A New Approach to Rapid PN Code Acquisition Using Iterative Message Passing Techniques

Keith M. Chugg, *Member, IEEE,* and Mingrui Zhu

***Abstract*—Iterative message passing algorithms on graphs, which are generalized from the well-known turbo decoding algorithm, have been studied intensively in recent years because they can provide near-optimal performance and significant complexity reduction. In this paper, we demonstrate that this technique can be applied to pseudorandom code acquisition problems as well. To do this, we represent good pseudonoise (PN) patterns using sparse graphical models, then apply the standard iterative message passing algorithms over these graphs to approximate maximum-likelihood synchronization. Simulation results show that the proposed algorithm achieves better performance than both serial and hybrid search strategies in that it works at low signal-to-noise ratios and is much faster. Compared with full parallel search, this approach typically provides significant complexity reduction.**

***Index Terms*—Loopy graphical models, message passing, pseudonoise (PN) code acquisition, ultrawideband (UWB).**

I. Introduction

SPREAD-SPECTRUM (SS) techniques are used in many military communication systems to provide some combination of ranging capabilities, anti-jam protection, low probability of detection and/or interception, and multiple-access capability. A common form of SS is direct-sequence spread-spectrum (DS/SS) in which the transmitter multiplies a binary data sequence by a higher rate pseudorandom or pseudonoise (PN) binary sequence. This procedure is referred as *spreading* because it results a binary signal occupying a much wider spectrum than the original data. Other SS methods, such as frequency hopping (FH) and hybrid DS-FH are also commonly used in military systems. Ultrawideband (UWB) systems are extreme cases of SS and are often characterized by low duty cycle trains of very narrow pulses. In all of these cases, spreading is achieved via a pseudorandom sequence. To enable autonomous reception, periodic PN sequences are used. For military communication systems, long PN sequences (e.g., long period) are desirable as the use of shorter PN sequences makes the link susceptible to repeat-back jamming or interception/detection via delay and correlate methods [2].

At the receiver's side, *despreading* must be performed before the demodulation of the data sequence. This is accomplished by generating a local replica of the PN code and synchronizing it to the one that is embedded in the received signal. Thus, quickly achieving and then maintaining PN code synchronization is critical because even a small misalignment can cause catastrophic signal-to-noise ratio (SNR) degradation. Typically, this task is performed in two steps: PN code acquisition, where a coarse alignment of the two PN codes is produced to within one code-chip interval, and code tracking. The SNR of the observations during this acquisition process is very low since the processing gain has not yet been realized prior to despreading.

The most widely used and studied methods for acquiring PN sequences are full parallel search, serial search [3], and hybrid search [2]. In each of these, correlations between the incoming, noisy SS waveform and the locally generated reference are formed. In order to acquire a PN code with long period quickly, the time duration of these correlations must be a small fraction of the whole PN code sequence. In the full parallel case, correlations are formed for all possible PN code alignments so that the minimum time to achieve reliable acquisition is determined by how long one must correlate to reliably detect the correct alignment. This is the maximum-likelihood (ML) decision for the PN code phase based on the set of observations. Since the number of correlations needed for full parallel search is the period of the PN sequence, this method is infeasible for military systems using very long PN codes. Simple serial search represents the other extreme wherein only one of the correlations used in full-parallel search is formed and a threshold test is performed to determine if it is the correct alignment. If the threshold test fails, another set of observations is collected and used to correlate against another reference PN code alignment. Since many such tests are required, simple serial search provides relatively slow acquisition for long PN codes. Hybrid search tests a small set of possible alignments in parallel and then repeats this test on another set of observations until the correct alignment is discovered.

Full parallel search is fast to acquire, but complex. Serial search is simple, but slow to acquire. Hybrid search provides, at best, a linear-scale tradeoff between these two extremes (e.g., a hybrid search with four parallel correlators is four times faster and four times as complex as serial search). In this paper, we present the first method for achieving PN acquisition at low SNR as fast as full-parallel search, but with significantly lower complexity.[1] Our approach is based on the paradigm of message-passing on graphical models and more specifically, iterative message passing algorithms (iMPAs) and graphical models with cycles [5]–[8]. This is a generalization of the "turbo" de-

Manuscript received April 1, 2004; revised January 6, 2005. This work was supported in part by the Army Research Office under Grant DAAD19-01-1-0477. This paper was presented in part at the IEEE MILCOM Conference 2003, Boston, MA.

The authors are with the Communication Sciences Institute, Department of Electrical Engineering, University of Southern California, Los Angeles, CA 90089-2565 USA (e-mail: chugg@usc.edu; mingruiz@usc.edu).

Digital Object Identifier 10.1109/JSAC.2005.845424

[1]Sequential search [4] is a suboptimal method for fast acquisition, but is highly vulnerable to noise and interference signals [2]. It is not widely used and will not be discussed in this paper.

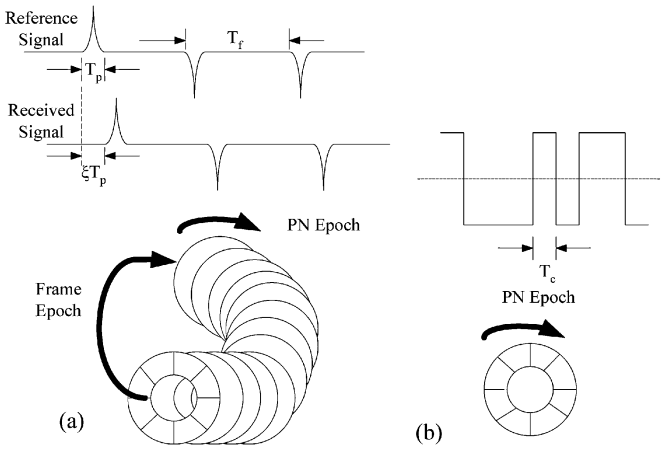

Fig. 1. Sample waveform and diagram of the associated PN acquisition problem for two spread-spectrum systems. (a) A low duty cycle UWB system where the frame epoch and PN code phase must be determined. (b) A direct-sequence system where only PN code phase need be acquired. The DS system is modeled with the complex baseband equivalent signal.

coding algorithm [9]. In this approach, no correlation to a single PN reference signal is computed explicitly. Rather, the global structure in the PN sequence is modeled as a set of coupled local constraints and correlations are formed against these incomplete, local structures. This may be viewed as an approximation to ML acquisition, much in the same way that a turbo decoder is an approximation of the ML decoder for a concatenated code. Therefore, our approach suffers a small performance degradation relative to full parallel search.

Our primary motivation for this problem is a UWB system using long PN sequences, for which fast PN acquisition is a critical necessity. To illustrate this, consider a low duty cycle train of narrow pulses with PN sequence randomization received in noise[2]

$$r(t) = \sum_{k=0}^{M-1} \sqrt{E_c}(-1)^{x_k}\omega_r(t - kT_f - \xi T_p) + n(t) \qquad (1)$$

where E_c is the energy per pulse ("chip") $\omega_r(t)$ of duration $T_p, x_k \in \{0,1\}$ is a PN code pattern, T_f is the *frame time* or time between pulses, ξ is an unknown shift or *frame epoch*, and $n(t)$ is additive noise. A sample waveform for a noise-free UWB signal of this form is shown in Fig. 1(a). The PN acquisition problem is also diagrammed in Fig. 1 in terms of a search over potential timing bins. This UWB synchronization problem is more difficult than the corresponding classical DS/SS problem because the frame epoch must be acquired simultaneously with the PN pattern. The number of candidate frame epochs to be searched is on the order of $T_f/T_p \gg 1$, and for each of these a complete PN acquisition search is required. The search bins for a PN acquisition problem are commonly diagramed with a "PN phase wheel," as shown in Fig. 1(b), corresponding to one period of the PN code. The corresponding diagram for the UWB system is the "PN-phase/frame-epoch taurus" shown in Fig. 1(a).

For UWB systems with long PN codes, extremely fast PN acquisition is required. This is not only due to the high level of timing uncertainty described above, but also the fact that the true frame epoch will certainly drift due to oscillator imperfection and/or platform mobility. More specifically, if the bins in Fig. 1(a) are tested sequentially and the frame epoch is drifting, it is possible that the search will never locate the true epoch—i.e., this may result in a "chasing one's tail" situation. Therefore, for a fixed, hypothesized frame epoch, it would be desirable to search all possible PN pattern phases in parallel. It is also desirable to complete this search based on a relatively small number of observations and with reasonable implementation complexity. The method presented in this paper provides an attractive solution to this problem that cannot be achieved using traditional PN acquisition strategies. Similar methods have been applied to the sparse intersymbol interference (S-ISI) channels by Chen [11], [12, Ch. 3].

This paper is organized as follows. Section II contains the signal models considered, Section III contains approximate analysis of the traditional approaches to PN acquisition, and Section IV describes the graphical modeling and iMPAs applied to PN acquisition. Simulation results are provided in Section V and conclusions are drawn in Section VI.

II. Signal Models

Linear feedback shift register (LFSR) sequences having the maximum possible period for a r-stage shift register are called maximal-length sequences or m-sequences [13]. They have been successfully employed in a wide range of SS systems and many other spreading codes can be derived from them. A binary r-stage LFSR is shown in Fig. 2(a). At time k, let x_k be the output, so that $x_{k+i}, 0 \le i \le r-1$ is the value of the ith register and the constraint is

$$0 = g_0 x_{k+r} \oplus g_1 x_{k+r-1} \oplus \cdots \oplus g_{r-1} x_{k+1} \oplus g_r x_k \qquad (2)$$

where $\oplus$ is modulo 2 addition and $g_i \in \{0,1\}, 0 \le i \le r$, are feedback coefficients. The *generating polynomial* is $g(D) = g_0 + g_1 D + \cdots + g_{r-1} D^{r-1} + g_r D^r$, where D is the unit delay operator [13]. The maximum achievable period of a r-stage LFSR is $N = 2^r - 1$ and is achieved for primitive $g(D)$ when the initial register contents are not all zero. Note that for primitive $g(D), g_0 = g_r = 1$ holds. The (infinitely long) periodic output sequence $\underline{x}$ generated then can be written as $\underline{x} = x_0, x_1, \ldots, x_{r-1}, x_r, x_{r+1}, \ldots, x_{N-1}, x_N, \ldots$, where $x_{N+i} = x_i$. In fact, this LFSR is a finite-state machine (FSM), with evolution determined entirely by the initial contents of the registers, or the initial FSM state. Specifically, the initial FSM state is the $(r \times 1)$ vector $\mathbf{u} = (x_0, x_1, \ldots, x_{r-1})^T$, where T denotes transposition.

The goal of code acquisition is to find the phase of the sequence present in the received signal, where $\underline{x}, D\underline{x}, \ldots, D^{N-1}\underline{x}$ are defined as phases of $\underline{x}$. In most practical scenarios with long PN codes, only part of this long sequence is observable, so the problem can be stated as: for a given number of M noisy observations $\{z_k\}_0^{M-1}$ estimate the initial state $\mathbf{u}$. Also, the number

[2]In this paper, we focus on this model where the PN randomization is done by antipodal modulation of the pulses, which has been used in the UWB prototype proposed by Berkeley Wireless Research Center [10]. Other methods use pulse position modulation (PPM) by the PN sequence. In our method, one needs the likelihood of the chip value for a given noisy observation, so application to PPM methods and other models is straightforward.

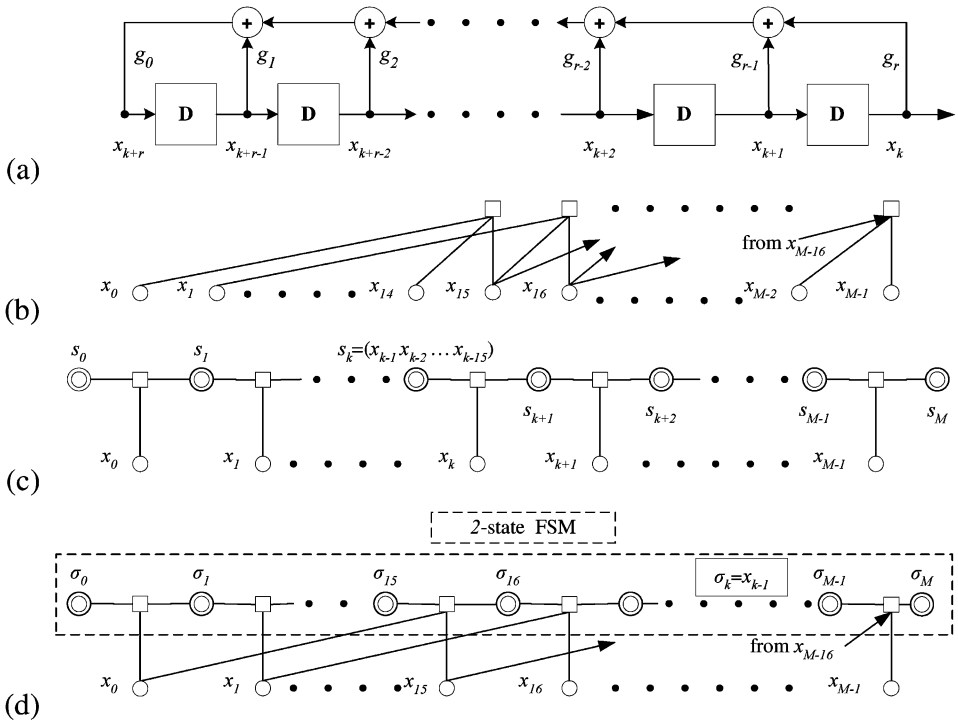

Fig. 2. Methods for modeling LFSRs. (a) Shows the generator diagram for an r-stage LFSR. (b)–(d) Shows different graphical models for the same 15-stage LFSR with $g(D) = 1 + D + D^{15}$.

of observations is much larger than the length of the shift register, but much less than the period (i.e., $r \ll M \ll N$). A simplified model for these observations is

$$\begin{aligned} z_k &= \sqrt{E_c} y_k(\mathbf{u}) \cdot e^{j\theta_c} + n_k \\ &= \sqrt{E_c}(-1)^{x_k(\mathbf{u})} \cdot e^{j\theta_c} + n_k, \quad 0 \le k \le M-1. \end{aligned} \tag{3}$$

This model captures both the DS/SS and UWB systems illustrated in Fig. 1, where E_c is the signal energy per chip (pulse) and n_k is complex circular additive white Gaussian noise (AWGN) having variance $N_0/2$ for each of the real and imaginary parts. The term with θ_c is applicable only to the traditional DS/SS system and models the effect of an unknown carrier phase, assumed to be constant over the observation interval. We have explicitly denoted the dependency of x_k and the corresponding anitpodally modulated y_k on the initial state of the LFSR $\mathbf{u}$. This model is simplified because it does not consider the effects of oversampling the chip rate, potential frequency offsets, jammers, etc. Nonetheless, this is the standard model used for basic characterization of PN acquisition algorithms [2]. The model in (3) can be written in vector form as

$$\mathbf{z} = \sqrt{E_c}\mathbf{y}(\mathbf{u})e^{j\theta_c} + \mathbf{n} \tag{4}$$

where $\mathbf{n} = (n_0, n_1, \ldots, n_{M-1})^T$ is a complex circular Gaussian vector with zero mean and covariance matrix $(N_0/2)\mathbf{I}_M$ for each the real and imaginary parts, where $\mathbf{I}_M$ is the $M \times M$ identity matrix.

Since the simple model in (3) is common in the DS/SS literature, let us consider how it applies to the UWB system modeled in (1), where $n(t)$ is AWGN with power spectral density level of $N_0/2$. Using an estimate the frame epoch $\hat{\xi}$, the discrete observation z_k is obtained by the following processing: during time interval $[kT_f, (k+1)T_f)$, the UWB receiver aligns a pulse matched filter at $kT_f + \hat{\xi}T_p$, and samples the output at $kT_f + (\hat{\xi}+1)T_p$. If ξ is constant over the time interval $[0, M \cdot T_f]$ and $\hat{\xi} = \xi$, a model of the form (3) results. Specifically, since the UWB signal in (1) has no sinusoidal carrier, the real part of z_k from (3) is obtained with $\theta_c = 0$. In this example, the chip interval T_c equals T_f, the frame time, as there is only one pulse every T_f s. Therefore, while we will characterize acquisition time in terms of the number of chips observed, this value should be interpreted appropriately for the UWB and DS/SS cases.

In a DS/SS system, θ_c is typically unknown at the point of PN acquisition because the SNR before despreading is too low to enable carrier phase synchronization and PN acquisition is performed noncoherently. In this case, T_c is the time duration of a single PN chip and θ_c is modeled as a random variable uniformly distributed over $[0, 2\pi]$. which is constant over the duration of M observations.

For compactness, we will use (3)–(4) for these two cases: 1) the DS/SS system with no carrier phase knowledge and 2) the UWB system. Note that if one had knowledge of θ_c for the DS/SS case, the model would be the same as the model adopted for the UWB system.

III. Performance Characteristics of Traditional PN Acquisition Algorithms

As briefly described in Section I, the three traditional approaches to PN acquisition all form correlations between the noisy observation and a local reference generated with a hypothesized PN phase (i.e., despreaders). Specifically, for the

chip-spaced model in (3) there are N possible PN phases which we denote by $\mathbf{u}_i, 0 \le i \le N-1$. The correlation to the ith PN phase using M observations is

$$r_i = r(\mathbf{u}_i) = \frac{1}{M}\mathbf{y}^T(\mathbf{u}_i)\mathbf{z} \tag{5}$$

where $\mathbf{y}(\mathbf{u}_i)$ is the noise-free signal in (4) with the actual initial state replaced by the hypothesized state $\mathbf{u}_i$. The correlation statistic r_i comprises two parts: a Gaussian noise term and a partial-period PN autocorrelation [13] term of the form $\mathbf{y}^T(\mathbf{u}_i)\mathbf{y}(\mathbf{u}_j)e^{j\theta_c}$. The autocorrelation properties of m-sequences imply that this is nearly zero for $i \neq j$. Therefore, the statistic in (5) is similar to the correlator output for a detector for N-ary orthogonal modulation in AWGN and methods similar to those used in evaluating the performance of orthogonal modulations can be employed for the analysis of traditional PN acquisition algorithms.

Without loss of generality, assume that the actual initial state is $\mathbf{u}_0$, so that

$$r_0 = \frac{1}{M}\mathbf{y}^T(\mathbf{u}_0)\mathbf{z} = \sqrt{E_c}e^{j\theta_c} + \omega_0 \tag{6}$$

$$r_i = \frac{1}{M}\mathbf{y}^T(\mathbf{u}_i)\mathbf{z} = \sqrt{E_c}\frac{1}{M}\mathbf{y}^T(\mathbf{u}_i)\mathbf{y}(\mathbf{u}_0)e^{j\theta_c} + \omega_i \quad 1 \le i \le N-1 \tag{7}$$

where the independent identical distributed (i.i.d.) sequence ω_i is complex circular Gaussian with real and imaginary parts having zero mean and variance $(N_0/2M)$. For m-sequences it can be shown [13], [2] that the set of random variables $\{r_i\}_1^{N-1}$ can be approximately modeled as independent identically distributed (i.i.d.) zero-mean complex Gaussian random variables with variance $(2 \cdot E_c + N_0)/(2M)$ in the imaginary and real parts. Specifically, the nonzero partial-period correlation between PN phases has been modeled as a small amount of additional Gaussian noise. This approximation is used throughout the analysis that follows in this section and is justified numerically in Section V.

A. Full Parallel Search

Full parallel search finds the ML estimate of the initial state through exhaustive search over the N possible values, yielding the estimate $\hat{\mathbf{u}} = \arg\max_{\mathbf{u}_i} p(\mathbf{z} \,|\, \mathbf{u}_i)$, where $p(\mathbf{z} \,|\, \mathbf{u}_i)$ is the likelihood of $\mathbf{u}_i$ and $\mathbf{z}$ is defined in (4). The acquisition time for full parallel search is just the observation length M, but the memory requirements and computational complexity both grow linearly in N, which increases exponentially with the length of the LFSR.

The probability of correct acquisition, P_{ACQ}, for full parallel search can be computed approximately using the model in (6) and (7), since the set of correlations $\{r_i\}_{i=0}^{N-1}$ is a set of sufficient statistics for the model of (3). More precisely, for UWB systems, $v_i = \Re\{r_i\}$ with $\theta_c = 0$ are the variables to be compared, while for the DS/SS with unknown θ_c, $|r_i|$ is the relevant test statistic. In the former case, this is the output of a depredator and in the latter case, this is the output of an in-phase/quadrature (I/Q) despreader followed by an envelope detector.

For the UWB system, correct acquisition is declared only when v_0 is the largest correlator output so that

$$P_{\mathrm{ACQ}}^{(C)} = \int_{-\infty}^{\infty} P(v_1 < v_0, \ldots, v_{N-1} < v_0 \,|\, v_0) p(v_0)\, dv_0 \approx \int_{-\infty}^{\infty} \left[1 - Q\left(\frac{t + \sqrt{\frac{2ME_c}{N_0}}}{\sqrt{\frac{2E_c}{N_0} + 1}}\right)\right]^{N-1} \frac{e^{-\frac{t^2}{2}}}{\sqrt{2\pi}}\, dt \tag{8}$$

where $Q(\cdot)$ is the complementary cumulative distribution function of a standard Gaussian random variable, defined as $Q(t) = \int_t^{\infty}(e^{-u^2/2})/(\sqrt{2\pi})du$.

The probability of acquisition of noncoherent full parallel search can be computed using the same approximation in (6) and (7). Correct acquisition is declared only when $|r_0|$ is larger than all other $|r_i|$, so that, via methods similar to those employed for analyzing noncoherent orthogonal modulations, we obtain

$$P_{\mathrm{ACQ}}^{(\mathrm{NC})} \approx \int_0^{\infty} \left[1 - e^{\frac{-t^2}{\frac{4E_c}{N_0}+2}}\right]^{N-1} t \times \exp\left[-\frac{t^2}{2} - \frac{ME_c}{N_0}\right] \mathrm{I}_0\left[\sqrt{\frac{2ME_c t}{N_0}}\right] dt \tag{9}$$

where $\mathrm{I}_0(\cdot)$ is the modified Bessel function of zeroth order [14, Ch. 2].

B. Simple Serial Search

For the simplified model in (3), simple serial search computes the likelihood for one candidate initial phase $p(\mathbf{z} \,|\, \mathbf{u}_i)$ using the set of M observation $\mathbf{z}$ [3]. More precisely, for the UWB, the real part of r_i is compared with a threshold, and for the noncoherent DS/SS case $|r_i|$ is compared with a threshold. If the threshold is not exceeded, the current set of M observations is discarded, and correlation over another M observations is computed to test another initial state.[3] In this case, the M observations correspond to one *dwell time* [3], T_d and are assumed to be nonoverlapping. This process continues until acquisition is declared.

Simple serial search reduces the memory requirements significantly and works well at low SNR. However, it is slow since one needs to try roughly half of the possible PN alignments in order to locate the correct one. More formally, without *a priori* information on the PN phase, the mean acquisition time is [3]

$$T_{\mathrm{ACQ}}^{(s)} = \frac{2 + (2 - P_D)(N-1)(1 + KP_{\mathrm{FA}})}{2P_D} \cdot T_d \tag{10}$$

where P_D is the probability of detection for a single-dwell test, P_{FA} is the probability of false alarm, and K is the penalty time for a false alarm, measured in dwell times. Considering the most optimistic case, where $P_D = 1$ and $P_{\mathrm{FA}} = 0$, we have $(T_{\mathrm{ACQ}})/(T_d) = (N+1)/(2) = 2^{r-1}$. So, unlike full parallel search, simple serial search takes much more than M chip times to acquire on average.

[3]The reference state must be adjusted for the fact that the tests take place on different observation sets and the actual PN phase has continued to evolve.

Also, it can be shown that[4]

$$P_{\mathrm{ACQ}}^{(s)} = \frac{P_D \cdot [1-(1-P_{\mathrm{FA}})^N]}{NP_{\mathrm{FA}}} \tag{11}$$

where P_D and P_F are the probability of detection and false alarm, respectively, for a single dwell test. Specifically, $P_D(P_{\mathrm{FA}})$ is the probability that the threshold is exceeded when the correct (incorrect) PN phase is used. These can be computed using the same method employed to obtain (8) and (9).

C. Hybrid Search

Hybrid (serial/parallel) search uses C_p parallel correlators to test phases in parallel. Like serial search, multiple dwells on different observation sets are generally required. The performance is again a function of the single dwell probabilities of detection and false alarm

$$P_{\mathrm{FA}}^{(h)} = 1-(1-P_{\mathrm{FA}})^{C_p} \tag{12}$$

$$P_D^{(h)} = 1-(1-P_D)(1-P_{\mathrm{FA}})^{C_p-1} \tag{13}$$

$$T_{\mathrm{ACQ}}^{(h)} = \frac{2-P_D-(1-P_D)(C_p-1)P_{\mathrm{FA}}}{2[P_D+(1-P_D)(C_p-1)P_{\mathrm{FA}}]} \cdot \frac{1+KC_pP_{\mathrm{FA}}}{C_p} \cdot (NT_d) \tag{14}$$

where $P_{\mathrm{FA}}^{(h)}, P_D^{(h)}$, and $T_{\mathrm{ACQ}}^{(h)}$ are the false alarm probability, global detection probability and mean acquisition time of C_p-correlator hybrid search respectively. If C_p is small ($C_p \ll N$) and the false alarm penalty is neglected, $P_{\mathrm{FA}}^{(h)} \simeq C_pP_{\mathrm{FA}}, P_D^{(h)} \simeq 1-(1-P_D)[1-(C_p-1)P_{\mathrm{FA}}]$, and $T_{\mathrm{ACQ}}^{(h)} \simeq (1/C_pT_{\mathrm{ACQ}}^{(s)})$. Therefore, hybrid search can only trade complexity with mean acquisition time linearly. Furthermore, when C_p is sufficiently large, the false alarm penalty will dominate and no further improvement in $T_{\mathrm{ACQ}}^{(h)}$ will be achieved [2]. The probability of acquisition of hybrid search can be obtained using (11), with P_D, P_{FA}, and N replaced by $P_{\mathrm{FA}}^{(h)}, P_D^{(h)}$, and N/C_p, respectively.

IV. Graphical Models of M-Sequences and iMPAs for PN Acquisition

Graphical modeling and iterative message-passing algorithms have become widely applicable to inference problems in communications and signal processing, most notably decoding of modern error correction codes. A graphical model captures constraints on variables by connecting *variable nodes* to *configuration check nodes* that constrain the configurations of the connected variables.[5] For example, consider the set of m-sequence outputs $\{x_k\}_{k=0}^{M-1}$. One graphical model is a single check node with these M binary variables connected. While there are 2^M possible combinations of these binary variables, the check node enforces the constraint that only $N = 2^r - 1$ of these are allowable configurations. There are other graphical models that can enforce the same set of constraints. These are obtained by factoring this global constraint (i.e., involving all variables) into a sets of interdependent check nodes, each enforcing only local constraints (i.e., involving only a subset of variables). An example of this is shown in Fig. 2(b) for the m-sequence with generating polynomial $g(D) = 1 + D + D^{15}([100\,003]_8)$ of degree 15, where we use the convention that variable nodes are circles and check nodes are squares. Note that each check node enforces the constraint that $x_k \oplus x_{k-1} \oplus x_{k-15} = 0$ for the appropriate value of k. Thus, the number of valid local configurations is 4—i.e., $(x_k, x_{k-1}, x_{k-15}) \in \{(0,0,0),(0,1,1),(1,0,1),(1,1,0)\}$. In general, let the number of allowable configurations at a check node be C and index these by a variable c—e.g., $C = 4$ and the four configurations correspond to $c = 0, 1, 2, 3$, respectively.

For a given graphical model, there is a well-defined message-passing algorithm that repeatedly passes messages across edges in both directions. The MPA combines and marginalizes messages on variables over the constraints associated with the check nodes. Specifically, each check node will accept incoming messages, characterizing some form of soft-decision information, on the variables connected to it. These messages, which are sent from connected variable nodes, are then combined to obtain soft-decision information (metrics) on all valid local configurations. Finally, these local configuration metrics are marginalized to produce output metrics. Variable nodes with more than one connection can be viewed as incorporating an equality constraint as will become evident.

As a specific example, consider the graph in Fig. 2(b) and assume the UWB model. Then there is initial chip-level soft-decision channel information of the form $\mathrm{M_{ch}}[x_k] = -\ln(p(z_k \,|\, x_k)$ at the variable node for x_k. These become the initial input messages for all three check nodes connected to x_k. Under this convention, a large message means that the conditional value for x_k is highly unlikely and small message corresponds to high confidence in that conditional value. Therefore, we use the term metric and message interchangeably in the following. Focusing on a check node constraining (x_k, x_{k-1}, x_{k-15}), let the incoming message on x_i be $\mathrm{MI}[x_i]$. Note that for each variable the message is a list of numbers for each conditional value of the variable—e.g., in this case $\mathrm{MI}[x_i]$ is shorthand for a list of two numbers: $\mathrm{MI}[x_i = 0]$ and $\mathrm{MI}[x_i = 1]$. With the valid configurations indexed by $c, x_i(c)$ is defined for each of these configurations. The processing associated with a configuration check node can be viewed as a two step process

$$\mathrm{M}[c] = \sum_i \mathrm{MI}[x_i(c)] \quad \text{(combining)} \tag{15}$$

$$\mathrm{MO}[x_i] = \min_{c:x_i} \mathrm{M}[c] - \mathrm{MI}[x_i] \quad \text{(marginalization)} \tag{16}$$

[4]It is assumed that the system acquires within one single search of the N possible PN alignments. If this were not the case, threshold tests are not necessary and a full search could be achieved [2]. Also, an absorbing false alarm state [2] is assumed.

[5]The graphical convention adopted is explicit in time index, so that, for example, x_{10} and x_{11} are distinct nodes, but implicit in value, so that, for example, $x_{10} = 0$ and $x_{10} = 1$ are captured in one variable node. This differs from trellis diagrams which are explicit in both time index and variable value.

where $c : x_i$ means all configurations consistent with the conditional value x_i. For example, the output message for x_k produced by the check node constraining (x_k, x_{k-1}, x_{k-15}) is

$$\begin{aligned}\mathrm{MO}_{\mathrm{cc}}[x_i = 0] = \min\{&\mathrm{MI}_{\mathrm{cc}}[x_{k-1} = 0] + \mathrm{MI}_{\mathrm{cc}}[x_{k-15} = 0];\\ &\mathrm{MI}_{\mathrm{cc}}[x_{k-1} = 1] + \mathrm{MI}_{\mathrm{cc}}[x_{k-15} = 1]\}\\ \mathrm{MO}_{\mathrm{cc}}[x_i = 1] = \min\{&\mathrm{MI}_{\mathrm{cc}}[x_{k-1} = 0] + \mathrm{MI}_{\mathrm{cc}}[x_{k-15} = 1];\\ &\mathrm{MI}_{\mathrm{cc}}[x_{k-1} = 1] + \mathrm{MI}_{\mathrm{cc}}[x_{k-15} = 0]\}\end{aligned}$$

which uses the fact that configurations $c = 0, 1$ are consistent with $x_k = 0$ and $c = 2, 3$ are consistent with $x_k = 1$, and input (output) messages to the configuration check node have been denoted by $\mathrm{MI}_{\mathrm{cc}}[\cdot]$ ($\mathrm{MO}_{\mathrm{cc}}[\cdot]$).

As mentioned, variable nodes connected to multiple check nodes have an implicit equality constraint, so that message updates take place at variable nodes too. Specifically, consider a variable x and suppose that this variable is connected to L check nodes and that $\mathrm{MI}_v[x^{(l)}]$ is the incoming message from the lth check node to a variable node, then the output message returned to the lth check is

$$\mathrm{MO}_v\left[x^{(l)}\right] = \mathrm{M}_{\mathrm{ch}}\left[x = x^{(l)}\right] + \sum_{m=0, m\neq l}^{L-1} \mathrm{MI}_v\left[x^{(m)} = x^{(l)}\right] \tag{17}$$

which should be interpreted as an equation for each conditional value of x—e.g., for binary x, one for $x^{(l)} = 0$ and another for $x^{(l)} = 1$. Note that this is equivalent to (15) and (16), where each valid configuration corresponds to all connected variables taking the same value. As a concrete example, consider the variable node for x_k in Fig. 2(b), which is connect to three checks constraining (x_k, x_{k-1}, x_{k-15}), (x_{k+1}, x_k, x_{k-14}), and $(x_{k+15}, x_{k+14}, x_k)$. This node has a channel message $\mathrm{M}_{\mathrm{ch}}[x_k]$ and three messages that were output from the previous activation of the connected check nodes. The variable node will return to a given check node the sum of the messages from the other two checks and the channel metric.

The message update (15)–(17) are general and define the processing for all standard MPAs. There are different choices for the format of the messages and the combining and marginalization operators. In the above discussion, we used messages in the form of negative-log of probabilities and min-sum marginalization and combining. In the numerical results, we also consider $\min^*$-sum marginalization and combining where

$$\min^*(x, y) = \min(x, y) - \ln(1 + e^{|x-y|}). \tag{18}$$

Specifically, $\min^*$-sum algorithms perform the processing in (15)–(17) with the min operators replaced by $\min^*$ operators.

While the above defines the processing associated with message updating, in order to specify a MPA, one must define the graph (connectivity and constraint definitions) and an *activation schedule*, which is the order that the variable nodes and check nodes are activated, including when the processing is terminated. When the algorithm terminates, hard decision information can be inferred from the messages by selecting the conditional value with smallest metric. A basic result in this area is that if a graph has no cycles, then there is a schedule for which the MPA is optimal. In other words, by repeatedly updating messages using simple local constraints, one can compute the same messages that would be computed using a single global constraint. The advantage is that the processing of many local constraints can be much smaller than that associated with a single global constraint. Roughly, any activation schedule that passes messages from each node to all other nodes on a cycle-free graph is optimal and the MPA converges to the same result that would have been obtained by processing the global constraint directly.

When the graphical model has cycles, the same message updating rules can be used, but the approaches are suboptimal heuristics, which we refer to as iMPAs. Specifically, little has been proven about the convergence properties and the long-term evolution of the messages for these algorithms when cycles are present. It has been observed empirically, however, that iMPAs are very effective and often yield performance near that of the optimal solution. Empirical results suggest that the iMPA heuristic is most effective when there are no very short cycles and when the cycle structure is highly irregular (i.e., pseudo-random). The advantage of using graphs with cycles is that the complexity of the resulting iMPA can be significantly less than that of any MPA associated with a cycle-free graphical model. In the m-sequence example, the global constraint has $N = 2^r - 1 = 32\,767$ configurations, while the graph of Fig. 2(b) has $M - 15$ check nodes, each having four valid configurations. For cases of practical interest $4(M - 15)$ is much less than N, so that message passing on the graph in Fig. 2(b) may yield significantly lower complexity.

The graphical model associated with a particular set of constraints is not unique and selecting different models will yield a different MPA. One way to alter a graph is to include *hidden* variables that are neither the input nor output of the system.[6] For example, the same m-sequence modeled in Fig. 2(b) can be modeled by the cycle-free graphical model in Fig. 2(c), in which the hidden variables s_k, indexing all values of $(x_{k-1}, x_{k-2}, \ldots, x_{k-15})$, have been added and are denoted with double-lined circles to distinguish them from the output variables. These hidden variables are simply the state of the FSM that represents that LFSR. An optimal MPA algorithm on this graph is known as the *forward-backward algorithm* (FBA) [12]. In the FBA, messages are sent forward (left to right) starting at s_0 and ending at s_M, and then backward from s_M to s_0. This is one activation schedule that results in an optimal MPA and further activation of the check nodes does not change the message values. It follows from the definition of the nonzero s_k for an m-sequence that each state takes $2^r - 1$ values and each local check node has $2^r - 1$ valid configurations. In fact, at the end of the forward recursion, the messages at s_M are the $N = 2^r - 1$ correlations computed by the full parallel search approach to PN sequence acquisition. This illustrates the importance of cycles in the graphical model to achieve low complexity iMPAs.

[6]The channel messages for these variables are taken to be zero for all conditional values.

A third graphical model for the m-sequence with $g(D) = 1 + D + D^{15}$ is shown in Fig. 2(d), where hidden variables $\sigma_k = x_{k-1}$ are added and the check nodes enforce the constraint $\sigma_k \oplus \sigma_{k+1} \oplus x_{k-15} = 0$ and $\sigma_{k+1} = x_k$. The three graphs shown in Fig. 2 all completely capture the constraint of the m-sequence structure fully and without redundancy. The graph in Fig. 2(d) can also be viewed as decomposing the 15-stage shift register into a two-stage shift register with a long delayed, feedback loop. This is emphasized by the box in Fig. 2(d) that outlines the subgraph corresponding to an FSM with state σ_k. A natural iMPA schedule for this graph is to activate the variable nodes to set the transition metrics of the FSM subgraph, then run the FBA on the two-state FSM subgraph, then send messages back to the variable nodes. This will be considered one iteration. The details of this iMPA are given in the Appendix.

For completeness, the schedule for the iMPA algorithm run on the graph in Fig. 2(b) will be to activate all variable nodes in parallel, then all check nodes in parallel, etc. One activation of all check and variable nodes will be defined as one iteration.

A final, hard decision on the variable x_k is obtained using the soft decision

$$\mathrm{M}[x_k] = \mathrm{M}_{\mathrm{ch}}[x_k] + \sum_{m=0}^{L-1} \mathrm{MI}[x^{(m)} = x_k] \quad 0 \le k \le M-1 \tag{19}$$

which is the channel metric plus all incoming messages to the variable node x_k. Specifically, if $\mathrm{M}[x_k = 1] < \mathrm{M}[x_k = 0]$, then $\hat{x}_k = 1$ is decided, otherwise, $\hat{x}_k = 0$ is decided. We modify this standard approach slightly for the PN acquisition problem. Specifically, this method can be used to obtain a hard decision on x_k for all M time indices. Ideally, these decisions would all be consistent with the same initial state $\mathbf{u}$, but this is not always observed. Note that decisions on x_k for any r-consecutive time indices imply a decision for the initial state and such decisions can be made at any iteration. Thus, to provide better performance, $\lfloor M/r \rfloor$ estimates of the initial state are obtained by using $\lfloor M/r \rfloor$ nonoverlapping r-variable intervals at each iteration. The iMPA is stopped after a maximum number of iterations and the state estimate that appears most frequently is selected as the final decision for the initial state.

A. Graphical Models for Other m-Sequences

Careful inspection of the above development implies that our approach is most desirable when the generating polynomial is sparse, i.e., there are only a few ones in $g(D)$. For example, considering graphical models of the form shown in Fig. 2(b), the number of configurations for each check node grows exponentially with the number of nonzero feedback coefficients in $g(D)$ and the number of cycles also increases with this parameter. Some examples from [15] are listed in Table I.

There are many graphical models for a given set of constraints and there is no systematic procedure for specifying a good cyclic graphical model—i.e., one that will yield an iMPA with low complexity and good performance. Although this process remains more art than science, we illustrate the technique further by considering other generating polynomials and potential loopy graphical models. A graphical model with no hidden variables of the form shown in Fig. 2(b) can be constructed for any LFSR with $g(D)$ specified. If there are some groupings of nonzero terms in the feedback polynomial, then one may consider defining an FSM to capture these local constraints as was done in Fig. 2(d), for example.

TABLE I
EXAMPLES OF SPARSE GENERATING POLYNOMIALS FOR m-SEQUENCES [15]

Degree	Octal representation of generating polynomial
15	$[100003]_8$,$[140001]_8$,$[100021]_8$,$[104001]_8$
18	$[1000201]_8$,$[1004001]_8$,$[1000077]_8$,$[1760001]_8$
29	$[4000000005]_8$,$[5000000001]_8$
31	$[20{,}000{,}000{,}011]_8$,$[22{,}000{,}000{,}001]_8$

Consider the generating polynomial $g_{34}(D) = 1 + D^{19} + D^{20} + D^{33} + D^{34}$, so that $x_k = x_{k-19} \oplus x_{k-20} \oplus x_{k-33} \oplus x_{k-34}$. The cyclic graph with no hidden variables, corresponding to Fig. 2(b), is shown in Fig. 3(a). Another model is shown in Fig. 3(b) that uses hidden variables $\sigma_k^a = x_{k-20}, \sigma_k^b = x_{k-34}$, and $\sigma_k^c = x_{k-33} \oplus x_{k-34}$. This graph has two acyclic subgraphs that correspond to two-state FSMs with states given by σ_k^a and σ_k^b, respectively. Therefore, this may be viewed as decomposing a 2^{34}-state FSM into two coupled two-state FSMs. One iteration of the corresponding iMPA on this graph corresponds running the FBA on the two FSMs with activation of all variable nodes and hidden variable nodes between before each FBA is run.

B. Relation to Low-Density Parity-Check (LDPC) Codes and Further Reading

An LDPC code [16], [17] is a linear parity-check code with a parity-check matrix that has a small number of ones. Specifically, every valid codeword $\mathbf{c}$ satisfies $\mathbf{Hc} = \mathbf{0}$ where $\mathbf{c}$ is an $(n \times 1)$ binary vector and $\mathbf{H}$ is an $(n - k) \times n$ binary matrix, where we adopt the conventional notation of k input bits mapping to n coded bits via $(n - k)$ parity-check equations [18]. The standard graphical model for this is similar to that shown in Fig. 2(b), where there are n variable nodes representing the coded bits and the check capture the $(n - k)$ even-parity constraints. The iMPA algorithm described in the context of Fig. 2(b) is the same as the standard iterative decoder for LDPC codes. In fact, the structure imposed by the LFSR in (2) can be written as $\mathbf{H}_{\mathrm{LFSR}}\mathbf{x} = \mathbf{0}$, where $\mathbf{H}_{\mathrm{LFSR}}$ is a $((M - r) \times M)$ binary matrix and $\mathbf{x}$ is the vector of x_k values. Viewing m-sequences as a form of error correction code is not new; the corresponding codes are known as maximum length codes [18] and are of rate r/N in our notation. Since we consider only $M \ll N$ channel observations, our approach can be considered iterative decoding of punctured maximum length codes. Thus, the sparse property of the generating polynomial is akin to the low-density property of the LDPC $\mathbf{H}$ matrix. This interpretation does not imply that the m-sequence defines a code as powerful as an LDPC code because the structure of the ones in the $\mathbf{H}_{\mathrm{LFSR}}$ implies a relatively localized set of variable constraints and a very regular cycle structure. Both of

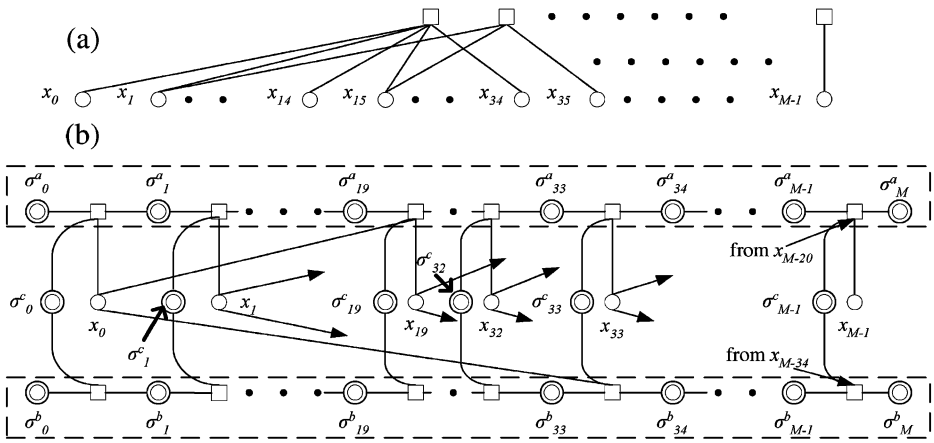

Fig. 3. Two graphical models for the 34-stage LFSR with $g_{34}(D) = 1 + D^{19} + D^{20} + D^{33} + D^{34}$.

these properties are avoided in the construction of good LDPC code parity-check matrices.

Adding hidden variables takes one away from the direct correspondence with LDPC codes, although the hidden binary variables can be viewed as coded bits that have been punctured from a larger LDPC code. However, the graph in Fig. 3(b) is very similar to that of a parallel concatenated convolutional code or "turbo" code [9]. In fact, it is well known that iterative decoding of turbo codes, LDPC codes and other turbo-like codes can be viewed as applying the same iMPA paradigm described above. The contribution of this work is demonstrate that this same conceptual approach can be applied to the problem of PN acquisition and this has powerful practical consequences.

There are a number of conventions for graphical modeling and describing the resulting iMPAs. Our convention most closely follows that of factor graphs [7], which generalizes the earlier work of Wiberg [8]. Wiberg generalized the work of Tanner [19], who developed graphical models without hidden variables for linear block codes analogous to that shown in Fig. 2(b), to include hidden variables. Wiberg also noted the impact of cycles and made the connection between iterative decoding and previously known optimal algorithms such as the FBA. Other conventions use configuration variable nodes in place of check nodes [6], [20] and make connection to known approaches in computer science [21]. In some of these conventions, directed graphs are used for modeling [12], [20], but it is undirected cycles that affect the optimality of the resulting iMPA because messages propagate in all directions. Finally, belief propagation, the sum-product algorithm, the turbo-principle, and other terms are used synonymously with iterative message passing.

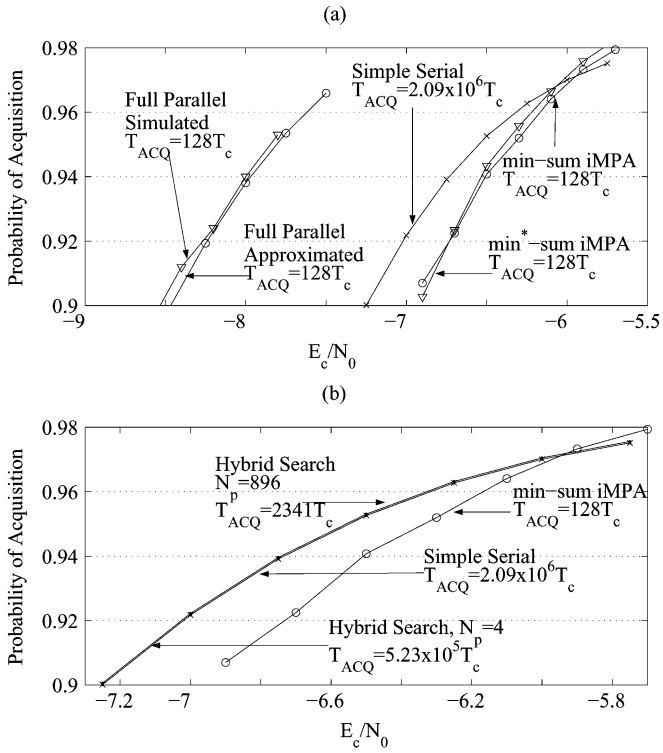

Fig. 4. Comparison of acquisition performance of various approaches for the UWB system with perfect frame synchronization and m-sequence generated by $g(D) = 1 + D + D^{15}$. All iMPA simulations are based on 100 iterations. Simple serial and hybrid searches use $M = 128$ chip times per dwell, while the iMPA and full parallel approaches use $M = 128$ total observations. (a) Compares the iMPA against the traditional simple serial and full parallel approaches. (b) Compares against hybrid search. (a) iMPA versus full parallel and simple serial, $M = 128, T_d = 128\,T_c, \theta_c = 0$, (b) iMPA versus hybrid search, $M = 128, T_d = 128\,T_c, \theta_c = 0$.

V. Simulation Results

A. Simulation Results for M-Sequence $[100\,003]_8$

We first consider simulation of the UWB system with perfect frame synchronization for the m-sequence generated by an 15-stage LFSR with $g(D) = 1 + D + D^{15}$. Unless otherwise specified, the performance of the traditional PN acquisition schemes is computed using the approximations stated in Section III. The threshold for both serial and hybrid searches is determined using $P_{\rm FA}$ of 10^{-6}. Algorithms are evaluated using $P_{\rm ACQ}$ versus E_c/N_0, acquisition time, and complexity.

1) UWB Systems With Perfect Knowledge of Frame Epoch: The performance of serial, full-parallel, hybrid, and the iMPA corresponding to Fig. 2(d) is shown in Fig. 4. The min-sum and min*-sum iMPAs have similar performance, each approximately 1.6 dB (in E_c/N_0) worse than that of the ML exhaustive search and 0.3 dB worse than that of the simple serial search. This quantifies the performance degradation due to cycles in the

TABLE II
COMPARISON OF ACQUISITION TIME (T_{ACQ}), MEMORY COMPLEXITY (R_m) AND COMPUTATIONAL COMPLEXITY (R_a) FOR THE m-SEQUENCE DEFINED BY $g(D) = 1 + D + D^{15}$. BOTH FULL PARALLEL SEARCH AND iMPA HAVE M OBSERVATIONS, $T_d = MT_c$ FOR SIMPLE SERIAL SEARCH AND HYBRID SEARCHES, AND THE iMPA RUNS 100 ITERATIONS. THE iMPA IS BASED ON THE GRAPH OF FIG. 2(d)

	Parallel	Serial	iMPA	hybrid $C_p = 4$	hybrid $C_p = 896$
T_{ACQ}	MT_c	$2^{r-1}MT_c$	MT_c	$2^{r-1}MT_c/C_p$	$2^{r-1}MT_c/C_p$
	$(128T_c)$	$(2.09 \cdot 10^6 T_c)$	$(128T_c)$	$(5.23 \cdot 10^5)$	$(2341T_c)$
R_m	$2^r(32736)$	1	$7M(896)$	$C_p(4)$	$C_p(896)$
R_a	$2^r M$	$2^{r-1}M$	$1700M$	$2^{r-1}M$	$2^{r-1}M$
	$(4.19 \cdot 10^6)$	$(2.09 \cdot 10^6)$	$(2.18 \cdot 10^5)$	$(2.09 \cdot 10^6)$	$(2.09 \cdot 10^6)$
$R_a \cdot T_c/T_{ACQ}$	2^r	1	1700	C_p	C_p
	(32767)			(4)	(896)

model of Fig. 2(d) and also suggests that min-sum processing is preferred in practice for this application since it is less complex and more robust to imperfect gain control [12]. The performance gain of C_p-correlator hybrid search, relative to simple serial search, even for large C_p, is insignificant. Though not explicitly presented here, simulations also show that the iMPA over Fig. 2(d) is about 0.5 dB better than the iMPA over Fig. 2(b) (i.e., the Tanner Graph).

The acquisition times of these algorithms are also given in Fig. 4. Both full parallel search and iterative MPAs achieve code acquisition in $128T_c$, where T_c is the chip interval [the frame time for the UWB system in Fig. 1(a)]. In contrast, the mean acquisition times of simple serial search and 4-correlator and 896-correlator hybrid search are $2.09 \cdot 10^6 T_c, 5.23 \cdot 10^5 T_c$, and $2341T_c$, respectively. Thus, the iMPAs are 16 000 times faster than the simple serial search and 18 times faster than 896-correlator hybrid search. These are conservative estimates since the penalty time for false acquisition in the serial and hybrid case has been assumed to be zero.

The complexity of these algorithms, measured both in terms of memory requirements (R_m) and the total number of arithmetic operations (R_a), is summarized in Table II. Values in parenthesis correspond to numerical results obtained using $M = 128$. Full parallel search requires a memory 36 times more than the iMPA, and the iMPA requires a memory 896 times more than the simple serial search but the same as the $C_p = 896$ hybrid search. In terms of computational complexity, the full parallel requires about 20 times the number of computations required for the iMPA, and simple serial and the two hybrid strategies each requires about 10 times the number of computations required for the iMPA. Thus, the iMPA provides a relatively low complexity approach to search all PN code alignments in parallel with reasonable performance.

Since all of the computations must be performed during the acquisition time, another measure of interest is this complexity normalized by the mean acquisition time T_{ACQ}. This is also shown in Table II, where the $R_a \cdot T_c/T_{\mathrm{ACQ}}$ of full parallel search is about 20 times that of the proposed iMPA. The proposed iMPA is 1700 times as complex as simple serial search but only 2 times as complex as the $C_p = 896$ hybrid search when measured by this metric.

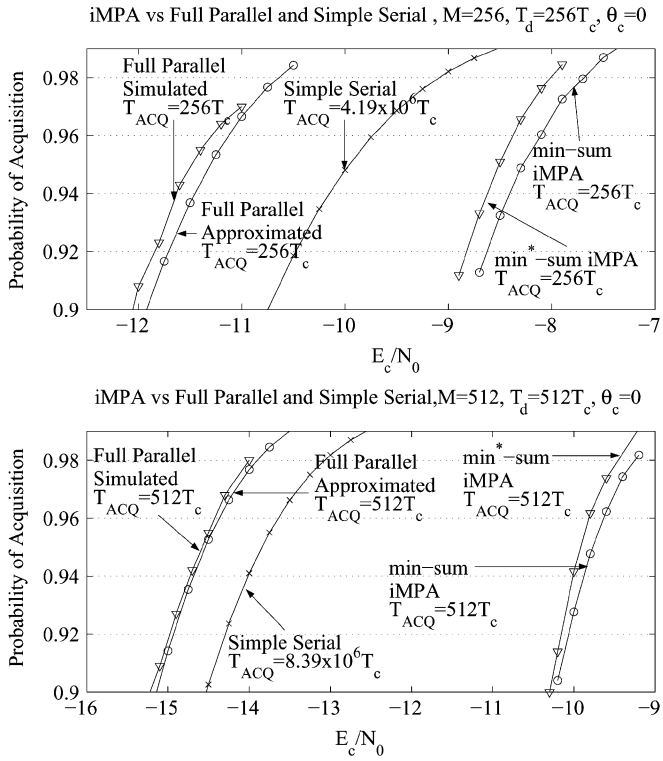

Fig. 5. Effects of increasing the observation window for various approaches for the UWB system with perfect frame synchronization and m-sequence generated by $g(D) = 1 + D + D^{15}$. All iMPA simulations are based on 100 iterations. Simple serial search use M chip times per dwell, while the iMPA and full parallel approaches use M total observations. (a) Shows $M = 256$ and (b) shows $M = 512$.

As illustrated in Fig. 5, doubling the length of the observation window provides approximately 3 dB of E_c/N_0 improvement for both the serial and full parallel search. This is expected since doubling the number of observations roughly doubles the ratio between the partial-period correlation [13] under the correct (in-phase) and out-of-phase alignments. On the other hand, the performance of the iterative MPA does not improve much when the observation length increases. This is shown in Fig. 6, where the minimum value of E_c/N_0 required to achieve $P_{\mathrm{ACQ}} = 0.9, (E_c/N_0)_{\mathrm{req}}$, is plotted against M for the various approaches. The degradation in $(E_c/N_0)_{\mathrm{req}}$ for the iMPA

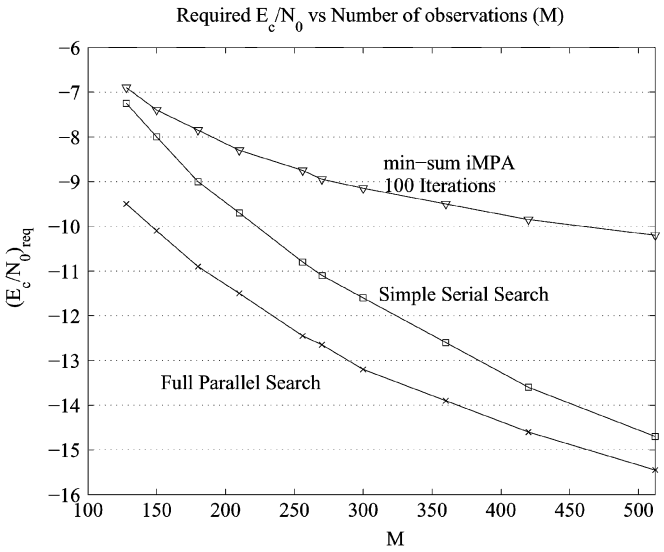

Fig. 6. Summary of the performance gain obtained with larger observation windows for the iMPA, serial search, and full parallel search for the UWB system with perfect frame synchronization and m-sequence generated by $g(D) = 1 + D + D^{15}$. Traits are summarized using $(E_c/N_0)_{\rm req}$ versus M, where $(E_c/N_0)_{\rm req}$ is the lowest SNR for which $P_{\rm ACQ} = 0.9$ can be achieved. Both full parallel search and min-sum iMPA have M observations and $T_d = MT_c$ in simple serial search.

relative to full parallel search is less than 2 dB when $M = 128$, but is more than 5 dB when $M = 512$. It is reasonable to conclude that this property of the iMPA is due to the regular cycle structure of the graph in Fig. 2(d)—i.e., each variable is involved in a cycle with minimum length 30.

Finally, as demonstrated in Figs. 4 and 5, the approximate analysis in Section III matches the simulated performance for full parallel search reasonably well. In the subsequent results, only the approximation analysis from Section III is presented.

2) Traditional DS/SS Systems With No Carrier Phase Knowledge: As described in Section III, traditional approaches use envelope detectors after I/Q PN code correlators to provide a test statistic when the carrier phase is unknown. This approach is not applicable to the iMPA because the iMPA does not directly compute correlations against the PN code, but rather over sequences that capture some substructure [e.g., the two-state FSM structure in Fig. 2(d)].

In order to apply the iMPA approach for the noncoherent DS/SS case, we use a method based on generalized likelihood [12]. Specifically, a finite number of candidate θ_c values are considered. For example, suppose four candidate phase values were considered: $\tilde{\theta}_c \in \{0, \pi/2, \pi, 3\pi/2\}$. Then, four versions of the iMPA can be run, each using $\mathrm{M}_{\rm ch}[x_k] = p(ze^{-j\tilde{\theta}_c} \mid x_k)$ for the specific value of $\tilde{\theta}_c$. The final decision for the PN alignment is taken from the iMPA with the best soft-decision information (i.e., largest difference between best decision and second best decision).

Simulation results are shown in Fig. 7 along with the curves of the ideal case where θ_c is known. The eight candidate phase approach works well, at the cost of an increase in complexity by a factor of 8, whereas an additional 2 dB degradation is observed when four candidate phase values are used.

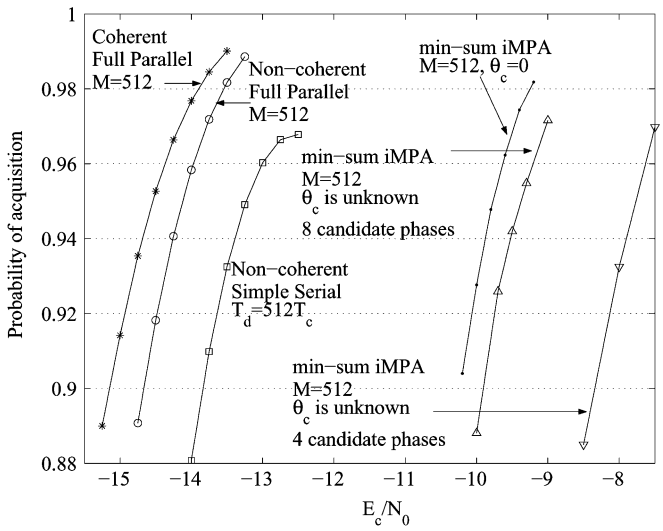

Fig. 7. Performance of iterative MPA in traditional DS/SS system with unknown carrier phase and m-sequence generated by $g(D) = 1 + D + D^{15}$. The block size M of both full parallel search and 100-iteration min-sum iMPA over Fig. 2(d) is 512, dwell time T_d for simple serial search is $512T_c$.

This approach can also be viewed as a simple form of joint phase estimation and PN acquisition, where the phase estimator is based on a simple quantized approximation. Other approaches for joint parameter estimation and iterative message passing [12, Ch. 4] can also be applied and other unknown parameters (e.g., a frequency offset) could be included as well using similar techniques.

B. Simulation Results for Other m-Sequences

In Section V-A, the iMPA based on the graphical model in Fig. 2(d) was investigated for one specific m-sequence with $g(D) = 1 + D + D^{15}$, where three nonzero g_i's appear at the two ends. Noting that since binary primitive polynomials have at least three nonzero g_i's and the shortest cycle in the graphs representing an r-stage LFSR has length at most $2r$, this $g(D)$ is the "most favorable" m-sequence with $r = 15$ for the proposed iMPA acquisition algorithm.

In this section we evaluate our approach for different graphical models and for different generating polynomials using the UWB system model. The generators considered are $g_{22}(D) = 1 + D + D^{22}([20\,000\,003]_8), g_{18}(D) = 1 + D^{11} + D^{18}([1\,004\,001]_8)$, $g_{15}(D) = 1 + D^5 + D^6 + D^8 + D^{10} + D^{12} + D^{15}([112\,541]_8)$, and $g_{34}(D) = 1 + D^{19} + D^{20} + D^{33} + D^{34}([300\,006\,000\,001]_8)$. The generated m-sequences are denoted as $\underline{x}_{22}, \underline{x}_{18}, \underline{x}_{15}$, and $\underline{x}_{34}$, respectively, and the corresponding 100 iteration min-sum iterative MPAs are denoted as $\mathrm{iMPA}_{22}, \mathrm{iMPA}_{18}, \mathrm{iMPA}_{15}$, and iMPA_{34}, respectively. More specifically, these are based on the following graphical models: iMPA_{22} is based on a graph similar to that in Fig. 2(d) with $\sigma_k = x_{k-1}, \mathrm{iMPA}_{18}$, and $0\mathrm{iMPA}_{15}$ are based on (Tanner) graphs similar to that in Fig. 2(b), and iMPA_{34} is based on the graph in Fig. 3(b). For comparison purposes, the m-sequence used in Section V-A is denoted as $\underline{x}_0$ and the corresponding iterative MPA is denoted as iMPA_0.

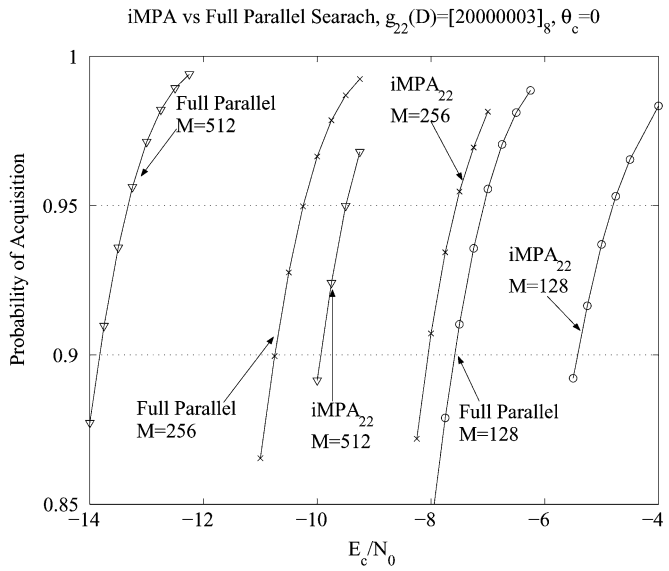

Fig. 8. Performance of iMPA_{22}: 100 iteration min-sum, $g_{22}(D) = D^{22} + D + 1, N = 2^{22} - 1 = 4\,194\,303$ for the UWB system with perfect frame synchronization.

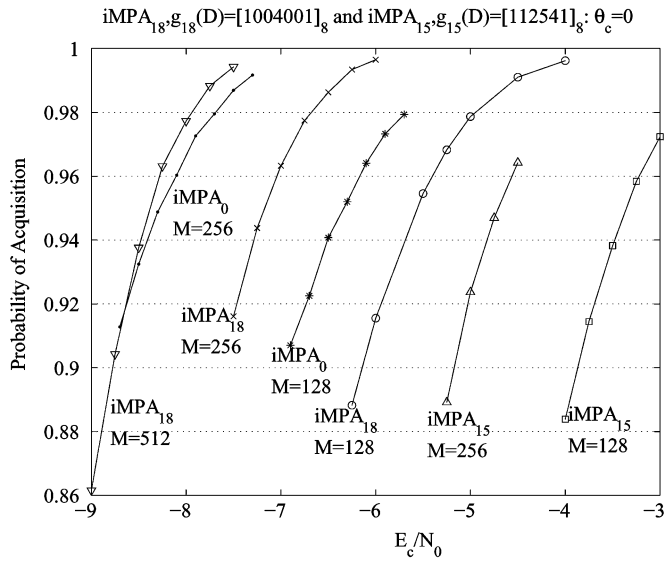

Fig. 9. Performance of iMPA_{18} and iMPA_{15}: 100 iteration min-sum processing over Tanner graphs. For the UWB system with perfect frame synchronization.

Fig. 8 contains simulation results for iMPA_{22}. Since $g_{22}(D)$ has three nonzero coefficients appearing at the two ends, it is another "most favorable" m-sequence with longer period $N = 2^{22} - 1 = 4\,194\,303$. Comparing with curves in Figs. 4 and 5, we observe that the iMPA_0 performs 1.5 dB better than iMPA_{22} when M is 128. A likely explanation for this effect is that the length-128 out-of-phase partial-period correlation [13] of $\underline{x}_{22}$ is much larger than that of $\underline{x}_0$. However, when M is doubled, iMPA_{22} gains more than iMPA_0 does, and when $M = 512$, they have nearly the same performance. This effect is most likely due to the fact that the underlying graph of iMPA_{22} has shortest cycles of the length 44, whereas iMPA_0 is running on graph with shortest cycles of length 30. This is evidence that the property of diminishing benefits of increasing

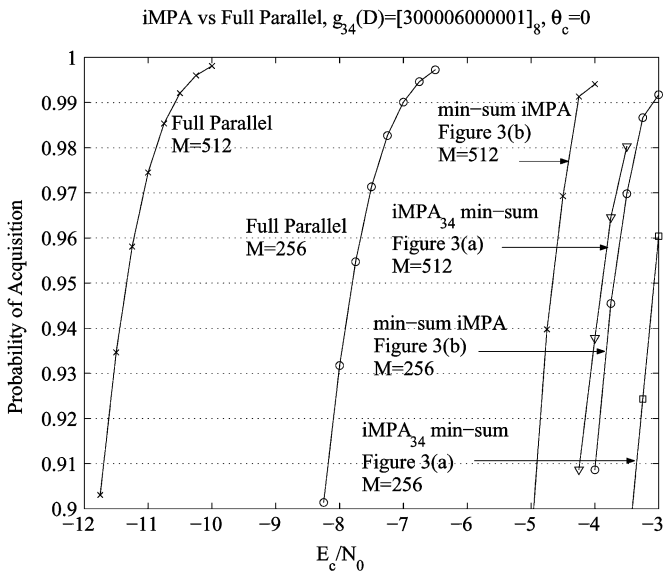

Fig. 10. Performance of iMPA for 34-stage LFSR with $g_{34}(D) = 1 + D^{19} + D^{20} + D^{33} + D^{34}$: 100 iteration min-sum over Fig. 3(a) and (b). For the UWB system with perfect frame synchronization.

the observation interval is due in part to the length of the shortest cycle in the graph.

Fig. 9 contains simulation results for iMPA_{18} and iMPA_{15}. Although iMPA_{18} provides performance gain when M is doubled, it does not perform as well as iMPA_0 (the iMPA_{18} with $M = 512$ has nearly the same performance as iMPA_0 with $M = 256$). The length of cycles, six in this case, is a likely explanation for this effect. On the other hand, the iMPA_{15} performs poorly: for $M = 128$, the iMPA_0 is 3 dB better than the iMPA_{15}; and when M is doubled, the iMPA_{15} has less than 1 dB performance gain.

Simulation results of iMPA_{34} are plotted in Fig. 10. This includes results for both iMPA_{34}, based on the graph in Fig. 3(b), and the iMPA based on the graph of Fig. 3(a). The former performs approximately 0.5 dB better than the latter, but both perform poorly relative to that of full parallel search.

Summarizing the results of the iMPA simulations, we conclude that good performance is possible for relatively small observation windows, but the performance does not improve with increasing M as quickly as that of traditional approaches. The likely cause for this is the regular cycle structure in the underlying graphical models. One possible way to alleviate the effects of cycles is to damp the messages to avoid convergence to a poor solution. In [1], the method of filtering messages to damp out rapid fluctuations was applied to the problem at hand. This yielded a performance gain of approximately 0.5 dB at the cost of significantly more memory complexity. In the following section, we suggest an approach that achieves a similar performance enhancement with less complexity than the baseline iMPA.

C. Verification Scheme

A verification scheme is required if there is the possibility that no signal is present. For example, in the UWB system in Fig. 1(a), if the hypothesized frame epoch is incorrect, there is

no signal present during observation times, so the *null-hypothesis* should be considered. In this section, we suggest a verification scheme and also use this verification scheme to better capitalize on additional observations by using the iMPA over multiple time windows. The following development assumes the UWB model but can be directly generalized to the noncoherent DS/SS case.

The iMPA can be viewed as a method for generating likely initial states, for each of which a traditional correlation threshold test could be performed. The proposed heuristic for postprocessing the iMPA decisions is based on this observation. Specifically, the baseline iMPA using I iterations is run up to V times, each time with a slightly perturbed set of channel observations. After each of these runs, a state estimate $\hat{\mathbf{u}}$ is obtained and the correlation statistic $v(\hat{\mathbf{u}}) = \Re\{r(\hat{\mathbf{u}})\}$, where $r(\mathbf{u})$ is defined in (5), is computed. If $v(\hat{\mathbf{u}}) > \eta$, acquisition is declared, otherwise, the observation set is perturbed and the process is repeated. Assuming that $P_{\rm ACQ}$ is required to be at least 0.9, which is commonly used in the code acquisition literature [2], the threshold can be selected as

$$P_r\{v(\hat{\mathbf{u}}) > \eta\} = Q\left(\frac{\eta - \sqrt{E_c}}{\sqrt{N_0/2M}}\right) \geq 0.9 \longrightarrow \eta = Q^{-1}(0.9)\cdot\sqrt{N_0/2M} + \sqrt{E_c}. \quad (20)$$

The way in which the observation set is perturbed is that the signs of the S least reliable observations are flipped. More precisely, since the sign of $\Re\{z_k\}$ provides a decision on x_k without regard to the PN code structure, $|\Re\{z_k\}|$ is a measure of the quality or reliability of this chip-level observation [i.e., a large positive (negative) value corresponds to high confidence that $x_k = 0 (x_k = 1)$] So, after each run of the iMPA, the signs of the S least reliable observations are flipped. Note that after each time the iMPA is run, the signs of the observations already flipped remain flipped and another S are selected to be altered. Simulation results not presented here indicate that this modification provides an improvement of approximately 0.5 dB, relative to the iMPA_0, with the total number of iterations decreased by roughly 30%.

To further improve performance, multiple time windows of size M can be combined together. Specifically, given M_2 nonoverlapping windows of size M observations each, the above modified iMPA can be used to obtain an initial state estimate and a correlation statistic for each. The state estimate with largest correlation is then selected as the final decision. Clearly, the larger the M_2 is, the better the algorithm performs. However, the larger the M_2, the longer the acquisition time and since rapid acquisition is desired, a small M_2 is preferred. This defines a modified iMPA, which we denote by $\mathrm{iMPA}^{(v)}(I, V, S, M_2)$, where M_2 is the number of nonoverlapping observation sets of size M. The parameters V and S set the maximum number of times the baseline, I iteration, iMPA is run per observation set and the number of signs flipped between these runs, respectively.

Simulation results for $\mathrm{iMPA}^{(v)}(I = 25, V = 8, S = 20, M_2 = 4)$, and $M = 512$ are shown in Fig. 11. Compared with the iMPA using one window of $M = 512$ observations, this modified algorithm has a 3-dB performance gain. Also,

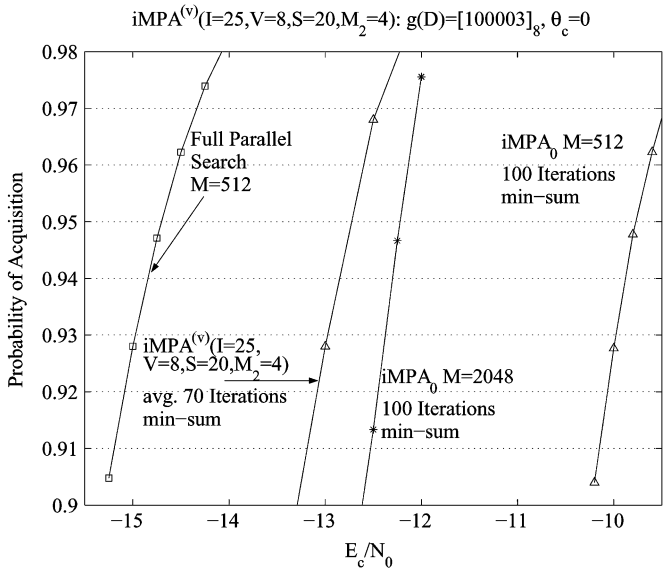

Fig. 11. Improvement obtained by verification scheme to combine multiple windows of observations together. For the UWB system with perfect frame synchronization and m-sequence generated by $g(D) = 1 + D + D^{15}$.

using $\mathrm{iMPA}^{(v)}$ to combine four windows of size 512 outperforms the baseline iMPA operating on 2048 observations with significantly less complexity.

Considering a practical scenario where the energy per bit to N_0 ratio required is 7 dB and the spreading ratio is $128 = 21$ dB, the PN code acquisition algorithm should work at $(E_c/N_0)_{\rm req} = -14$ dB. Results from Fig. 11 show that this can be achieved with an acquisition time of 2048 chip times using $\mathrm{iMPA}^{(v)}(I = 25, V = 8, S = 20, M_2 = 4)$. Referring to Fig. 5, simple serial search works at $(E_c/N_0)_{\rm req}$, but requires $8.39 \cdot 10^6$ observations on average, which is substantially slower than the proposed approach.

1) Joint PN and Frame Epoch Acquisition for the UWB System: As a final example we return to the UWB system in Fig. 1(a) when neither the PN alignment nor the frame epoch is known at the receiver. A PN acquisition algorithm should be able to detect the null-hypothesis rapidly so that a hypothesized frame epoch can be discarded and another is investigated. This cannot be achieved by either serial search or hybrid search because the whole uncertainty region must be searched before a null declaration can be made. On the other hand, the iMPA not only achieves rapid code acquisition when the signal is present, but also can determine null-hypothesis quickly. This is further enhanced by "early stopping," i.e., it is not necessary to run all the iterations to recognize a null-hypothesis. To do this, another threshold $\eta_{\rm ES} < \eta$ is needed.

The frame epoch is estimated in a serial manner (i.e., starting with $\tilde{\xi} = 0$, then $\tilde{\xi} = 1$, then $\tilde{\xi} = 2$, etc.) until the correct frame epoch is detected. For a given hypothesized frame epoch, referring to the $\mathrm{iMPA}^{(v)}$ in Section V-C, if the best initial state estimate obtained has $v(\hat{\mathbf{u}}) < \eta_{\rm ES}$, the null-hypothesis is declared. Then, a new hypothesized frame epoch is considered and the $\mathrm{iMPA}^{(v)}$ restarts with a new set of observations based on this hypothesized frame epoch.

Both the $C_p = 896$ hybrid search and $\mathrm{iMPA}^{(v)}(I = 25, V = 8, S = 5, M_2 = 1)$ are examined for the UWB system since

they have similar memory requirements. Referring to (1), the m-sequence is generated by $g(D) = 1+D+D^{15}$ and the frame epoch is estimated in a serial manner. Results are summarized in Table III, where the number of observations of the iMPA$^{(v)}$ is $M = 128$ and the dwell time for $C_p = 896$ hybrid search is $T_d = 128T_f = 128T_c$. Furthermore, there are 1000 possible bins to be searched for the frame epoch ξ in each frame (i.e., $T_f/T_p = 1000$). The modified iMPA compares very favorably to hybrid search both in terms of complexity and acquisition time. Specifically, the proposed iMPA is about 18 times faster and 46 times less complex than the $C_p = 896$ hybrid search. Thus, the proposed iMPA-based acquisition algorithm is even more favorable relative to traditional hybrid/serial search strategies for low duty-cycle UWB systems where joint frame/PN synchronization is required.

TABLE III
T_{ACQ} AND R_a OF $C_p = 896$ HYBRID SEARCH AND THE PROPOSED iMPA$^{(v)}$: JOINT FRAME/PN SYNCHRONIZATION IN THE UWB EXAMPLE CONSIDERED IN SECTION V-C1

	T_{ACQ}	R_a
$C_p = 896$ hybrid search	$1.17 \cdot 10^6 T_f$	$2.10 \cdot 10^9$
iMPA$^{(v)}$	$6.4 \cdot 10^4 T_f$	$4.5 \cdot 10^7$

VI. Conclusion and Future Work

Iterative techniques are well known to be applicable in a wide range of applications, and in this paper we applied this principle to address the PN code acquisition problem. Simulation results showed that the iterative message passing algorithms based on sparse cyclic graphical models worked well. Specifically, it is the first method that can search all possible PN phases in parallel with complexity significantly lower than optimal full parallel search and good low SNR performance. This approach is especially favorable when the block size is relatively small.

One undesirable characteristic of the iMPA approach is that the availability of larger observation sets does not improve performance as much as in traditional approaches. This is apparently due to the regular, short cycle structure in the underlying graphical model which causes the algorithm to converge based predominately on an initial portion of the observation window. We addressed this shortcoming by considering verification postprocessing that allows the results of the iMPA operating on subwindows to be combined. This same verification processing also enabled us to detect the absence of signal quickly, thus making this approach even more attractive for low duty cycle UWB waveforms.

A message passing PN search algorithm with low complexity may also find other applications in noncooperative military communication links. For example, the ability to acquire a long PN code with a short observation interval would enable one to acquire a spread-spectrum signal with data modulation present. Evaluating the iMPA acquisition algorithm when multipath is present, joint channel estimation/PN synchronization, and hardware architectures are interesting topics for future research.

Finally, it is interesting to consider the design of pseudorandom sequences that are inherently generated by more random-like sparse loopy graphical models. In this paper, we considered existing m-sequences and suggested simple graphical models that are not ideal for application of the iMPA heuristic due to the regular structure of short cycles. Also, the complexity of the local constraints used is low (e.g., two-state FSMs), thus making the effects of this cycle structure likely more detrimental and slowing convergence. It may be useful to consider LFSR sequences that do not achieve maximum period, but have generating polynomials with more consecutive ones that can be grouped into FSM subgraphs with stronger local structure. Finally, investigating systematic methods for extracting effective cyclic graphical models for arbitrary systems is a challenging and interesting direction for further research and significant progress in this direction would directly apply to the PN acquisition problem considered.

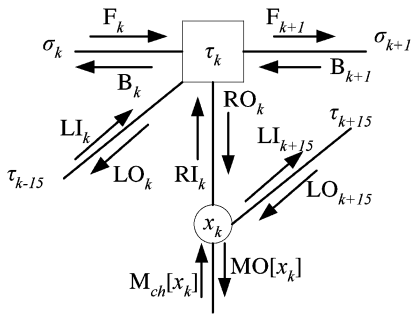

Fig. 12. Detailed notation of the input and output messages associated with one check node in Fig. 2(d).

Appendix

In this appendix, we clarify the messages passed along edges in Fig. 2(d). A check node from Fig. 2(d) is redrawn in Fig. 12. The configurations of this check node are indexed by the value of the transition variable $\tau_k = (\sigma_k, \sigma_{k+1})$. Also shown in Fig. 12, specific labels are given to messages passed along these edges. The chip-level soft-decision channel information is

$$\mathrm{M_{ch}}[x_k] = \frac{2\sqrt{E_c} z_k (-1)^{x_k}}{N_0} \quad x_k = 0, 1 \tag{21}$$

where z_k is the relevant real part of the observation in (3) and the local configuration metric is

$$\mathrm{M}[\tau_k] = \mathrm{F}_k[\sigma_k] + \mathrm{RI}_k[x_k] + \mathrm{LI}_k[x_{k-15}] + \mathrm{B}_{k+1}[\sigma_{k+1}]. \tag{22}$$

Note that the values of the variables σ_k, x_k, x_{k-15}, and σ_{k+1} are determined when a conditional value of τ_k is set and the dependency of these variables on τ_k is not explicitly shown in this Appendix. Excluding the $\mathrm{F}_k[\cdot]$ and $\mathrm{B}_{k+1}[\cdot]$ from $\mathrm{M}[\tau_k]$, the remaining sum may be viewed as a generalized state transition metric used during the FBA stage of the iMPA.

With (22), a compact way of expressing the message updating in min-sum form is[7]

$$\mathrm{F}_{k+1}[\sigma_{k+1}] = \min_{\tau_k : \sigma_{k+1}} \mathrm{M}[\tau_k] - \mathrm{B}_{k+1}[\sigma_{k+1}] \quad \sigma_{k+1} = 0, 1 \tag{23}$$

[7]Since the term subtracted in each of (23)–(29) is constant over all terms in the minimization, each equation can be written in a form where (22) is simplified by cancelling that term *priori* to minimization.

$$\mathrm{B}_k[\sigma_k] = \min_{\tau_k:\sigma_k} \mathrm{M}[\tau_k] - \mathrm{F}_k[\sigma_k] \quad \sigma_k = 0,1 \tag{24}$$

$$\mathrm{LO}_k[x_{k-15}] = \min_{\tau_k:x_{k-15}} \mathrm{M}[\tau_k] - \mathrm{LI}_k[x_{k-15}] \quad x_{k-15} = 0,1 \tag{25}$$

$$\mathrm{RO}_k[x_k] = \min_{\tau_k:x_k} \mathrm{M}[\tau_k] - \mathrm{RI}_k[x_k] \quad x_k = 0,1 \tag{26}$$

$$\mathrm{MO}[x_k] = \mathrm{LO}_{k+15}[x_k] + \mathrm{RO}_k[x_k] \quad x_k = 0,1 \tag{27}$$

$$\mathrm{LI}_k[x_{k-15}] = \mathrm{RO}_{k-15}[x_{k-15}] + \mathrm{M}_{\mathrm{ch}}[x_{k-15}] \quad x_{k-15} = 0,1 \tag{28}$$

$$\mathrm{RI}_k[x_k] = \mathrm{LO}_{k+15}[x_k] + \mathrm{M}_{\mathrm{ch}}[x_k] \quad x_k = 0,1. \tag{29}$$

Similarly, min*-sum messages can be obtained by replacing min operators in (22)–(26) by $\min^*$.

Pseudocode of the Proposed iMPA Algorithm on Fig. 2(d)

Step 1) Initialization: $\mathrm{F}_0[\sigma_0], \mathrm{B}_M[\sigma_M], i \leftarrow 0, I \leftarrow$ maximum number of iterations

$$\mathrm{M}_{\mathrm{ch}}[x_k] \leftarrow \frac{2\sqrt{E_c} z_k (-1)^{x_k}}{N_0}; \quad \mathrm{LI}_k[x_{k-15}] \leftarrow \mathrm{M}_{\mathrm{ch}}[x_{k-15}]; \quad \mathrm{RI}_k[x_k] \leftarrow \mathrm{M}_{\mathrm{ch}}[x_k].$$

Step 2) Forward-backward algorithm: Updating $\mathrm{F}_k[\sigma_k]$ and $\mathrm{B}_k[\sigma_k], 0 \le k \le M-1$, sequentially using (23) and (24), respectively. $\mathrm{F}_0[\sigma_0] \rightarrow \mathrm{F}_1[\sigma_1] \rightarrow \cdots \mathrm{F}_k[\sigma_k] \cdots \rightarrow \mathrm{F}_M[\sigma_M]; \mathrm{B}_M[\sigma_M] \rightarrow \cdots \mathrm{B}_{k+1}[\sigma_{k+1}] \rightarrow \mathrm{B}_k[\sigma_k] \cdots \rightarrow \mathrm{B}_0[\sigma_0]$.

Step 3) Update $\mathrm{LO}_k[x_{k-15}]$ and $\mathrm{RO}_k[x_k] : 0 \le k \le M-1$, using (25) and (26), respectively. Then, $i \leftarrow i+1, \mathrm{LI}_k[x_{k-15}]$, and $\mathrm{RI}_k[x_k]$ are updated using (28) and (29), respectively.

Step 4) Selecting candidate decisions: $\lfloor M/15 \rfloor$ nonoverlap (intermediate) estimates of the initial state are obtained using $\mathrm{M}_k[x_k] = \mathrm{M}_{\mathrm{ch}}[x_k] + \mathrm{MO}[x_k], 15i \le k \le 15i+14, i = 0, 1, \ldots \lfloor M/15 \rfloor$. The decision rule is $\hat{x}_k = 0$ when $\mathrm{M}_k[x_k = 0] < \mathrm{M}_k[x_k = 1]$, and $\hat{x}_k = 1$, otherwise.

Step 5) If $i < I$, go to Step 2); otherwise, the estimate that appears the most times in Step 4) is selected to be final estimate of the initial state.

References

[1] M. Zhu and K. M. Chugg, "Iterative message passing techniques for rapid code acquisition," in *Proc. IEEE Military Commun. Conf.*, 2003.

[2] M. K. Simon, J. K. Omura, R. A. Scholtz, and B. K. Levitt, *Spread Spectrum Communications Handbook*. New York: McGraw-Hill, 1994.

[3] A. Polydoros and C. L. Weber, "A unified approach to serial search spread-spectrum code acquisition," *IEEE Trans. Commun.*, vol. 32, no. 5, pp. 542–560, May 1984.

[4] R. B. Ward, "Acquisition of pseudonoise signals by sequential estimation," *IEEE Trans. Commun.*, vol. COMM-13, no. 4, pp. 475–483, Dec. 1965.

[5] S. M. Aji, "Graphical Models and Iterative Decoding," Ph.D. dissertation, California Inst. Technol., Pasadena, CA, 1999.

[6] S. M. Aji and R. J. McEliece, "The generalized distributive law," *IEEE Trans. Inf. Theory*, vol. 46, pp. 325–343, Mar. 2000.

[7] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.

[8] N. Wiberg, "Codes and Decoding on General Graphs," Ph.D. dissertation, Linköping Univ., Linköping, Sweden, 1996.

[9] C. Berrou, A. Glavieux, and P. Thitmajshima, "Near shannon limit error-correcting coding and decoding: Turbo-codes," in *Proc. Int. Conf. Commun.*, Geneva, Switzerland, May 1993, pp. 1064–1070.

[10] I. D. O'Donnell, S. W. Chen, B. T. Wang, and R. W. Brodersen, "An integrated, low power, ultra-wideband transceiver architecture for low-rate, indoor wireless systems," in *Proc. IEEE CAS Workshop Wireless Commun. Netw.*, Sep. 2002.

[11] X. Chen, "Iterative Data Detection: Complexity Reduction and Applications," Ph.D. dissertation, Univ. Southern California, Los Angeles, CA, Dec. 1999.

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

[13] S. W. Golomb, *Shift Register Sequences, Revised Edition*. Laguna Hills, CA: Aegean Park, 1982.

[14] J. G. Proakis, *Digital Communications*, 3rd ed. New York: McGraw-Hill, 1995.

[15] R. L. Peterson, R. E. Ziemer, and D. E. Borth, *Introduction to Spread-Spectrum Communications*. Englewood Cliffs, NJ: Prentice-Hall, 1995.

[16] R. G. Gallager, *Low-Density Parity-Check Codes*. Cambridge, MA: MIT Press, 1963.

[17] D. J. C. MacKay and R. M. Neal, "Near Shannon limit performance of low density parity check codes," *Inst. Elect. Eng. Electron. Lett.*, vol. 32, pp. 1645–1646, Aug. 1996.

[18] S. Lin and J. D. Costello, *Error Control Coding: Fundamentals and Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1983.

[19] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inf. Theory*, vol. IT-27, no. 5, pp. 533–547, Sep. 1981.

[20] R. J. McEliece, D. J. C. MacKay, and J. F. Cheng, "Turbo decoding as an instance of Pearl's "belief propagation" algorithm," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 2, pp. 140–152, Feb. 1998.

[21] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, CA: Morgan Kaufmann, 1988.

Keith M. Chugg (S'88–M'95) received the B.S. degree (high distinction) in engineering from Harvey Mudd College, Claremont, CA, in 1989, and the M.S. and Ph.D. degrees in electrical engineering from the University of Southern California (USC), Los Angeles, in 1990 and 1995, respectively.

During the 1995–1996 academic year, he was an Assistant Professor with the Electrical and Computer Engineering Department, University of Arizona, Tucson, AZ. In 1996, he joined the Electrical Engineering Department, USC, where he is currently an Associate Professor. Along with his former Ph.D. students, A. Anastasopoulos and X. Chen, he is coauthor of the book *Iterative Detection: Adaptivity, Complexity Reduction, and Applications* (Norwell, MA: Kluwer). He is a cofounder of TrellisWare Technologies, Inc., where he is Chief Scientist. His research interests are in the general areas of signaling, detection, and estimation for digital communication and data storage systems. He is also interested in architectures for efficient implementation of the resulting algorithms.

Dr. Chugg has served as an Associate Editor for the IEEE Transactions on Communications and was Program Co-Chair for the Communication Theory Symposium at GLOBECOM 2002.

Mingrui Zhu received the B.S. degree in electronics engineering from Tsinghua University, Beijing, China. He is currently working towards the Ph.D. degree in electrical engineering at the University of Southern California (USC), Los Angeles.

His research interests are in the areas of iterative detection algorithms and wireless communication systems.

Mr. Zhu received the Fred W. Ellersick Award for the best unclassified paper at MILCOM 2003.