}$. Following on that line, the *C* node represents a verification mode that can confirm the CD with the *C-end* edge, labelled $P_{CD} / (1 - P_{MD})\, z^{k\,M}$, because the probability is $P_{CD} / (1 - P_{MD})$ and the verification stage time is $\tau_{pt} = k\, M\, T_c$, or a WD can happen with $P_{WD} / (1 - P_{MD})\, z^{k\,M}$, on the *C-D* edge. In this last case, a new detection try is run: the *D-D* edge represents the parity failure (MD), while *D-C* is the case of right parity. We remark that the D stage corresponds to the B one, but is split to simplify the next calculations.

Computing the MGF, we simplify the flow graph of Fig. 5 as shown in Fig. 6, where

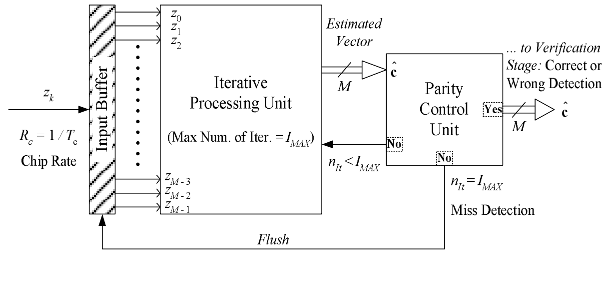

Fig. 3. Iterative detection unit (iDU) with iMPA.

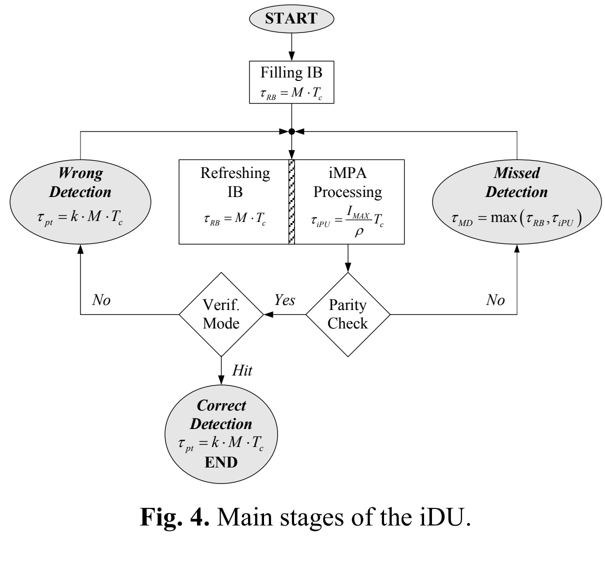

Fig. 4. Main stages of the iDU.

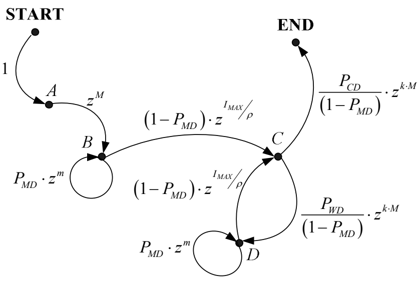

Fig. 5. Markov chain of the iDU.

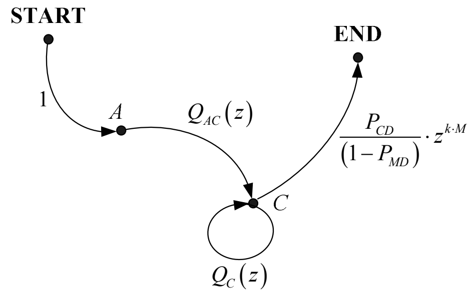

Fig. 6. Simplified flow graph of the iDU.

$$Q_{AC}(z)=\frac{(1-P_{MD})\cdot z^{\frac{I_{MAX}}{\rho}+M}}{1-P_{MD}\cdot z^m} \quad \text{and} \quad Q_C(z)=\frac{P_{WD}\cdot z^{k\cdot M+\frac{I_{MAX}}{\rho}}}{1-P_{MD}\cdot z^m} \tag{4}$$

so the MGF is (using (4) equations)

$$U(z)=\frac{P_{CD}}{(1-P_{MD})}\cdot Q_{AC}(z)\cdot\frac{1}{1-Q_C(z)}=\frac{P_{CD}\cdot z^{\frac{I_{MAX}}{\rho}+M\cdot(k+1)}}{(1-P_{MD}\cdot z^m)-P_{WD}\cdot z^{\frac{I_{MAX}}{\rho}+k\cdot M}} \tag{5}$$

we can check that (using (3)) U(1) = P_{CD} / (1 – P_{MD} – P_{WD}) = P_{CD} / P_{CD} =1, as it should be.

The mean of the acquisition time is derived from (5) ([2], [13], and [14])

$$\frac{\mu_{iDU}}{T_c}=\left.\frac{\partial U(z)}{\partial z}\right|_{z=1}\overset{U(1)=1}{\equiv}\left.\frac{\partial\ln[U(z)]}{\partial z}\right|_{z=1}=\left[\frac{I_{MAX}}{\rho}+M\cdot(k+1)\right]+\frac{P_{MD}\cdot m+P_{WD}\cdot\left(\frac{I_{MAX}}{\rho}+k\cdot M\right)}{P_{CD}} \tag{6}$$

and the variance is ([2], [13], and [14])

$$\begin{aligned}\frac{\sigma_{iDU}^2}{T_c^2}&=\left[\frac{\partial^2U(z)}{\partial z^2}+\frac{\partial U(z)}{\partial z}-\left(\frac{\partial U(z)}{\partial z}\right)^2\right]_{z=1}\overset{U(1)=1}{\equiv}\left[\frac{\partial^2\ln[U(z)]}{\partial z^2}+\frac{\partial\ln[U(z)]}{\partial z}\right]_{z=1}=\\&=\left(\frac{P_{MD}\cdot m+P_{WD}\cdot\left(\frac{I_{MAX}}{\rho}+k\cdot M\right)}{P_{CD}}\right)^2+\frac{P_{MD}\cdot m^2+P_{WD}\cdot\left(\frac{I_{MAX}}{\rho}+k\cdot M\right)^2}{P_{CD}}\;.\end{aligned} \tag{7}$$

Both these results ((6) and (7)) allow to evaluate the time performance of a coherent iDU with respect to the full parallel and serial search ones, as performed in the next section.

IV. COMPARISON WITH PARALLEL/SERIAL SEARCH AND SIMULATION RESULTS

This section compares the iDU to the full parallel and the serial search algorithms (respectively referred to as FPA and SSA). This comparison is performed in terms of detection performance, acquisition time, and implementation complexity. More specifically, the detection performance of each method is measured in terms of P_{WD}, P_{MD}, and P_{CD} as a function of the *signal-to-noise ratio* (SNR ≡ E_c / N_0). These curves are obtained by simulating the DS/SS communication system described in the section II. The time performance of each algorithm is characterized by the mean and the variance of its acquisition time. In particular, for the iDU, these parameters can be evaluated by (6) and (7). In case of a coherent SSA, considering that the dwell time is $\tau_D = M\,T_c$, we have (see [14])

$$\begin{aligned}&\frac{\mu_{SSA}}{T_c}\simeq\frac{2\cdot(k\cdot P_{CD}+1)+q\cdot(2-P_{CD})\cdot(1+k\cdot P_{FA})}{2\cdot P_{CD}}\cdot M,\ \text{when}\ q\gg1\\&\frac{\sigma_{SSA}^2}{T_c^2}\approx M^2\cdot q^2\cdot(1+k\cdot P_{FA})^2\cdot\left[\frac{1}{12}-\frac{1}{P_{CD}}+\frac{1}{P_{CD}^2}\right],\ \text{when}\ \mathrm{q}\gg k\cdot(1+k\cdot P_{FA})\ \text{and}\ q\gg1\end{aligned} \tag{8}$$

where q is the number of cells to be searched that depends on the SSA update size (ie. if the update size is one-half chip $q = 2\ N$, where N is the sequence period, [14]), and P_{FA} is the *false alarm probability*. The mean and variance of the FPA acquisition time can be computed by the following equations (assuming $\tau_D = M\ T_c$),

$$\frac{\mu_{FPA}}{T_c} = \frac{1+k}{P_{CD}} \cdot M \quad \text{and} \quad \frac{\sigma_{FPA}^2}{T_c^2} = \frac{(1+k)^2}{P_{CD}^2} \cdot (1-P_{CD}) \cdot M^2 . \tag{9}$$

To complete the comparison of these algorithms, an analysis in terms of implementation complexity is necessary. This can be done by counting the number of *sum*-operators of each method per detection try and comparing these figures. In the case of the Min-Sum algorithm, we assume that a *min*-operator is approximately equivalent to a *sum*-operator as outlined in [19].

We will perform this comparison on the specific example of the *m*-sequence $g_{18} = [1000201]_8$, with degree $r = 18$ and period $N = 262'143$. For the iMPA, at the receiver side (see Fig. 1), 1'024 observations (M) are collected. Furthermore, the SSA threshold is $\lambda = 0.85$. About the graphical model used by the iDU, we refer to [15] and [16]. Our Tanner graph (TG) is built grouping a set of sub-TGs constructed using equivalent sparse polynomials (with only 3 coefficients) of high order (more details are given in [15] and [16]). In this particular example a $YRGM_5$ (stands for *Yeung Redundant Graphical Model* of order 5, [15] and [16]) is implemented and the number of iteration is 30.

Some simulation results are shown in Fig. 7. In particular, the P_{FA} of the SSA is lower than the P_{WD} and P_{MD} of the iDU. Furthermore, it can be clearly neglected for SNR > –16 dB ($P_{FA} < 10^{-9}$). The detection probabilities of each algorithm (SSA, FPA, and iMPA) are given in Fig. 8. Of course, the best performance is provided by the MLA. The cross-over value SNR ≅ –13.8 dB splits the chart in two regions in which the iDU outperforms the SSA (SNR > –13.8 dB) and vice versa (SNR < –13.8 dB).

A comparison between the acquisition times of the SSA and the iMPA detector is contained in Tab. 1. To provide a realistic scenario, we can suppose a chip time $T_c \approx 0.1 \div 1$ μsec, that is typical of satellite positioning systems (as GPS, [20], and Galileo System, [21]), and an iteration time $T_{it} \approx 1$ μsec that can be considered a reasonable figure for the state-of-the-art of LDPC decoders. Therefore, $\rho = T_c / T_{it} \approx 0.1 \div 1$, so we can consider as the worst case: $\rho = 0.1$. Furthermore, we assume that the penalty time is the same for all the detectors and its value is proportional to 10 times the number of observations M, so $k = 10$. Finally, for the SSA two typical values of q are considered: N and $2\ N$ (it respectively means a search step of one chip or half a chip). All results contained in Tab. 1 are obtained considering the curves in Fig. 7 and Fig. 8 and the equations (6), (7), and (8). This table shows the huge gap in terms of the acquisition time between the iDU and the SSA. In particular, the iMPA detector has a mean time and a standard deviation about 10^5 times smaller than the SSA ones. This result is basically due to the q-factor that depends on the selected search step and on the sequence length.

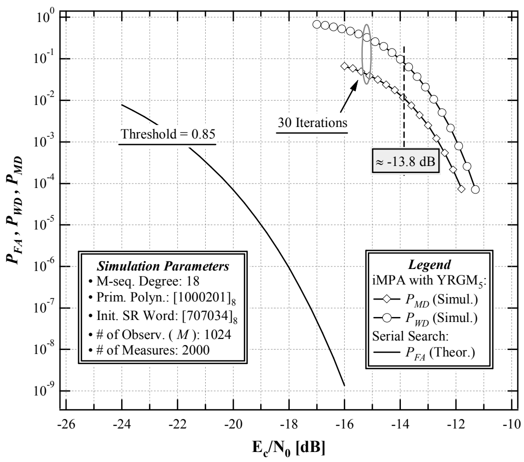

Fig. 7. The SSA P_{FA} vs the iMPA P_{WD} and P_{MD}.

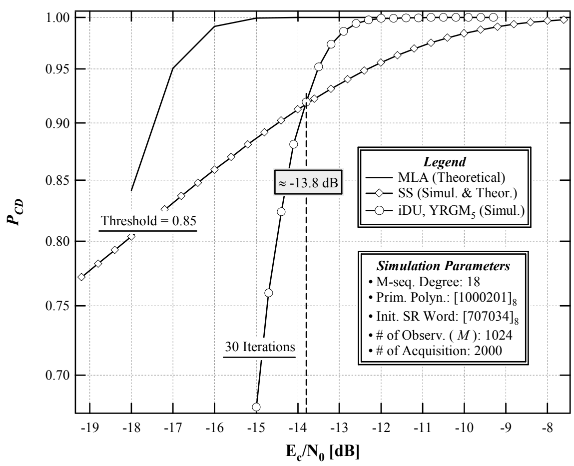

Fig. 8. P_{CD} of the iDU, the SSA, and the FPA.

Tab. 1. Comparison of the acquisition time between the SSA and the iDU.

SNR (dB)	SSA		iDU (iMPA Detector)			$q = N$		$q = 2\ N$	
	P_{FA}	P_{CD}	P_{WD}	P_{MD}	P_{CD}	μ_{iDU}/μ_{SSA}	$\sigma_{iDU}/\sigma_{SSA}$	μ_{iDU}/μ_{SSA}	$\sigma_{iDU}/\sigma_{SSA}$
–12	$< 10^{-9}$	≈ 0.956	≈ 0.0008	≈ 0.0002	≈ 0.999	$7.89\ 10^{-5}$	$3.07\ 10^{-6}$	$3.95\ 10^{-5}$	$1.53\ 10^{-6}$
–13.8	$< 10^{-9}$	≈ 0.92	≈ 0.07	≈ 0.01	≈ 0.92	$7.86\ 10^{-5}$	$2.67\ 10^{-5}$	$3.93\ 10^{-5}$	$1.33\ 10^{-5}$
–14.4	$< 10^{-9}$	≈ 0.902	≈ 0.155	≈ 0.021	≈ 0.824	$8.31\ 10^{-5}$	$4.12\ 10^{-5}$	$4.15\ 10^{-5}$	$2.06\ 10^{-5}$

A similar comparison between the iMPA and the FPA is performed in Tab. 2 (considering Fig. 8 and (6), (7), and (9)). In this case, the FPA implementation is optimal in terms of the acquisition time performance, but the iDU tends to its performance. More specifically, it is evident the difference between the iDU standard deviation and that of FPA, due to the highest P_{WD} and P_{MD} of the iMPA detector. Nevertheless, increasing the SNR, both the iDU acquisition time parameters (the mean and the standard deviation) tend to reduce the gap with respect to FPA. This result still demonstrates that iMPA is a sub-optimal solution of the MLA in terms of time performance.

Coming to the issue of complexity, let us call C_{Alg} the complexity of one detection algorithm (*Alg* is FPA, SSA, or iMPA). It is easy to prove that $C_{FPA} = M\,N = M\,(2^r - 1) = 268'434'432$ and $C_{SSA} \approx M = 1'024$. In case of the iMPA detector, the complexity strictly depends on the particular graphical model that has been built, and it is measured in agreement with the following equations

$$C_{iMPA} = T_{\Sigma}^{vn} + T_{\Sigma}^{cn}, \text{ where } \begin{cases} T_{\Sigma}^{vn} \le 2 \cdot \overline{N_{edg}^{vn}} \cdot N_{vn} \cdot I_{MAX} \\ T_{\min}^{cn} \approx T_{\Sigma}^{cn} \le \left(\overline{N_{edg}^{cn}} - 2\right) \cdot \overline{N_{edg}^{cn}} \cdot N_{cn} \cdot I_{MAX} \end{cases} \tag{10}$$

where T_{Σ}^{vn} and T_{Σ}^{cn} are the numbers of *sum*-operators of all variable/check nodes (a *min* is equivalent to a *sum*), N_{vn} and N_{cn} are the numbers of variable/check nodes of the TG, $\overline{N_{edg}^{vn}}$ and $\overline{N_{edg}^{cn}}$ are the mean numbers of edges per variable/check node. Using (10), C_{iMPA} can be computed and reported in Tab. 3 in comparison with the other algorithms. Our analysis shows that the iDU complexity is considerably smaller than that of the MLA implementation, and, of course, considerably higher than that of the SSA. Therefore the final conclusion is that the iMPA detector is a good trade-off between the FPA and SSA, because it allows to have a rapid detection (that tends to the FPA acquisition time) and good performance, in terms of correct detection probability at low SNR, with a complexity lower than a full parallel implementation.

V. CONCLUSIONS AND FUTURE WORKS

In this work, a novel detection technique that exploits iMPAs to perform spreading code acquisition is analyzed. It basically uses the channel soft information as messages to be exchanged within a graphical model with cycles to estimate the transmitted LFSR sequence, and so evaluating its code delay. In particular the graphical model can be implemented manipulating the generating polynomial structure of the sequence, as shown in [15] and [16].

The main feature that makes this algorithm very attractive is fast acquisition of long spread spectrum sequences. More specifically, the standard algorithms are not adequate to acquire these codes, because the full parallel implementation presents a rapid detection at a price of a high complexity, and a simple serial search results a low complexity algorithm but has a prohibitively long acquisition time. In this context, the iMPA detector is a good trade-off between these two techniques, because its correct detection probability is equivalent that of the SSA, but its performance in terms of acquisition time tends to that of a full parallel implementation obtained with a much lower complexity.

These conclusions, together with those in [5], [15], and [16], demonstrate that the iMPAs can yield low-complexity, unaided fast acquisition for long LFSR sequences, approximating the MLA implementation. Nevertheless, many topics still remain to be investigated, such as the acquisition with joint rough estimation of carrier phase and frequency, the search of other graphical models that could introduce more benefits, and more detailed considerations on the hardware implementation.

Tab. 2. Comparison of the acquisition time between the FPA and the iDU.

SNR (dB)	**FPA**	**iDU (iMPA Detector)**			μ_{iDU}/μ_{FPA}	σ_{iDU}	σ_{FPA}	$\sigma_{iDU}/\sigma_{FPA}$
	P_{CD}	P_{WD}	P_{MD}	P_{CD}				
–12	1	≈ 0.0008	≈ 0.0002	≈ 0.999	≈ 1.027	298.742	0	–
–13.8	≈ 1	≈ 0.07	≈ 0.01	≈ 0.92	≈ 1.099	3'020.783	≈ 0	–
–14.4	≈ 0.9995	≈ 0.155	≈ 0.021	≈ 0.824	≈ 1.204	4'995.884	≈ 251.997	19.825

Tab. 3. Comparison of the implementation complexity between of the iDU, the FPA, and the SSA.

YRGM$_5$ $I_{MAX} = 30$	$N_{vn} \equiv M$	N_{cn}	$\overline{N_{edg}^{vn}}$	$\overline{N_{edg}^{cn}}$	C_{iMPA}	C_{iMPA}/C_{SSA}	C_{iMPA}/C_{FPA}
	1'024	5'010	14.68	3	1'352'839.2	1'321.132	$1/198.42$

REFERENCES

[1] M. K. Simon, J. K. Omura, R. A. Scholtz, and B. K. Levitt, *Spread Spectrum Communications Handbook*. McGraw-Hill TELECOM, 2002

[2] J. G. Proakis, *Digital Communications*, 4th ed., S. W. Director, Ed. McGraw-Hill, 2001.

[3] R. L. Pickholtz, D. L. Schilling, and L. B. Milstein, "Theory of spread spectrum communications – A tutorial," *IEEE Transactions on Communications*, vol. COM-30, no. 5, pp. 855-884, May 1982.

[4] A. Polydoros, and C. L. Weber, "A unified approach to serial search spread-spectrum code acquisition – Part I and II," *IEEE Transactions on Communications*, vol. COM-32, no.5, pp. 542-560, May 1984.

[5] K. M. Chugg, and M. Zhu, "A new approach to rapid PN code acquisition using iterative message passing techniques," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 5, pp. 884-897, May 2005.

[6] B. Vigoda, J. Dauwels, M. Frey, N. Gershenfeld, T. Koch, H. A. Loeliger, and P. Merkli, "Synchronization of pseudo-random signals by forward-only message passing with applications to electronic circuits," *IEEE Transactions on Information Theory*, vol. 52, no. 8, pp. 3843-3852, August 2006.

[7] L. Yang, and L. Hanzo, "Iterative soft sequential estimation aided differential acquisition of *m*-sequences," in *Proc. Vehicular Technology Conference, 2003. VTC 2003-Spring. The 57th IEEE Semiannual*, vol. 3, 22-25 April 2003, pp. 1629-1633.

[8] N. Wiberg, "Codes and decoding on general graphs," Ph.D. dissertation, Linköping University, S-581 83 Linköping, Sweden 1996.

[9] R. M. Tanner, "A recursive approach to low-complexity codes," *IEEE Transactions on Information Theory*, vol. 27, no. 5, pp. 533-547, September 1981.

[10] F. R. Kschichang, B. J. Frey, and H. A. Loeliger, "Factor graph and the sum-product algorithm," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498-519, February 2001.

[11] H. A. Loeliger, "An introduction to factor graphs, " *IEEE Signal Processing Magazine*, vol. 21, issue 1, pp. 28-41, January 2004.

[12] K. M. Chugg, A. Anastasopoulos, and X. Chen, *Iterative Detection: Adaptivity, Complexity Reduction, and Applications*. Norwell, Massachusetts 02061: Kluwer Academic Publisher, 2001.

[13] A. Papoulis, and U. Pillai, *Probability, Random Variables and Stochastic Processes*, 4th ed. McGraw-Hill, 2002.

[14] J. K. Holmes, *Coherent Spread Spectrum Systems*. Wiley-Interscience Publications, USA 1982.

[15] F. Principe, K. M. Chugg, and M. Luise, "Rapid acquisition of Gold codes and related sequences using iterative message passing on redundant graphical models," in *Proc. MILCOM 2006 Conference*, Washington DC (USA), 23-25 October 2006.

[16] O. W. Yeung, and K. M. Chugg, "An iterative algorithm and low complexity hardware architecture for fast acquisition of long PN codes in UWB systems," *Springer J. VLSI and Signal Processing (Special issue on UWB Systems)*, vol. 43, April 2006.

[17] R. G. Gallager, "Low-density parity-check codes," *IRE Transactions on Information Theory*, pp. 21-28, January 1962.

[18] C. Berrou, A. Glavieux, and P. Thitmajshima, "Near Shannon limit error-correcting coding and decoding: turbo-codes (1)," in *Proc. Int. Conf. Commun.*, Geneva, Switzerland, May 1993, pp. 1064-1070.

[19] P. Robertson, E. Villebrun, and P. Hoeher, "A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log-domain," in *Proc. IEEE International Conference on Communications (ICC 95 Seattle)*, vol. 2, 18-22 June 1995, pp. 1009-1013.

[20] M. Aparicio and *et al.*, *Global Positioning Systems: Theory and Applications*, B. W. Parkinson and J. J. Spilker, Eds. 370 L'Enfant Promenade, SW, Washington, DC 20024-2518: American Institute Of Aeronautics, Inc., 1996, vol. I and II.

[21] Galileo Joint Undertaking, *Galileo Open Service Signal in Space Interface Control Document (OS SIS ICD) – Draft 0*, 23 May 1006, http://www.galileoju.com/page2.cfm#.