SYNCHRONIZATION IN LOW DUTY CYCLE UWB SYSTEMS USING ITERATIVE MESSAGE PASSING

by

Mingrui Zhu

A Dissertation Presented to the
FACULTY OF THE GRADUATE SCHOOL
UNIVERSITY OF SOUTHERN CALIFORNIA
In Partial Fulfillment of the
Requirements for the Degree
DOCTOR OF PHILOSOPHY
(ELECTRICAL ENGINEERING)

December 2005

UMI Number: 3219858

UMI Microform 3219858

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

Dedication

To my parents, Huaibei Zhu and Zhuxing Zhu, for their love, encouragement and wisdom.

Table of Contents

List of Tables

List of Figures

Abstract

Iterative message passing algorithms (MPAs) 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.

The goal of the first part of this work is to apply message passing techniques to the pseudo random code acquisition problems. To do that, good pseudo-noise patterns are represented by sparse graphical models, and the standard iterative MPAs are applied over the graphs to approximate maximum likelihood synchronization. Simulation results show that the iterative MPAs can achieve better performance than the traditional serial search in the sense that they work at low signal-to-noise ratios and are much faster. Compared to full parallel search, this approach typically provides significant complexity reduction.

However, the proposed algorithm does not perform well enough when the number of observations is relatively large, and we believe that the main reason for this is that the underlying graphical representation is not good enough. This motivates the second part of this work, specifically, understanding and analyzing the performance of iterative MPAs on bi-partite loopy graphs.

Using techniques of eigenvalue analysis, we derive bounds on the variable expansions of Tanner graphs. These bounds lead to bounds on stopping distance and stopping

redundancy, which are critical parameters of the Tanner graphs that can determine the performance of iterative decoding for binary erasure channels. Based on these results, we will also propose two criteria in an attempt to relate variable expansions of Tanner graphs with the performance of the associated iterative MPAs for binary symmetric channels and additive white Gaussian channels.

Chapter 1

Introduction

1.1 Motivation and Research Focus

Though the history of message passing techniques can be traced back to Gallager's work on Low-Density Parity-Check (LDPC) Codes [21, 22] forty years ago, they were not well recognized until the invention of Turbo Codes [10, 11]. From then on, message passing algorithms (MPAs) have found applications in a wide range of data detection problems because they can provide near optimal performance and significant complexity reduction. The main motivation of this work is to find new applications for the iterative message passing techniques.

Using various simulations, we will demonstrate that the iterative message passing techniques can be used to solve the pseudo random or pseudo noise (PN) code acquisition problem efficiently. However, to apply this techniques to practical communication systems, with strong noise and/or interference and without channel state information (CSI), the proposed algorithms need to be improved.

Determining the performance of iterative message passing algorithms on loopy graphs is still an open problem to the communication society. Therefore, it is difficult to analyze and improve our algorithm. Thus, the second part of this work tries to understand the behavior of iterative message passing algorithm by analyzing the expansion properties of the underlying graphs.

1.1.1 Rapid code acquisition for UWB

Since Ultra-Wideband Radio (UWB) systems [8, 48, 50, 72] utilize pseudo noise spread-spectrum modulation, they must be able to synchronize the received PN code with the receiver code by searching over the code phases, and in addition, over a uncertainty region of frame epochs. The traditional acquisition strategies may be insufficient for UWB systems because exhaustive parallel search is too complex, and simple serial search is too slow. Sometimes, hybrid search is implemented to trade complexity for acquisition speed. However, it can be shown that, under various situations where very long PN codes are deployed and/or the channel is time-varying, hybrid search can not provide satisfactory solutions as both implementable and fast enough to address the variation of the channel.

By exploring the structure of the PN code generators, we demonstrate that there are many PN patterns that can be represented by sparse loopy graphs. Thus, iterative message passing algorithms on sparse loopy graphs can be used to solve the rapid PN code acquisition problems [15, 76]. On additive white gaussian noise (AWGN) channels, the suggested algorithm works well. Simulation results show that it works at low signal-to-noise ratios (SNRs) and is much faster than the simple serial search. It is also shown

that the proposed algorithm can acquire the code phase within time interval comparable to exhaustive full parallel search while its complexity is significantly less. Furthermore, there exist verification schemes that can provide rough estimate of frame epoch.

It should be noted that the acquisition module of the UWB systems must work at extremely low SNR because of the lack of knowledge of the channel state information. Since the original iterative MPA does not provide a good approximation to the full parallel search within that SNR range, several heuristics are proposed to improve the performance of the proposed algorithm. However, the performance gains are usually not significant, and we believe that it is related to the "goodness" of the graphical representations. This is the main motivation of the second part of our work, which tries to provide something sensible that ties properties of the graphical representations with the performance of the associated iterative message passing algorithms.

1.1.2 Understanding iterative MPAs on loopy graphs

To understand the message passing algorithms and the nature of the solutions they find, researchers have focused on their algebraic structures at first. Perez, Seghers and Costello [41] used *distance spectrum* to explain why Turbo Codes have performance close to Shannon capacity at low SNRs but have a relative high error floor at high SNRs. Benedetto and Montorsi [6, 7] introduced the ideas of *uniform interleaver* and *effective free distance* to explain how iterative decoding benefits from *interleaving gain* at low SNRs.

More recent studies focused on the graphical representations of the codes, and message passing algorithms are defined on these graphs. This not only generalizes the idea of

turbo decoding, but also provides more insights into the decoding process of both Turbo Codes and LDPC codes. It is now well known that once the graph is given, the standard processing is well defined and only schedule needs to be specified. The standard message passing algorithms referred to in this dissertation are well defined in [2, 3, 14, 28, 29, 39, 40, 70], and iterative message passing algorithms or iMPAs refer to MPAs over loopy graphs. Some related works are summarized below.

- Tanner-Wiberg Graphs [59, 70]
- Belief Networks and Pearl's Belief Propagation Algorithm [29, 35, 39, 40]
- Factor Graphs [28, 29, 71]
- Expander graphs and expander codes [12, 55]
- Generalized Minimum Distance (GMD) decoding [1]
- Junction Trees, Junction Graphs and Generalized Distribution Law [2, 3]
- Generalized Belief Propagation (GBP) algorithms [4, 57, 58]
- Normal realizations of codes on graphs [19]
- Iterative decoding on Tanner Graphs for Binary Erasure Channel (BEC) [17, 31, 32, 37, 43, 51]
- Density evolution and Capacity-Achieving LDPC codes [46, 47]
- Pseudo-Codewords, Pseudo-Weight and their relations to the performance of iterative decoding [20, 65–67]

It has been proven that the standard message passing algorithms running on acyclic graphs, (*i.e.,*graphs without loops), provide globally optimal solutions [2, 3, 28]. By demonstrating the connections between MPAs and variational approaches to approximate free energy, recent results [4, 57, 58] also showed that the standard message passing algorithms can only converge to a stationary point of *approximate free energy*. When the underlying graph is acyclic, this approximate free energy is convex so that the stationary point is the unique global minimum. However, the behavior of the iterative message passing algorithms on loopy graphs is not theoretically understood and in general seems quite complex.

As most communications problems can be represented by bi-partite loopy graphs, denoted as G_T, of variable nodes and check (factor) nodes [28, 29, 59, 70, 71], it is interesting to investigate the behavior of iMPAs over G_T's instead of general graphs. The idea of bi-partite graphical models was first introduced by Tanner [59] to describe families of codes which are generalizations of the LDPC codes [21, 22], thus G_T's are usually referred to as Tanner Graphs. In Tanner's original formulation, all variables are codeword symbols and hence "visible", Wiberg [70] then introduced "hidden" state variables and suggested applications beyond coding. Their ideas were further generalized by Kschischang, Frey and Loeliger [28, 29] to be applied to functions. In this work, we represent our problems in a way similar to Tanner graphs where variable nodes are also referred to as broadcasters and check (factor) nodes are also referred to as modulo-2 adders, respectively, to emphasize their arithmetic operations. Furthermore, a "local function" [28, 29], which was not included in Tanner's original work, is associated with each node.

It should be noted that the bi-partite graphical representation is usually not unique, and it is an open problem in general to find a "good" Tanner graph for a given problem in the sense that the associated message passing algorithms can have bit error rate (BER) or block error rate close to that of an optimal decoder. Previous researches addressed this problem in various ways. Using graph analysis and linear programming, Tanner [60] derived lower bounds on minimum distance for a given LDPC code and suggested that these lower bounds may be closely related to the performance of iterative decoding. His bit-oriented bound and parity-oriented bound obtained by graph analysis were later used by Shin, Kim and Song [52, 53], as a guideline to classify good and bad codes, to analyze and construct *block-wise irregular* LDPC codes. Using similar techniques as those used in [60], Vontobel and Koetter derived an algebraic eigenvalue-based lower bound and a linear-programming-based lower bound on the minimum pseudo-weight of binary linear codes [65], and they argued that the minimum *pseudo-weight* of *pseudo-codewords* relates to the performance of iterative decoding. However, all these eigenvalue-based lower bounds apply to LDPC codes with some degree of regularity only and their linear programming bounds are usually too complicated to be evaluated.

Sipser and Spielman's approach [55] to this problem focused on the expansion property of the Tanner graph, *i.e.,* the ratio between the number of check nodes connected a set of variable nodes and the number of edges incident from the same set of variables. By carefully designing their iterative decoding algorithms, they argued that, for Binary Symmetric Channels (BSCs), their algorithms can correct a number of random errors if the expansion property of underlying graph is good enough. This argument was generalized by Burshtein and Miller [12] to analyze Gallager's hard decoding and soft decoding

(with clipping) [21, 22] algorithms. Using expander-based arguments, they proved that, if the length of the LDPC code is sufficiently large and these algorithms can correct a sufficiently large fraction of the errors, they can eventually correct all errors.

Furthermore, very nice results exist for iterative decoding on loopy Tanner graphs on the erasure channels because the performance of the iterative MPA is completely determined by the *stopping sets* [17] of G_T. The size of the smallest stopping sets was defined as *stopping distance* [37, 51], which is function of the specific graphical representation. The minimum number of *single parity-check* nodes, which a G_T must contain so that its stopping distance equals the minimum distance of the corresponding binary linear code, was defined by Schwartz and Vardy [51] as *stopping redundancy*. It should be noted that the concept of stopping distance is closely related to minimum pseudo-weight of pseudo-codewords [65], as the minimal AWGN-Channel (AWGNC) pseudo-weight is always smaller than the minimal BEC pseudo-weight which equals the minimal stopping set weight.

However, the ultimate goal is still to solve this problem for AWGN channels, which is the most commonly used model for communication systems and probability inference problems. Our approach to this is twofold. First, we want to define some graph metric, which is closely related to the performance of the iterative MPAs. Second, we want to propose a method to compute this metric for any given Tanner graph. Tanner's linear programming bound may be a good graph metric, but it is very complex to obtain. Stopping distance and stopping redundancy point out the direction on how to improve the performance of the iMPAs, but stopping distance itself is not a good metric, and for binary linear code with long block size, exhaustive searches are not feasible.

One interesting observation is that stopping sets have expansion of 1/2, and both stopping distance and the number of smallest stopping sets affect the performance of the iMPAs. Thus, generalized from results for BEC and BSC, it is conceivable that the average expansion property of G_T is a good graph metric. Unfortunately, it is also hard to obtain the exact average expansion of G_T.

Noting that there are some well-defined techniques in the field of spectral graph theory [16], lower bounds on stopping distance are derived for any given bi-partite graph [77] and Tanner's bit-oriented and parity-oriented bounds are then special cases of our results applying to regular LDPC codes. Furthermore, these techniques can aid in deriving a lower bound on the average expansion and it is conjectured that this bound can be used as the graph metric. However, we only define the graph metric in this work, rigorous proof and verification using computer simulations remain largely open.

It should be noted that there are other possible graph metrics. For example, the length and the number of shortest cycles in the graphical representation is conceivable to be a good graph metric too. This work is being carried on by one of my colleagues, Tom Halford [24].

1.2 Outline

This dissertation consists of 6 chapters, and each chapter is written in a self-contained manner. They are organized as following:

In chapter 2, iterative MPAs on sparse loopy graphs are applied to the pseudo random code acquisition problems. Simulation results show that this new approach can solve the

code acquisition problem more efficiently, compared to both simple serial search, full parallel search and hybrid search.

In chapter 3, techniques of eigenvalue analysis for Tanner graphs are introduced and they are used to derived bounds on the variable expansions.

In chapter 4, by analyzing the eigenvalues and eigenvectors of the normalized incidence matrix representing a Tanner graph, we derive lower bounds on its stopping distance. Using these lower bounds, an upper bound on stopping redundancy of the difference-set codes is derived as well.

In chapter 5, we will provide criteria aiming to relate the performance of iterative message-passing algorithms on Tanner graphs with the average variable expansions of these graphs. Though rigorous proofs are not provided, their correctness are suggested by expander-based arguments and an example. Also, possible steps to prove the criteria are provided.

Conclusions and possible directions for follow-on research are discussed in chapter 6.

Chapter 2

Message Passing Techniques for Rapid Code Acquisition

2.1 Introduction

Spread spectrum (SS) techniques are used in many 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 pseudo random or pseudo noise (PN) binary sequence. This procedure is referred to as *spreading* because it results in 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 communication systems. Ultrawideband (UWB) systems are extreme cases of SS systems and are often characterized by low duty cycle trains of very narrow pulses. In all of these cases, spreading is achieved via a PN sequence. To enable autonomous reception, periodic PN sequences are used. For most practical communication systems, long PN sequences (*i.e.,*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 [54].

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 code acquisition are full parallel search, serial search [44, 49] and hybrid search [54]. 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 PN code period. 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 equals the period of the PN sequence, this method is infeasible for practical 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 trade-off between these two extremes (*i.e.,*a hybrid search with four parallel correlators is four times faster and four times as complex as serial search). In this chapter we present the first method for achieving PN acquisition at low SNR as fast as full-parallel search, but with significantly lower complexity. 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 [2, 3, 28, 70]. This is a generalization of the "turbo" decoding algorithm [11]. 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.

It should be noted that there exists another suboptimal method for fast acquisition in the literature, *i.e.,*sequential search [68], which is also known as RASE (Rapid Acquisition

by Sequential Estimation). The basic idea of this strategy is to sequentially estimate the shift register state, *i.e.,* hard decisions, of the PN generator. A decade after the original paper, a modification of the RASE system was reported by Ward and Yiu [69], where a verification scheme was added to reduce the average acquisition time. Though RASE can provide rapid acquisition under certain scenarios, it is not widely used because it has the drawback of being highly vulnerable to noise and interference signals [54]. The main reason is that its estimation process is performed on a chip-by-chip basis and makes no use of the interference rejection capabilities of the PN signals. To address this problem, several modifications of Ward's initial RASE system were investigated, where Pearce and Ristenblatt [38] suggested a threshold decoding type of estimator similar to that used for block codes, Kilgus [26] suggested using the majority logic vote of a number of independent estimates to decide the initial state, and Alem and Weber [5] proposed an optimum Bayes detector instead of the simple threshold decision in the original RASE system. More recently, iterative soft sequential estimation methods were proposed independently by Yang and Hanzo [73, 74], and by Vigoda, Dauwels, Gershenfeld and Loeliger [64], trying to improve the performance of the RASE system. However, graphical representations are not discussed in their works. Since different graphical models and message passing algorithms have been discussed in our work, their work can be considered as a special case. Specifically, their work can be considered as forward-only message passing algorithms on Tanner graphical representation of the PN generator.

Our primary motivation for this problem is a UWB system [50, 72] 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

$$r(t) = \sum_{k=0}^{M-1} \sqrt{E_c}(-1)^{x_k}\omega_r(t - kT_f - \xi T_p) + n(t) \tag{2.1}$$

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*, $n(t)$ is additive Gaussian noise and M is number of pulses observed. It should be noted that, in this work, we focus on the 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 [36]. Other methods use pulse position modulation (PPM) by the PN sequence [72]. In our method, one needs the likelihood of the chip value for a given noisy observation, so application to PPM UWB systems and other models is straightforward.

A sample waveform for a noise-free UWB signal of this form is shown in Figure 2.1(a). The PN acquisition problem is also diagrammed in Figure 2.1 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 Figure 2.1(b), corresponding to one period of the PN

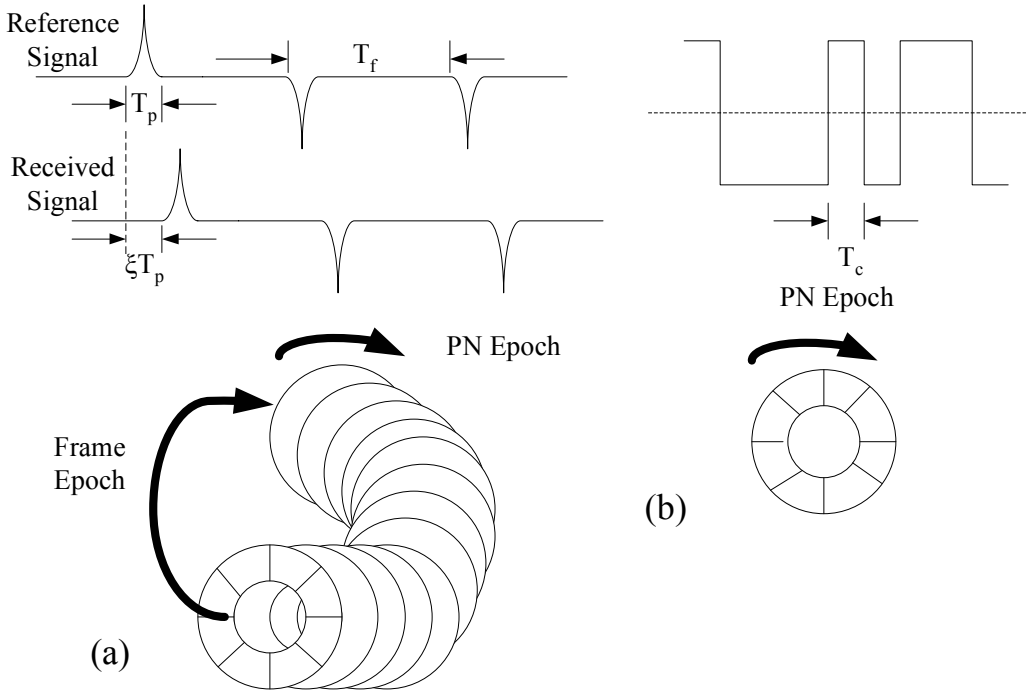

Figure 2.1: A sample waveform and diagram of the associated PN acquisition problem for two spread spectrum systems: (a) a low duty cycle UWB system where the frame epoch and PN code phase must be determined and (b) a direct sequence system where only PN code phase need be acquired. The DS system is modeled with the complex baseband equivalent signal.

code. The corresponding diagram for the UWB system is the "PN-phase/frame-epoch taurus" shown in Figure 2.1(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 [61]. More specifically, if the bins in Figure 2.1(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 chapter provides an attractive solution to this problem that cannot be achieved using traditional PN acquisition strategies. Similar methods have been applied to the sparse inter-symbol interference (S-ISI) channels by Chen [13][14, Ch. 3].

This chapter is organized as follows. Section 2.2 contains the signal models considered, Section 2.3 contains approximate analysis of the traditional approaches to PN acquisition, and Section 2.4 describes the graphical modeling and iMPAs applied to PN acquisition. Simulation results are provided in Section 2.5 and conclusions are drawn in Section 2.6.

2.2 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 [23]. 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 Figure 2.2(a). At time k, let x_k be the output, so that x_{k+i}, $0 \leq i \leq r-1$ is the value of the i^{th} register and the constraint is

$$0 = g_0 x_{k+r} \oplus g_1 x_{k+r-1} \oplus ... \oplus g_{r-1} x_{k+1} \oplus g_r x_k \tag{2.2}$$

where $\oplus$ is modulo 2 addition and $g_i \in \{0, 1\}$, $0 \leq i \leq r$, are feedback coefficients. The *generating polynomial* is $g(D) = g_0 + g_1 D + ... + g_{r-1} D^{r-1} + g_r D^r$, where D is the unit delay operator [23]. The maximum achievable period of a r-stage LFSR is $N = 2^r - 1$

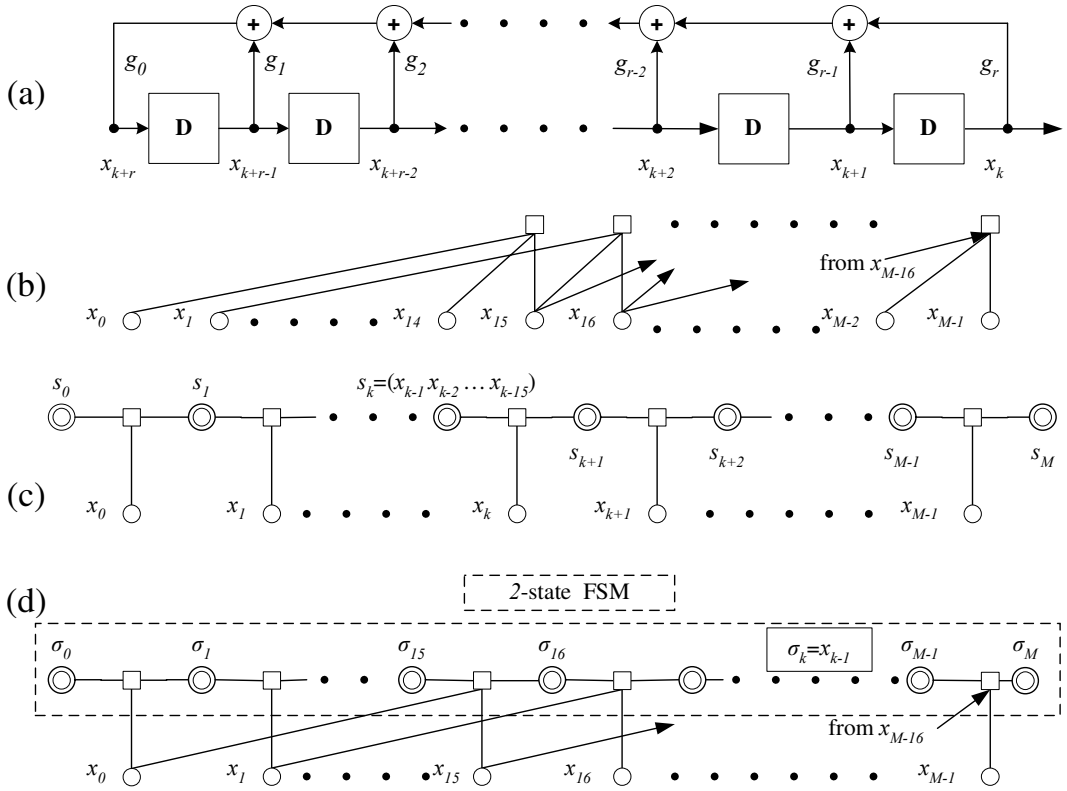

Figure 2.2: Methods for modeling LFSRs. Part (a) shows the generator diagram for an r-stage LFSR. Parts (b)-(d) show different graphical models for the same 15-stage LFSR with $g(D) = 1 + D + D^{15}$.

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 $\underline{x} = x_0, x_1, ...x_{r-1}, x_r, x_{r+1}, ...x_{N-1}, x_N, ...$, 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 $\mathbf{u} = (x_0, x_1, ...x_{r-1})^T$, where T denotes transposition.

The goal of code acquisition is to find the phase of the sequence present in the received signal, where $\underline{x}$, $D\underline{x}$, ..., $D^{N-1}\underline{x}$ 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 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

$$z_k = \sqrt{E_c} y_k(\mathbf{u}) \cdot e^{j\theta_c} + n_k = \sqrt{E_c}(-1)^{x_k(\mathbf{u})} \cdot e^{j\theta_c} + n_k, \qquad 0 \leq k \leq M-1 \tag{2.3}$$

This model captures both the DS/SS and UWB systems illustrated in Figure 2.1 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 [54]. The model in (2.3) can be written in vector form as

$$\mathbf{z} = \sqrt{E_c}\mathbf{y}(\mathbf{u})e^{j\theta_c} + \mathbf{n} \tag{2.4}$$

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

Since the simple model in (2.3) is common in the DS/SS literature, let us consider how it applies to the UWB system modeled in (2.1), where $n(t)$ is AWGN with power spectral density level of $N_0/2$. Using an estimate the frame epoch, $\widehat{\xi}$, the discrete observation z_k is obtained by the following processing: during the time interval $[kT_f, (k+1)T_f)$, the UWB receiver aligns a pulse matched filter at $kT_f + \widehat{\xi}T_p$, and samples the output at $kT_f + (\widehat{\xi}+1)T_p$. If ξ is constant over the time interval $[0, MT_f]$ and $\widehat{\xi} = \xi$, a model of the form (2.3) results. Specifically, since the UWB signal in (2.1) has no sinusoidal carrier, the real part of z_k from (2.3) 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 seconds. 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 traditional 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 (2.3)-(2.4) for these two cases: (i) the DS/SS system with no carrier phase knowledge and (ii) 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.

2.3 Performance Characteristics of Traditional Acquisition Methods

As briefly described in Section 2.1, the three widely used 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 (2.3) there are N possible PN phases denoted by $\mathbf{u}_i$, $0 \leq i \leq N-1$. The normalized correlation to the i-th PN phase using M observations is

$$r_i = r(\mathbf{u}_i) = \frac{1}{M}\mathbf{y}^T(\mathbf{u}_i)\mathbf{z} \tag{2.5}$$

where $\mathbf{y}(\mathbf{u}_i)$ is the noise-free signal in (2.4) 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 [23] 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 (2.5) is similar to the correlator output of 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

$$r_0 = \frac{1}{M}\mathbf{y}^T(\mathbf{u}_0)\mathbf{z} = \sqrt{E_c}e^{j\theta_c} + \omega_0 \tag{2.6}$$

$$r_i = \frac{1}{M}\mathbf{y}^T(\mathbf{u}_i)\mathbf{z} = \sqrt{E_c}\frac{1}{M}\mathbf{y}^T(\mathbf{u}_i)\mathbf{y}(\mathbf{u}_0)e^{j\theta_c} + \omega_i \quad 1 \leq i \leq N-1 \tag{2.7}$$

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 $\frac{N_0}{2M}$. For m-sequences it can be shown [23, 54] that the set of random variables $\{r_i\}_1^{N-1}$ can be approximately modeled as i.i.d., zero-mean, complex Gaussian random variables with variance $\frac{2 \cdot E_c + N_0}{2M}$ in the imaginary and real parts. Specifically, the non-zero 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 2.5.

2.3.1 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 $\widehat{\mathbf{u}} = \arg\max_{\mathbf{u}_i} p(\mathbf{z}|\mathbf{u}_i)$, where $p(\mathbf{z}|\mathbf{u}_i)$ is the likelihood of $\mathbf{u}_i$ and $\mathbf{z}$ is defined in (2.4). 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 (2.6)-(2.7), since the set of correlations $\{r_i\}_{i=0}^{N-1}$ is a set of sufficient statistics for the model of (2.3). 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 despreader 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

$$\begin{aligned} P_{\text{ACQ}}^{(C)} &= \int_{-\infty}^{\infty} P(v_1 < v_0, .., v_{N-1} < v_0 | v_0) p(v_0) dv_0 \\ &\approx \int_{-\infty}^{\infty} \left[1 - Q\left(\frac{t + \sqrt{\frac{2ME_c}{N_0}}}{\sqrt{\frac{2E_c}{N_0} + 1}} \right) \right]^{N-1} \frac{e^{-\frac{t^2}{2}}}{\sqrt{2\pi}} dt \end{aligned} \tag{2.8}$$

where $Q(\cdot)$ is the complementary cumulative distribution function of a standard Gaussian random variable, defined as

$$Q(t) = \int_t^{\infty} \frac{e^{-u^2/2}}{\sqrt{2\pi}} du$$

The probability of acquisition of noncoherent full parallel search can be computed using the same approximation in (2.6)-(2.7). 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

$$P_{\text{ACQ}}^{(NC)} \approx \int_0^{\infty} \left[1 - e^{\frac{-t^2}{\frac{4E_c}{N_0} + 2}} \right]^{N-1} t \exp\left[-\frac{t^2}{2} - \frac{ME_c}{N_0} \right] \mathrm{I}_0 \left[\sqrt{\frac{2ME_c t}{N_0}} \right] dt \tag{2.9}$$

where $\mathrm{I}_0(\cdot)$ is the modified Bessel function of zeroth order [45, Ch.2].

2.3.2 Simple serial search

For the simplified model in (2.3), simple serial search computes the likelihood for one candidate initial phase $p(\mathbf{z}|\mathbf{u}_i)$ using the set of M observations, $\mathbf{z}$ [44]. More precisely, for the UWB, the real part of r_i is compared to a threshold, and for the noncoherent DS/SS case $|r_i|$ is compared to 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.[1] In this case, the M observations correspond to one *dwell time* [44], T_d, which are assumed to be non-overlapping. 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 [44]:

$$T_{\text{ACQ}}^{(s)} = \frac{2 + (2 - P_D)(N - 1)(1 + KP_{FA})}{2P_D} \cdot T_d \tag{2.10}$$

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 $P_{FA} = 0$, we have $\frac{T_{\text{ACQ}}}{T_d} = \frac{N+1}{2} = 2^{r-1}$. So, unlike full parallel search, simple serial search takes much more than M chip times to acquire on average.

Also, it can be shown that[2]

$$P_{\text{ACQ}}^{(s)} = \frac{P_D \cdot [1 - (1 - P_{FA})^N]}{NP_{FA}} \tag{2.11}$$

[1] 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.

[2] 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 [54]. Also, an absorbing false alarm state [54] is assumed.

where P_D and P_F are the probability of detection and false alarm, respectively, for a single dwell test. Specifically, P_D (P_{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 (2.8) and (2.9).

It is also conceivable to use a serial search approach on a single dwell observation. This would be the case for example, if very fast signal processing resources were reused to correlate the same set of M observations before another set of M observation was available. This is not considered here and we reserve the term serial search to describe the traditional approach summarized above.

2.3.3 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 [54]

$$P_{FA}^{(h)} = 1-(1-P_{FA})^{C_p} \tag{2.12}$$

$$P_{D}^{(h)} = 1-(1-P_{D})(1-P_{FA})^{C_p-1} \tag{2.13}$$

$$T_{\mathrm{ACQ}}^{(h)} = \frac{2-P_D-(1-P_D)(C_p-1)P_{FA}}{2[P_D+(1-P_D)(C_p-1)P_{FA}]}\cdot\frac{1+KC_pP_{FA}}{C_p}\cdot(NT_d) \tag{2.14}$$

where $P_{FA}^{(h)}$, $P_D^{(h)}$ 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, $P_{FA}^{(h)} \simeq C_pP_{FA}$, $P_D^{(h)} \simeq 1-(1-$

$P_D)[1-(C_p-1)P_{FA}]$ and $T_{\text{ACQ}}^{(h)} \simeq \frac{1}{C_p} T_{\text{ACQ}}^{(s)}$. 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 $T_{\text{ACQ}}^{(h)}$ will be achieved [54]. The probability of acquisition of hybrid search can be obtained using equation (2.11), with P_D, P_{FA} and N replaced by $P_{FA}^{(h)}$, $P_D^{(h)}$ and N/C_p respectively.

2.4 Graphical Models and iMPAs for Code Acquisition

2.4.1 Graphical models for m-sequence $[100003]_8$ and the associated iMPAs

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[3]. 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 Figure 2.2(b)

[3]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.

for the m-sequence with generating polynomial $g(D) = 1 + D + D^{15}$ ($[100003]_8$) 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 – *i.e.,* $C = 4$ 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 Figure 2.2(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 – *i.e.,* 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

$$\begin{aligned} \mathrm{M}[c] &= \textstyle\sum_i \mathrm{MI}[x_i(c)] \quad \text{(combining)} && (2.15) \\ \mathrm{MO}[x_i] &= \min_{c:x_i} \mathrm{M}[c] - \mathrm{MI}[x_i] \quad \text{(marginalization)} && (2.16) \end{aligned}$$

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

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

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 $\mathrm{MI}_{cc}[\cdot]$ ($\mathrm{MO}_{cc}[\cdot]$).

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 l-th check node to a variable node, then the output message returned to the l-th check is

$$\mathrm{MO}_v[x^{(l)}] = \mathrm{M}_{ch}[x = x^{(l)}] + \sum_{m=0, m\neq l}^{L-1} \mathrm{MI}_v[x^{(m)} = x^{(l)}] \tag{2.17}$$

which should be interpreted as an equation for each conditional value of x – *i.e.,* for binary x, one for $x^{(l)} = 0$ and another for $x^{(l)} = 1$. Note that this is equivalent to (2.15)-(2.16) 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 Figure 2.2(b), which is connected 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 $\mathrm{M}_{ch}[x_k]$ 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 equations (2.15), (2.16) and (2.17) 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 [14] where

$$\min{}^*(x, y) = \min(x, y) - \ln\left(1 + e^{|x-y|}\right) \tag{2.18}$$

Specifically, min*-sum algorithms perform the processing in (2.15), (2.16) and (2.17) 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 sub-optimal heuristics, which we refer to as *iterative* message passing algorithms (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 near-optimal performance. 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 $N = 2^r - 1 = 32767$ configurations, while the graph of Figure 2.2(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 Figure 2.2(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.[4] For example, the same m-sequence modeled in Figure 2.2(b) can be modeled by the cycle-free graphical model in Figure 2.2(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)* [14]. 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

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

PN sequence acquisition. This illustrates the importance of cycles in the graphical model to achieve low complexity iMPAs.

A third graphical model for the m-sequence with $g(D) = 1 + D + D^{15}$ is shown in Figure 2.2(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 Figure 2.2 all completely capture the constraint of the m-sequence structure fully and without redundancy. The graph in Figure 2.2(d) can also be viewed as decomposing the 15-stage shift register into a 2-stage shift register with a long delayed, feedback loop. This is emphasized by the box in Figure 2.2(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 running on the graph in Figure 2.2(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

$$\mathrm{M}[x_k] = \mathrm{M}_{ch}[x_k] + \sum_{m=0}^{L-1} \mathrm{MI}[x^{(m)} = x_k] \quad 0 \le k \le M-1 \tag{2.19}$$

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,

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	$[20{,}000{,}000{,}011]_8$,$[22{,}000{,}000{,}001]_8$

Table 2.1: Examples of sparse generating polynomials for m-sequences [42].

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 $\lfloor M/r \rfloor$ non-overlapping 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.

2.4.2 Graphical models for other m-sequences

Careful inspection of the development in the previous section 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 Figure 2.2(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 [42] are listed in Table 2.1.

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 Figure 2.2(b) can be constructed for any LFSR with $g(D)$ specified. If there are some groupings of non-zero terms in the feedback polynomial, then one may consider defining an FSM to capture these local constraints as was done in Figure 2.2(d), for example.

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 Figure 2.2(b), is shown in Figure 2.3(a). Another model is shown in Figure 2.3(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.

2.4.3 Relation to LDPC codes and further reading

An Low-Density Parity-Check (LDPC) code [22, 34] is a linear parity check code with a parity check matrix that has a small number of nonzero entries. Specifically, every valid codeword $\mathbf{c}$ satisfies $\mathbf{Hc} = \mathbf{0}$ where $\mathbf{c}$ is an $(n \times 1)$ binary column 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 [30]. The standard graphical

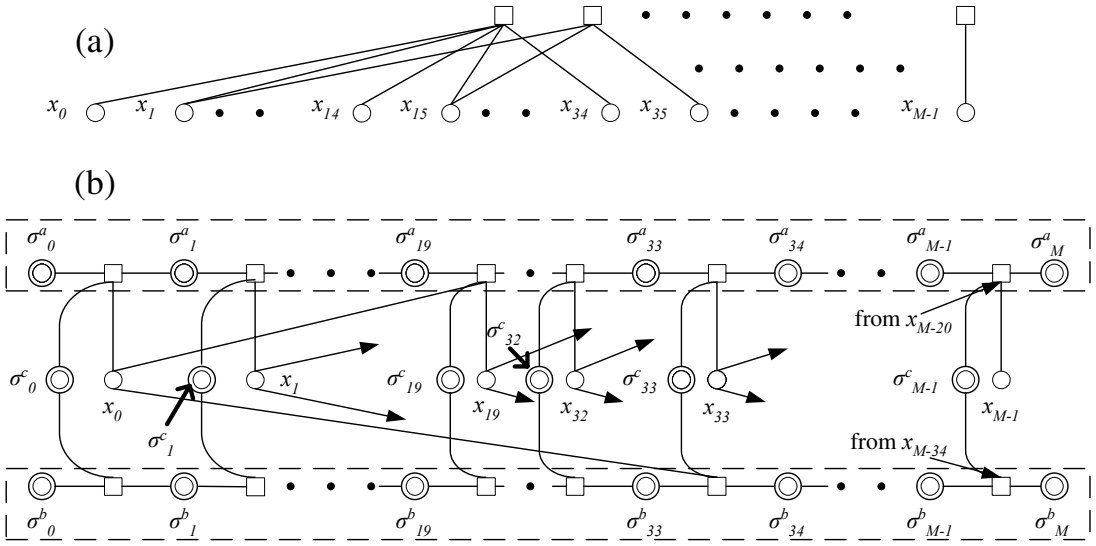

Figure 2.3: Two graphical models for the 34-stage LFSR with $g_{34}(D) = 1 + D^{19} + D^{20} + D^{33} + D^{34}$.

model for this is similar to that shown in Figure 2.2(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 Figure 2.2(b) is the same as the standard iterative decoder for LDPC codes. In fact, the structure imposed by the LFSR in (2.2) can be written as $\mathbf{H}_{LFSR}\mathbf{x} = \mathbf{0}$, where $\mathbf{H}_{LFSR}$ 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 [30] 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 $\mathbf{H}_{LFSR}$ implies a

relatively localized set of variable constraints and a very regular cycle structure. Both of 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 Figure 2.3(b) is very similar to that of a parallel concatenated convolutional code or "turbo" code [11]. 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 to 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 [28], which generalizes the earlier work of Wiberg [70]. Wiberg generalized the work of Tanner [59], who developed graphical models without hidden variables for linear block codes analogous to that shown in Figure 2.2(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 [3, 35] and make connection to known approaches in computer science [40]. In some of these conventions, directed graphs are used for modeling [14, 35], 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.

2.5 Simulation results

2.5.1 Simulation results for m-sequence $[100003]_8$

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 2.3. The threshold for both serial and hybrid searches is determined using P_{FA} of 10^{-6}. Algorithms are evaluated using P_{ACQ} v.s. E_c/N_0, acquisition time, and complexity.

2.5.1.1 UWB Systems with perfect knowledge of frame epoch

The performance of serial, full-parallel, hybrid, and the iMPA corresponding to Figure 2.2(d) is shown in Figure 2.4. 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 model of Figure 2.2(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 [14]. 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 running on Figure 2.2(d) is about 0.5 dB better than the iMPA running on Figure 2.2(b) (*i.e.,*the Tanner Graph).

The acquisition times of these algorithms are also given in Figure 2.4. Both full parallel search and iterative MPAs achieve code acquisition in $128T_c$, where T_c is the chip

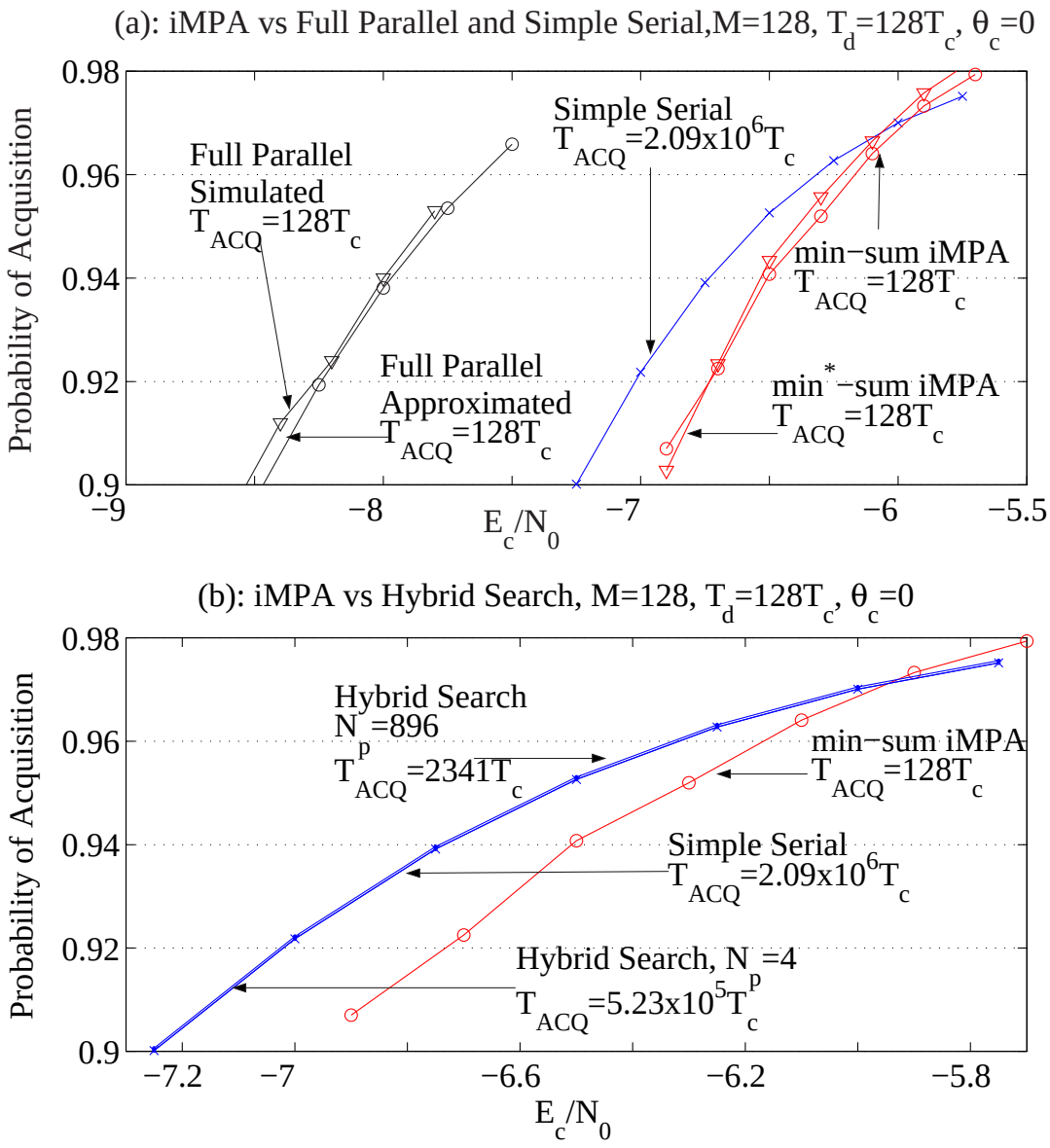

Figure 2.4: Comparison of acquisition performance of various approaches for the UWB system with perfect frame synchronization and m-sequence generated by $g(D) = 1 + D + D^{15}$. All iMPA simulations are based on 100 iterations. Simple serial and hybrid searches use $M = 128$ chip times per dwell while the iMPA and full parallel approaches use $M = 128$ total observations. Part (a) compares the iMPA against the traditional simple serial and full parallel approaches and (b) compares against hybrid search.

interval (the frame time for the UWB system in Figure 2.1(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 $2341 T_c$, respectively. Thus, the iMPAs are 16000 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 2.2. 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 2.2, 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.

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

Table 2.2: 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 Figure 2.2(d).

As illustrated in Figure 2.5, 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 [23] 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 Figure 2.6, where the minimum value of E_c/N_0 required to achieve $P_{\text{ACQ}} = 0.9$, $(E_c/N_0)_{req}$, is plotted against M for the various approaches. The degradation in $(E_c/N_0)_{req}$ for the iMPA 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 Figure 2.2(d) – *i.e.*,each variable is involved in a cycle with minimum length 30.

Finally, as demonstrated in Figs. 2.4-2.5, the approximate analysis in Section 2.3 matches the simulated performance for full parallel search reasonably well. In the subsequent results, only the approximation analysis from Section 2.3 is presented.

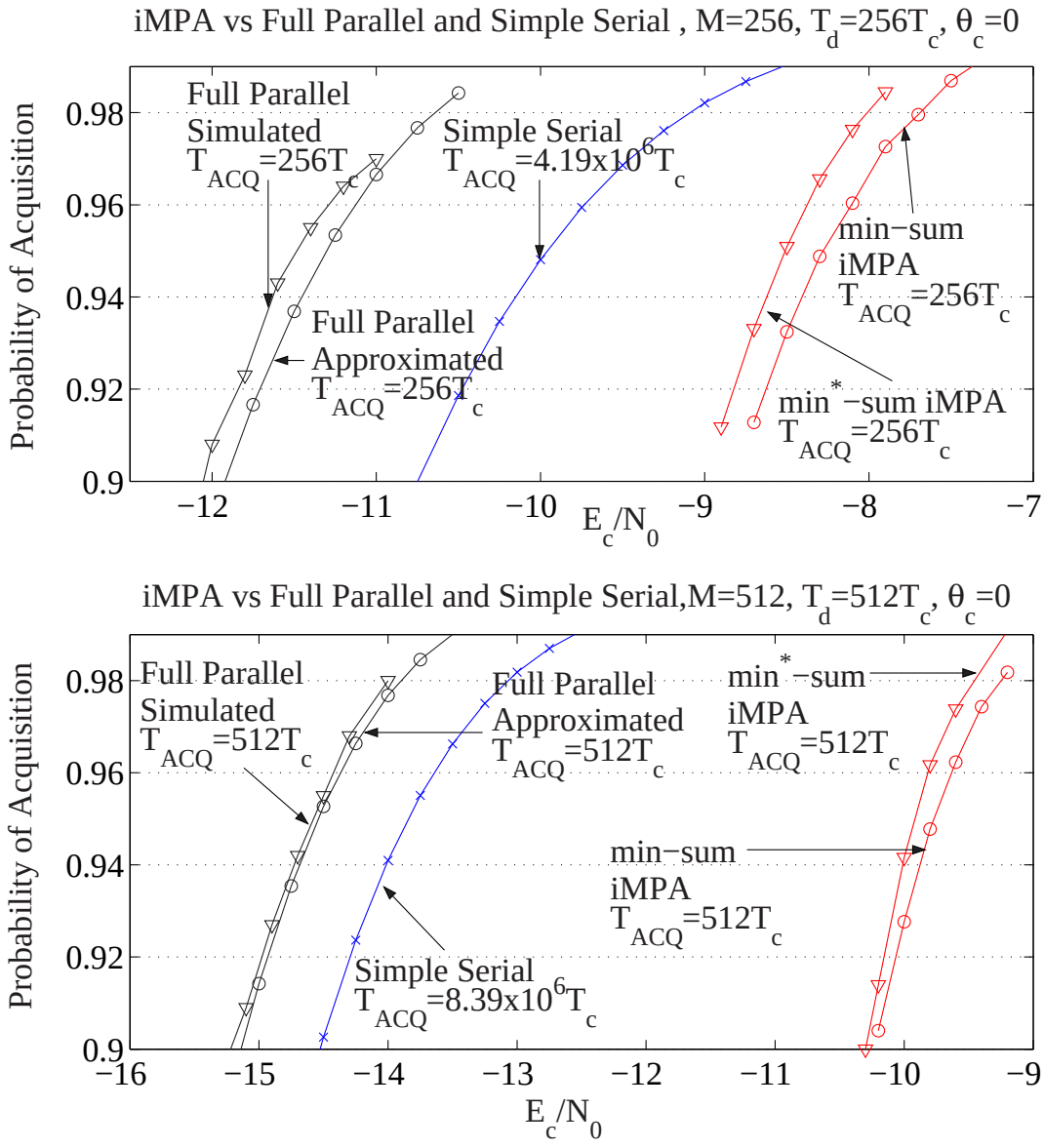

Figure 2.5: The effects of increasing the observation window for various approaches for the UWB system with perfect frame synchronization and m-sequence generated by $g(D) = 1 + D + D^{15}$. All iMPA simulations are based on 100 iterations. Simple serial search use M chip times per dwell while the iMPA and full parallel approaches use M total observations. Part (a) shows $M = 256$ and (b) shows $M = 512$.

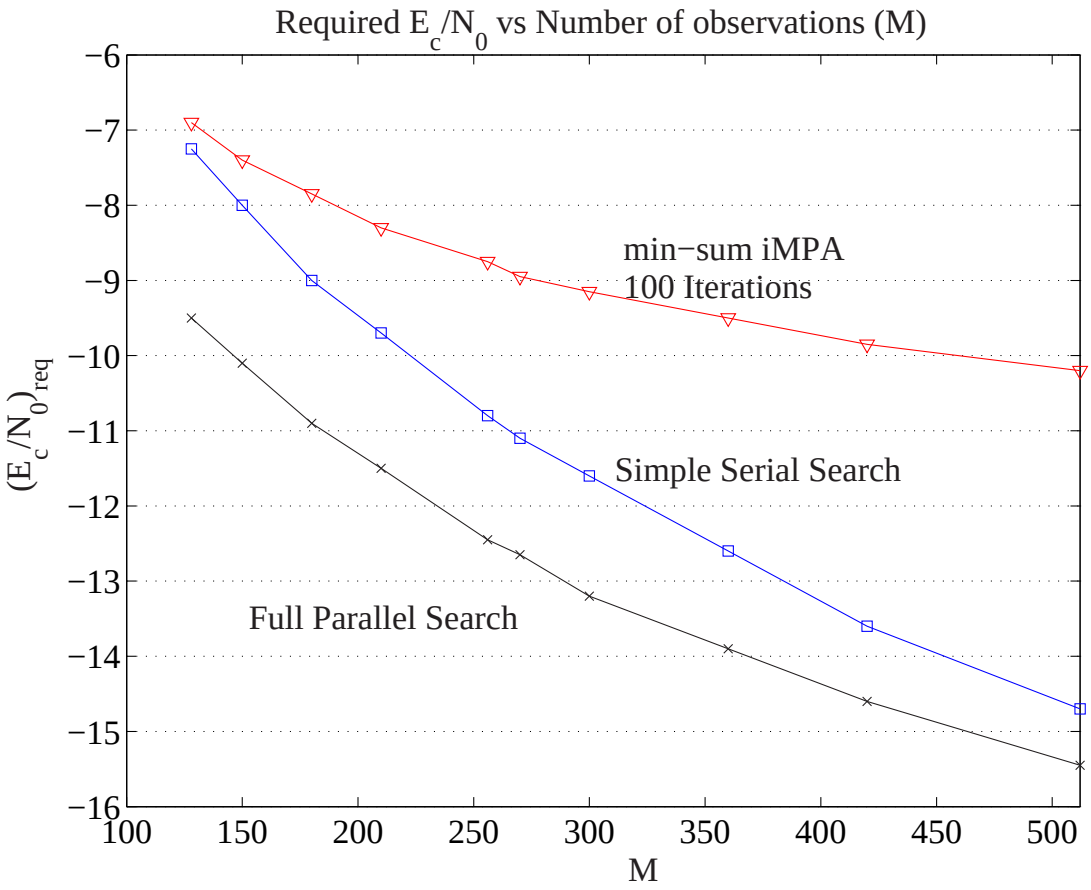

Figure 2.6: Summary of the performance gain obtained with larger observation windows for the iMPA, serial search, and full parallel search for the UWB system with perfect frame synchronization and m-sequence generated by $g(D) = 1+D+D^{15}$. Traits are summarized using $(E_c/N_0)_{req}$ vs. M where $(E_c/N_0)_{req}$ is the lowest SNR for which $P_{\text{ACQ}} = 0.9$ can be achieved. Both full parallel search and min-sum iMPA have M observations and $T_d = MT_c$ in simple serial search.

2.5.1.2 Traditional DS/SS systems with no carrier phase knowledge

As described in Section 2.3, 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 sub-structure (*i.e.,* the two state FSM structure in Figure 2.2(d)).

In order to apply the iMPA approach for the noncoherent DS/SS case, we use a method based on generalized likelihood [14]. Specifically, a finite number of candidate θ_c values are considered. For example, suppose four candidate phase values were considered: $\widetilde{\theta}_c \in \{0, \pi/2, \pi, 3\pi/2\}$. Then, four versions of the iMPA can be run, each using $\mathrm{M}_{ch}[x_k] = p(z_k e^{-j\widetilde{\theta}_c}|x_k)$ for the specific value of $\widetilde{\theta}_c$. 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 Figure 2.7 along with the curves of the ideal case where θ_c is known. The 8 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 4 candidate phase values are used.

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 [14, ch. 4] can also be applied and other unknown parameters (*i.e.,*a frequency offset) could be included as well using similar techniques.

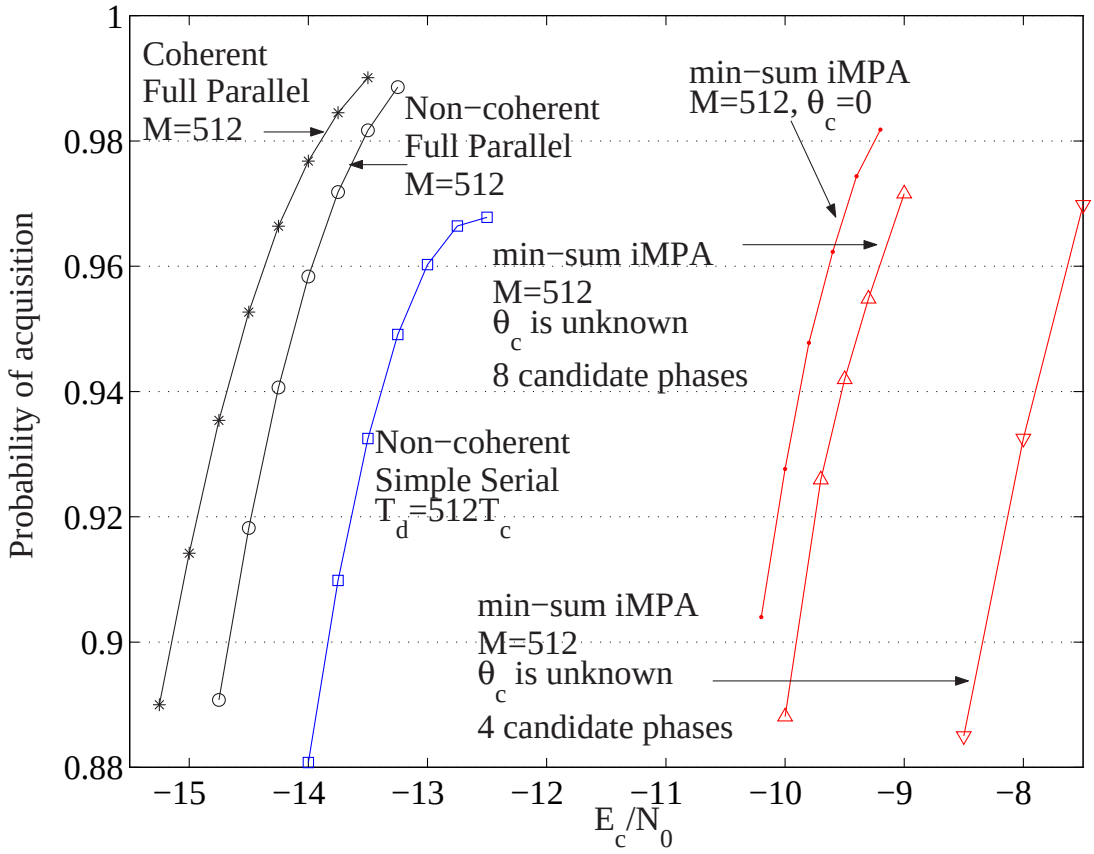

Figure 2.7: Performance of iterative MPA in traditional DS/SS system with unknown carrier phase and m-sequence generated by $g(D) = 1 + D + D^{15}$. The block size M of both full parallel search and 100-iteration min-sum iMPA on Figure 2.2(d) is 512, dwell time T_d for simple serial search is $512T_c$

2.5.2 Simulation results for other m-sequences

In Section 2.5.1, the iMPA based on the graphical model in Figure 2.2(d)) was investigated for one specific m-sequence with $g(D) = 1 + D + D^{15}$, where 3 non-zero g_i's appear at the two ends. Noting that since binary primitive polynomials have at least 3 non-zero g_i's and the shortest cycle in the graphs representing of the form in Figure 2.2(d)) 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: $g_{22}(D) = 1 + D + D^{22}([20000003]_8)$, $g_{18}(D) = 1 + D^{11} + D^{18}$ $([1004001]_8)$, $g_{15}(D) = 1 + D^5 + D^6 + D^8 + D^{10} + D^{12} + D^{15}([112541]_8)$ and $g_{34}(D) = 1 + D^{19} + D^{20} + D^{33} + D^{34}$ $([300006000001]_8)$. 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 iMPA_{22}, iMPA_{18}, 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 Figure 2.2(d) with $\sigma_k = x_{k-1}$, iMPA_{18} and iMPA_{15} are based on (Tanner) graphs similar to that in Figure 2.2(b), and iMPA_{34} is based on the graph in Figure 2.3(b). For comparison purposes, the m-sequence used in Section 2.5.1 is denoted as $\underline{x}_0$ and the corresponding iterative MPA is denoted as iMPA_0.

Figure 2.8 contains simulation results for iMPA_{22}. Since $g_{22}(D)$ has three non-zero coefficients appearing at the two ends, it is another "most favorable" m-sequence with longer period $N = 2^{22} - 1 = 4194303$. Comparing with curves in Figure 2.4 and 2.5, we

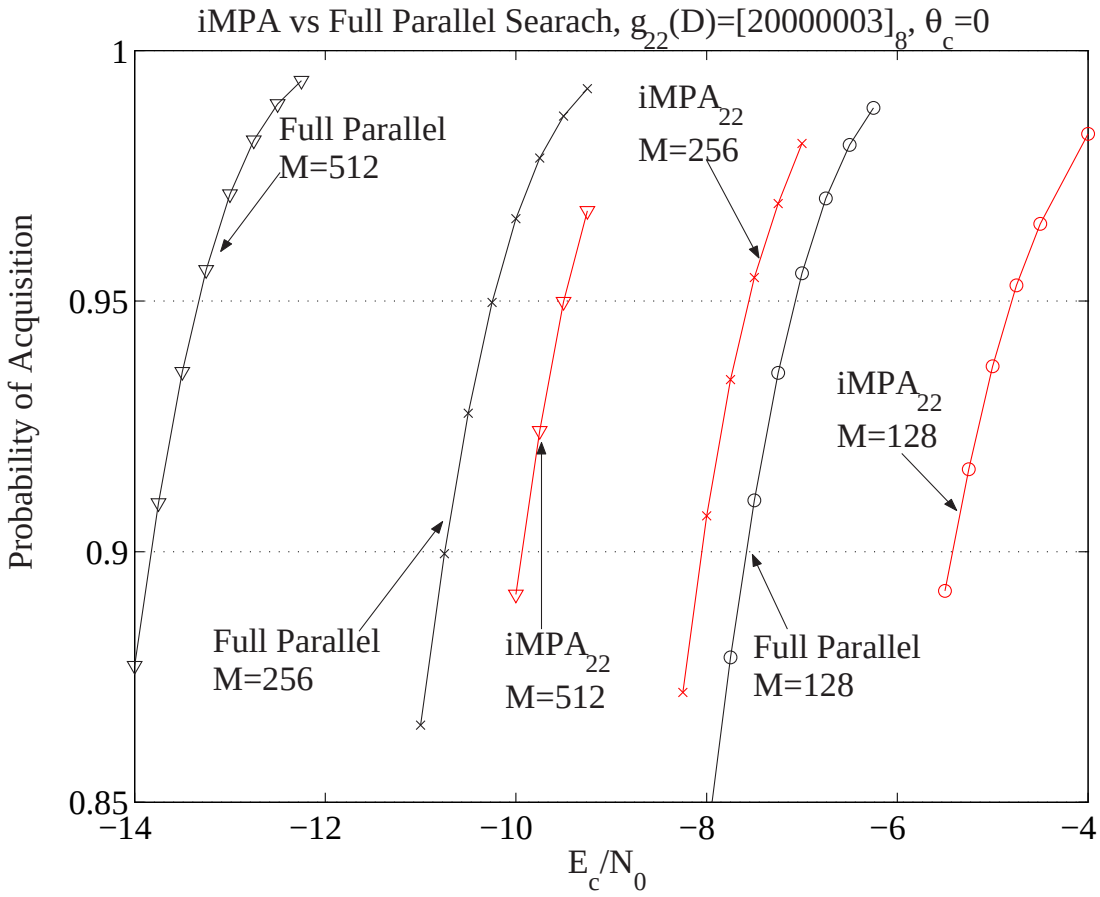

Figure 2.8: Performance of iMPA$_{22}$: 100 iteration min-sum, $g_{22}(D) = D^{22} + D + 1$, $N = 2^{22} - 1 = 4,194,303$ for the UWB system with perfect frame synchronization.

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

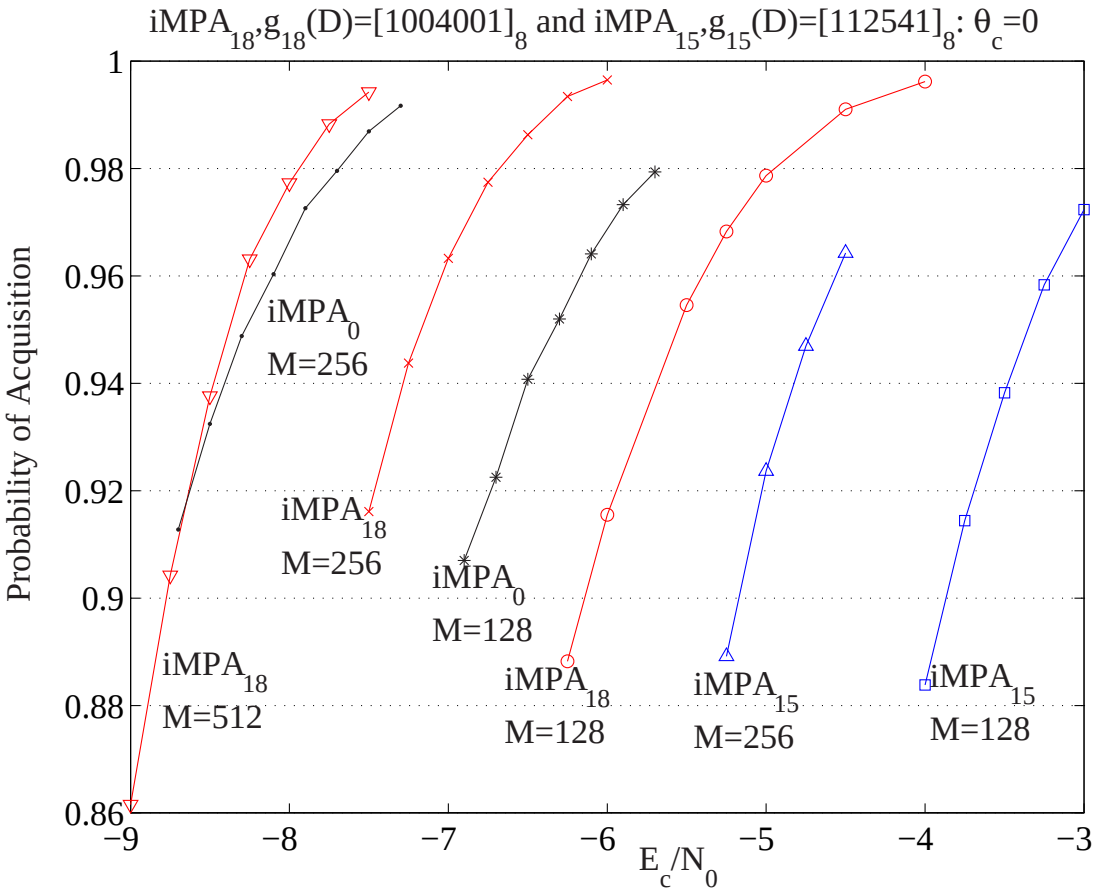

Figure 2.9: Performance of iMPA_{18} and iMPA_{15}: 100 iteration min-sum processing on Tanner graphs. For the UWB system with perfect frame synchronization.

Figure 2.9 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, 6 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 Figure 2.10. This includes results for both iMPA_{34}, based on the graph in Figure 2.3(b), and the iMPA based on the graph

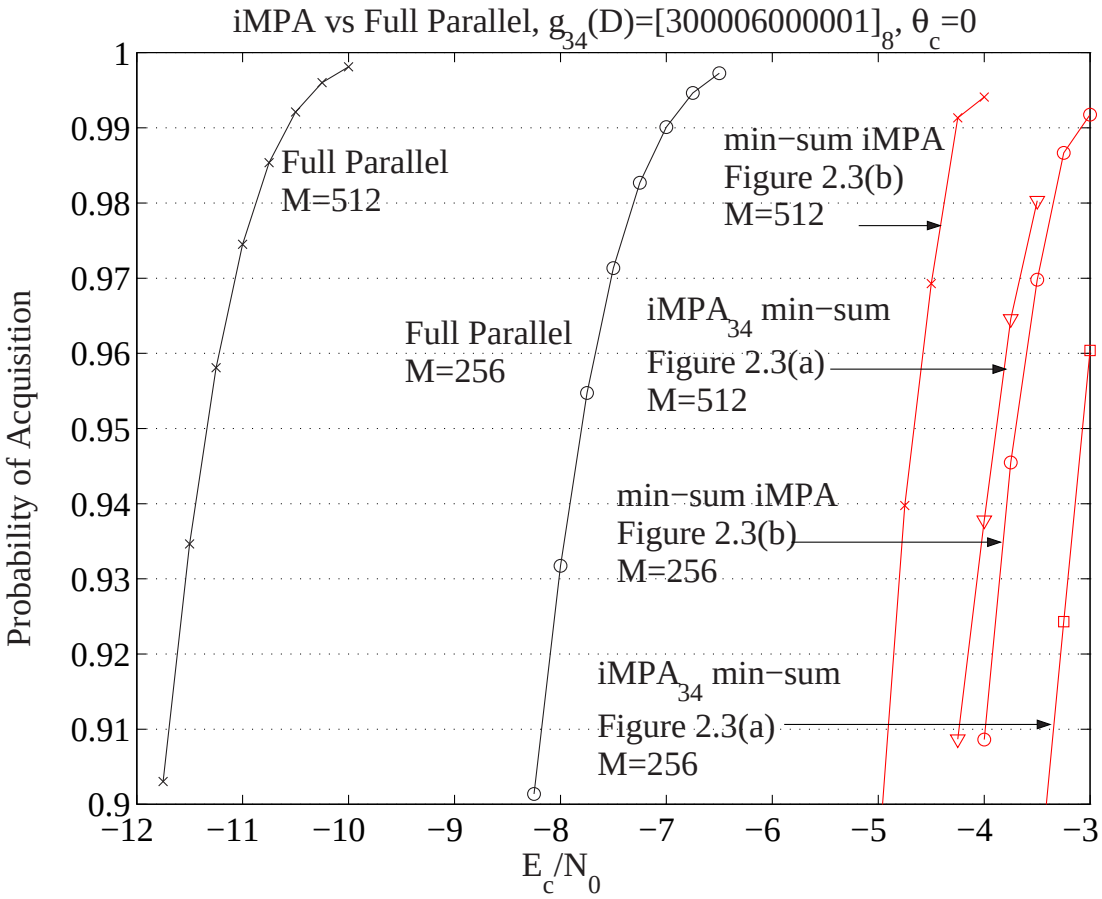

Figure 2.10: Performance of iMPA for 34-stage LFSR with $g_{34}(D) = 1 + D^{19} + D^{20} + D^{33} + D^{34}$: 100 iteration min-sum on Figure 2.3(a) and Figure 2.3(b). For the UWB system with perfect frame synchronization.

of Figure 2.3(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 [76], the method of filtering messages to damp out rapid fluctuations was applied to the problem at hand. Considering an edge labelled a_k, and let the standard *extrinsic information* [14] after the n-th iteration be $\mathrm{MO}^{(n)}[a_k]$, the actually message passed along the edge is

$$\mathrm{MO}_f^{(n)}[a_k] = g \cdot (h_0 \cdot \mathrm{MO}^{(n)}[a_k] + h_1 \cdot \mathrm{MO}^{(n-1)}[a_k]) \tag{2.20}$$

where

$$H(X) = h_0 + h_1 X^{-1} = \frac{\beta}{\sqrt{\beta^2 + (1-\beta)^2}} + \frac{1-\beta}{\sqrt{\beta^2 + (1-\beta)^2}} \cdot X^{-1}$$

is a unit-gain low-pass filter, g is the gain and $\mathrm{MO}_f^{(n)}[a_k]$ is the filtered soft-out information. The parameter β is used to adjust the bandwidth of the filter. Specifically, when $\beta = 1$, there is no filtering.

Simulation results for 100-iteration min-sum iMPA with and without filtering on Figure 2.2 (d) for the case that $M = 128$ and $M = 256$ are plotted in Figure 2.11 (a) and Figure 2.11 (b) respectively. It can be seen that the method of soft-information filtering can not improve the performance significantly. Specifically, it yields a performance gain of approximately 0.3 dB when $M = 128$ and 0.6 dB when $M = 256$. On the other hand, this method increases the memory complexity of the min-sum iMPA significantly. Therefore, it can be concluded that the method of soft-information filtering is not very useful to our problem.

In the following section, we will suggest another approach that achieves a similar performance enhancement with less complexity than the baseline iMPA.

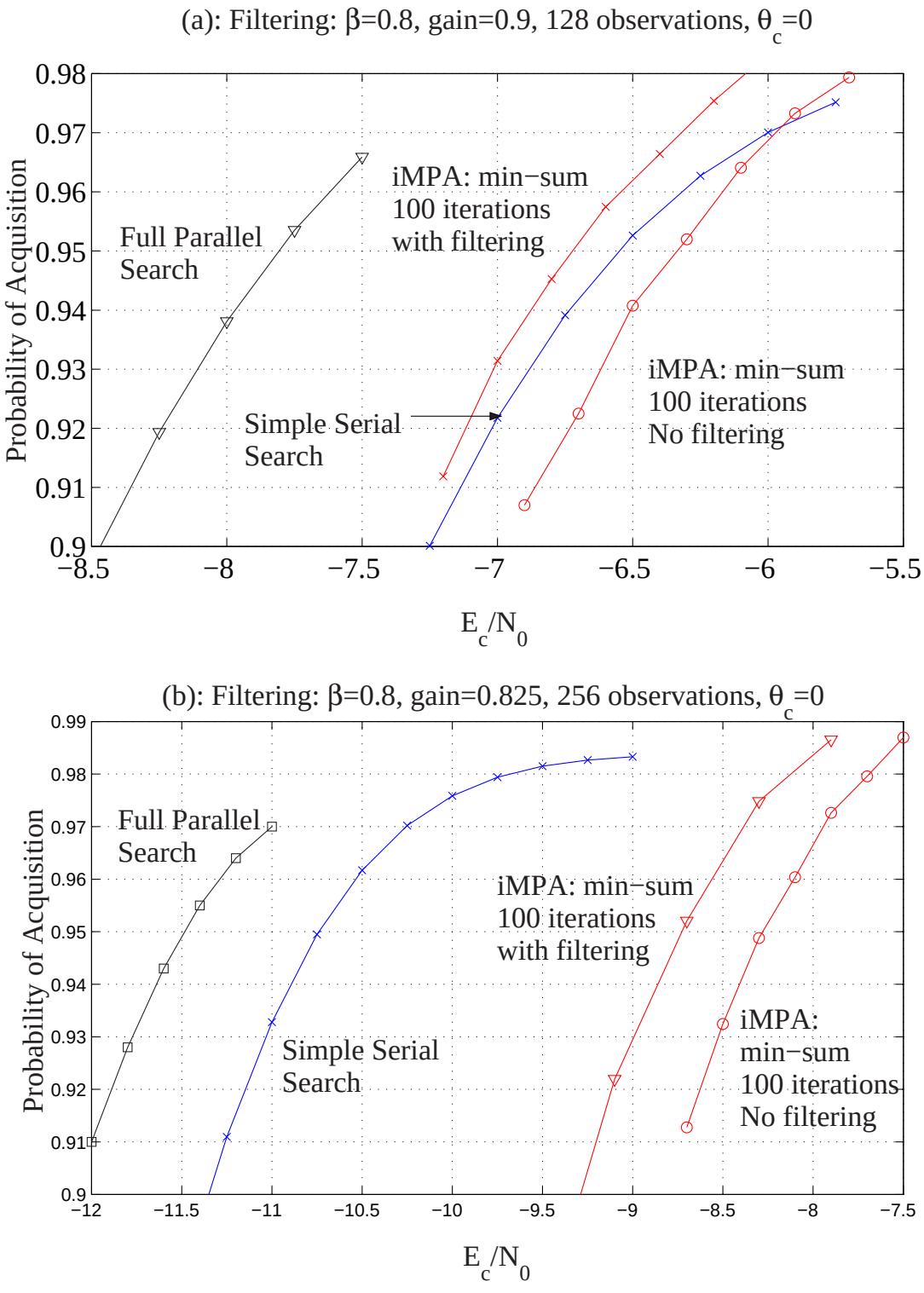

Figure 2.11: Improve the performance of iMPA using soft-information filtering: 100-iteration min-sum iMPA on Figure 2.2 (d), (a) $M = 128$; (b) $M = 256$.

2.5.3 Verification scheme

A verification scheme is required if there is the possibility that no signal is present. For example, in the UWB system in Figure 2.1(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 post-processing 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 $\widehat{\mathbf{u}}$ is obtained and the correlation statistic $v(\widehat{\mathbf{u}}) = \Re\{r(\widehat{\mathbf{u}})\}$, where $r(\mathbf{u})$ is defined in (2.5), is computed. If $v(\widehat{\mathbf{u}}) > \eta$, acquisition is declared, otherwise the observation set is perturbed and the process is repeated. Assuming that P_{ACQ} is required to be at least 0.9, which is commonly used in the code acquisition literature [54], the threshold can be selected as

$$P_r\{v(\widehat{\mathbf{u}}) > \eta\} = Q\left(\frac{\eta - \sqrt{E_c}}{\sqrt{N_0/2M}}\right) \geq 0.9 \quad \longrightarrow \quad \eta = Q^{-1}(0.9) \cdot \sqrt{N_0/2M} + \sqrt{E_c} \quad (2.21)$$

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

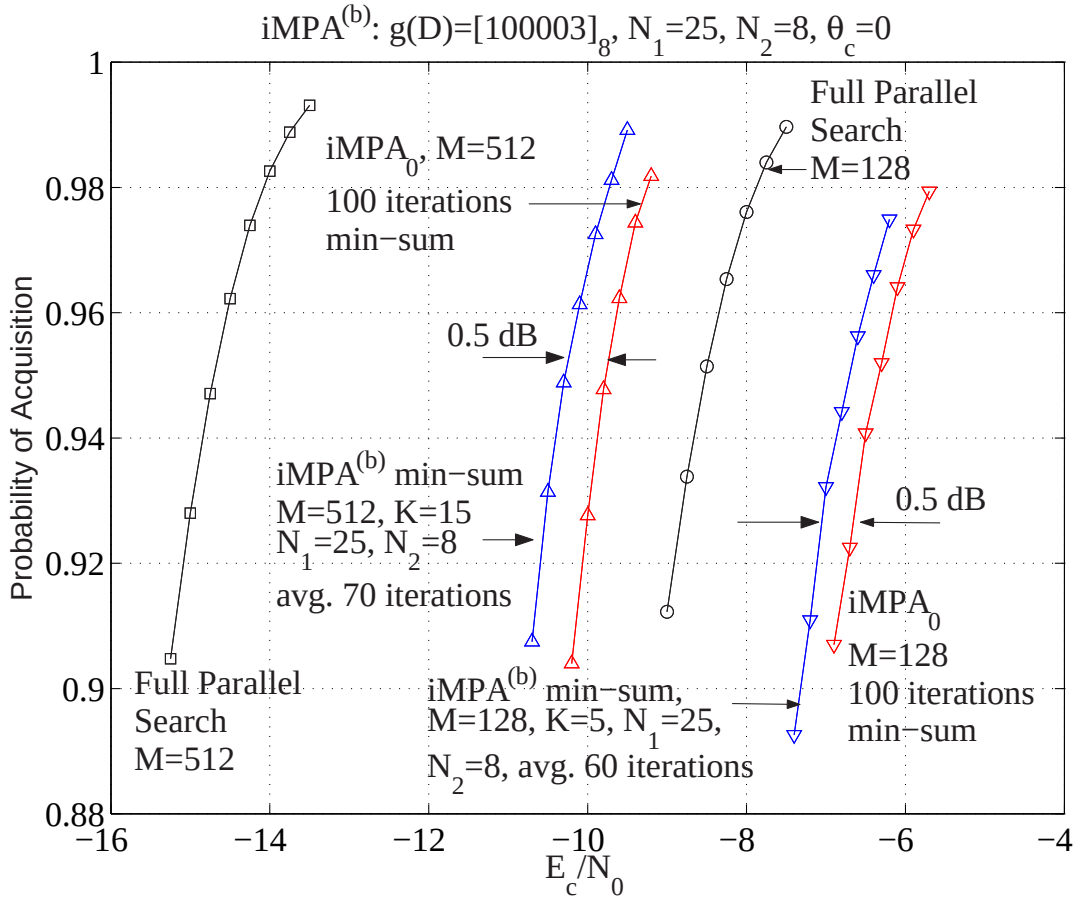

Figure 2.12: Improvement obtained by verification scheme for the UWB system with perfect frame synchronization and m-sequence generated by $g(D) = 1 + D + D^{15}$.

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. As presented in Figure 2.12, simulation results 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%, where the modify algorithm is denoted as iMPA$^{(b)}$.

To further improve performance, multiple time windows of size M can be combined together. Specifically, given M_2 non-overlapping 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. Since rapid acquisition is desired, a small M_2 is preferred. This defines a modified iMPA, denoted by iMPA$^{(v)}(I, V, S, M_2)$, where M_2 is the number of non-overlapping 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 iMPA$^{(v)}(I = 25, V = 8, S = 20, M_2 = 4)$ and $M = 512$ are shown in Figure 2.13. Compared to the iMPA$_0$ with $M = 512$ observations, this modified algorithm has a 3 dB performance gain. Also, using iMPA$^{(v)}$ to combine 4 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 $(E_c/N_0)_{req} = -14$ dB. Results from Figure 2.13 show that this can be achieved with an acquisition time of 2048 chip times using iMPA$^{(v)}(I = 25, V = 8, S = 20, M_2 = 4)$. Referring to Figure 2.5, simple serial search works at $(E_c/N_0)_{req}$, but requires $8.39 \cdot 10^6$ observations on average, which is substantially slower than the proposed approach.

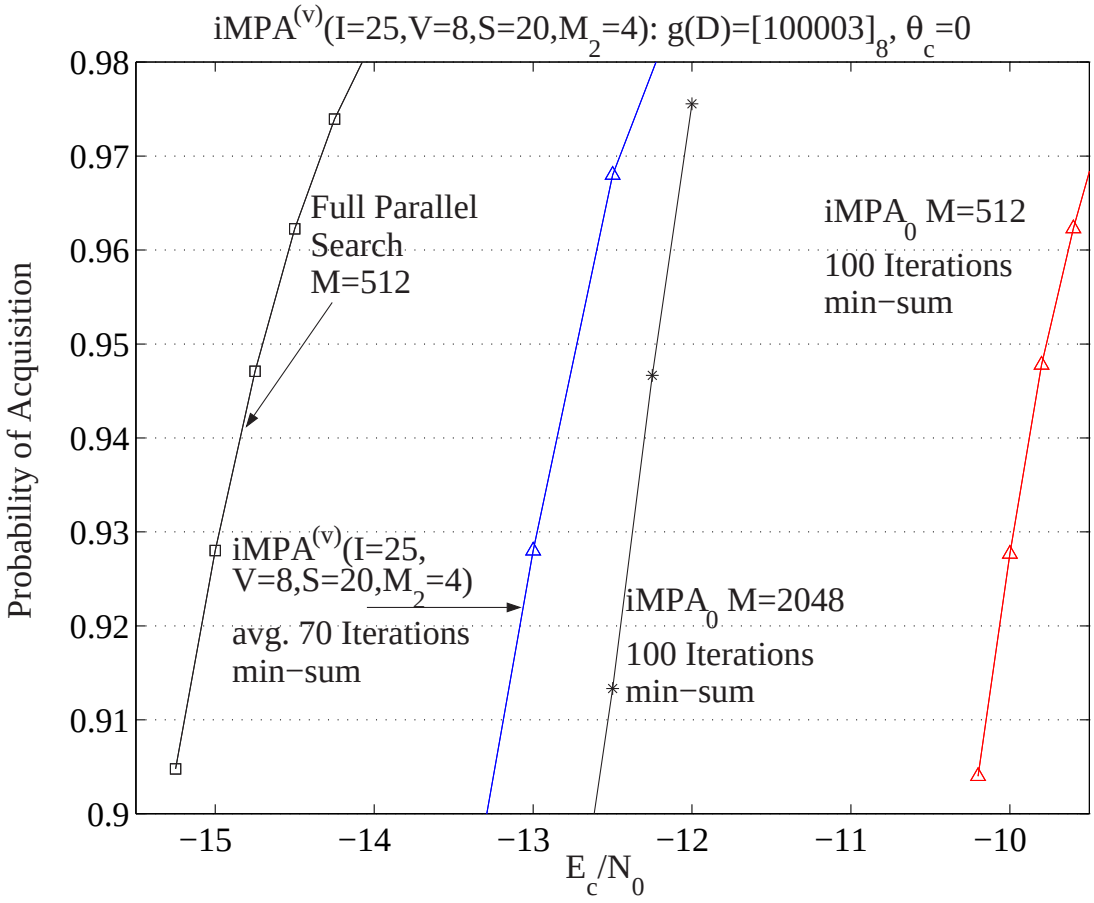

Figure 2.13: Improvement obtained by verification scheme to combine multiple windows of observations together. For the UWB system with perfect frame synchronization and m-sequence generated by $g(D) = 1 + D + D^{15}$.

2.5.4 Joint PN and frame epoch acquisition for the UWB system

As a final example we return to the UWB system in Figure 2.1(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 $\eta_{ES} < \eta$ is needed.

The frame epoch is estimated in a serial manner (*i.e.,*starting with $\widetilde{\xi} = 0$, then $\widetilde{\xi} = 1$, then $\widetilde{\xi} = 2$, etc.) until the correct frame epoch is detected. For a given hypothesized frame epoch, referring to the iMPA$^{(v)}$ in Section 2.5.3, if the best initial state estimate obtained has $v(\widehat{\mathbf{u}}) < \eta_{ES}$, the null-hypothesis is declared. Then, a new hypothesized frame epoch is considered and the iMPA$^{(v)}$ restarts with a new set of observations based on this hypothesized frame epoch.

Both the $C_p = 896$ hybrid search and 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 equation (2.1), 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 2.3, 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

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

Table 2.3: 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 2.5.4.

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.

2.6 Conclusion and future work

Iterative techniques are well known to be applicable in a wide range of problems, and in this chapter 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 post-processing that allows the results of the iMPA operating on sub-windows 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 non-cooperative 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 multi-path 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 pseudo-random sequences that are inherently generated by more random-like sparse loopy graphical models. In this chapter 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., 2-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 sub-graphs 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.

Chapter 3

Eigenvalue Analysis for Tanner graphs

3.1 Introduction

There are two well known facts that motivate the work in this chapter. First, eigenvalue analysis has been successfully used in spectral graph theory [16] for quite a long time to reveal several fundamental properties of graphs, such as the spectrum of the graph, connectivity and routing, diameter and girth, etc. The one of most interests is the connection between the eigenvalues and the expansion properties of the graphs. Second, to understand the behavior of iterative message passing algorithms on Tanner graphs, several researchers have suggested that iterative decoding would perform well if the underlying Tanner graphs had good expansion properties [12, 55]. For binary erasure channels, the concept of *stopping sets* [17, 37, 51] was introduced as the key parameter that determines the performance of message passing algorithms. Using our terminology, a stopping set is a subset of bit variables with vertex expansion no larger than $\frac{1}{2}$. Furthermore, though not precisely, similar arguments were used to analyze the iterative decoding on binary symmetric channels and white Gaussian noise channels.

In contrast to previous work where minimum expansions of subsets of variables are discussed for ensembles of linear codes, we will derive bounds on the average variable expansion for an arbitrary, specific Tanner graph. The reason that we want to introduce average expansion is that, for iterative decoding problems, not only is the expansion related to the performance, but also does the number of subsets with such expansion affect performance.

Therefore, after introducing the elements of graph representation and the associated incidence matrices in Section 3.2, we will demonstrate lower bounds on the vertex expansion of a given Tanner graph. The main technique used is to analyze the eigenvalues and correspondent eigenvectors of the *normalized incidence matrix* representing the graph. Section 3.4 contains summary of this chapter.

3.2 Graph Representations of Linear Codes

Considering a general bipartite graph G_T:

$$G_T = (X_n \cup Y_p, E) = (\{x_0, x_1, ..., x_{n-1}\} \cup \{y_0, y_1, ..., y_{p-1}\}, E) \tag{3.1}$$

where X_n and Y_p are the sets of vertices and $E = \{(x, y) : \ x \in X_n, \ y \in Y_p\}$ is the set of edges. This graph can be represented by a $p \times n$ incidence matrix $\mathbf{H}_p = [h_{ij}]$, the rows and columns of which correspond to Y_p and X_n respectively, such that $h_{ij} = 1$ if there is an edge between y_i and x_j, and $h_{ij} = 0$ otherwise, $0 \leq i \leq p-1$ and $0 \leq j \leq n-1$.

Let d_v denote the degree of vertex $v \in X_n \cup Y_p$, and let S denote a subset of $X_n \cup Y_p$, define

$$
\begin{aligned}
r_i &= \text{weight of row } i \text{ of } \mathbf{H}_p = d_{y_i} \quad 0 \le i \le p-1 && (3.2)\\
c_j &= \text{weight of column } j \text{ of } \mathbf{H}_p = d_{x_j} \quad 0 \le j \le n-1 && (3.3)\\
N(v) &= \text{the set of neighbors of } v = \{u : (v,u) \in E \ \text{ or } \ (u,v) \in E\} && (3.4)\\
N(S) &= \text{the set of neighbors of } S && (3.5)\\
\text{vol}(S) &= \text{the volume of } S = \sum_{v \in S} d_v && (3.6)
\end{aligned}
$$

Furthermore, we can define

$$
r_{\max} = \max_i r_i \quad r_{\min} = \min_i r_i \quad c_{\max} = \max_j c_j \quad c_{\min} = \min_j c_j \tag{3.7}
$$

and the $p \times n$ normalized incidence matrix :

$$
\mathbf{A}_p = [a_{ij}]_{p \times n} = \left[\frac{h_{ij}}{\sqrt{r_i \cdot c_j}} \right]_{p \times n} \tag{3.8}
$$

It can be shown that $\mathbf{A}_p^T \mathbf{A}_p$ and $\mathbf{A}_p \mathbf{A}_p^T$ share the same set of non-zero eigenvalues, among which the unique largest single eigenvalue is 1 [16]. Ordering the eigenvalues of $\mathbf{A}_p^T \mathbf{A}_p$ as $1 = \mu_0 > \mu_1 \ge \mu_2 ... \ge \mu_{p-1} > \mu_p = ... = \mu_{n-1} = 0$ if $p < n$ or $1 = \mu_0 > \mu_1 \ge \mu_2 ... \ge \mu_{n-1}$ otherwise, with corresponding orthonormal eigenvectors $\mathbf{e}_0, \mathbf{e}_1, ... \mathbf{e}_{n-1}$, it can also be shown that

$$
\mathbf{e}_0 = \frac{\mathbf{T}_d^{1/2} \mathbf{1}_n}{\sqrt{\text{vol}(G_T)}} \tag{3.9}
$$

where $\mathbf{T}_d = [t_{ij}]$ is a $n \times n$ diagonal matrix with $t_{jj} = c_j$, $0 \leq j \leq n-1$ and all entries of length-n column vector $\mathbf{1}_n$ are 1's. Similarly, let $\mathbf{e}'_0, \mathbf{e}'_1, ...\mathbf{e}'_{p-1}$ be the orthonormal eigenvectors of $\mathbf{A}_p\mathbf{A}_p^T$ corresponding to eigenvalues $1 = \mu_0 > \mu_1 \geq \mu_2 ... \geq \mu_{p-1}$, then,

$$\mathbf{e}'_0 = \frac{(\mathbf{T}'_d)^{1/2}\mathbf{1}_p}{\sqrt{\text{vol}(G_T)}} \tag{3.10}$$

where $\mathbf{T}'_d = [t'_{ij}]$ is a $p \times p$ diagonal matrix with $t'_{ii} = r_i$, $0 \leq i \leq p-1$. Now we are ready to present our results. However, it should be noted that this normalization technique has a long history and many applications in spectral graph theory. For more information about spectral graph theory, we direct the interested reader to [16].

Lemma 3.1. *For an arbitrary bipartite graph $G_T = (X_n \cup Y_p, E)$ and a subset S of X_n (or Y_p), we have*

$$\frac{\text{vol}(N(S))}{\text{vol}(S)} \geq \frac{1}{\mu_1 + (1-\mu_1)\frac{\text{vol}(S)}{\text{vol}(G_T)}} = \frac{\text{vol}(G_T)}{\mu_1 \text{vol}(G_T) + (1-\mu_1)\text{vol}(S)} \tag{3.11}$$

where μ_1 is the second largest eigenvalue of both $\mathbf{A}_p^T\mathbf{A}_p$ and $\mathbf{A}_p\mathbf{A}_p^T$ [1].

Proof of Lemma 3.1. *Considering $S \subseteq X_n$, define a $n \times 1$ column vector ψ_S as $(\psi_0, \psi_1, ...\psi_{n-1})^T$, where $\psi_j = 1$, if $x_j \in S$ and $\psi_j = 0$, otherwise. Expressing $\mathbf{T}_d^{1/2}\psi_S$ as a linear combination of the orthonormal eigenvectors of $\mathbf{A}_p^T\mathbf{A}_p$,*

$$\mathbf{T}_d^{1/2}\psi_S = \sum_{j=0}^{n-1} \langle \mathbf{T}_d^{1/2}\psi_S, \mathbf{e}_j \rangle \mathbf{e}_j = \sum_{j=0}^{n-1} a_j \mathbf{e}_j \tag{3.12}$$

[1] Similar results can be found in [16] for the graphs of regular row/column weights. However, extensions to the irregular case discussed in [16] are not fully developed and draw invalid conclusions. The proof of Lemma 3.1 is based on similar techniques and can be considered an extension of Chung's work.

and

$$a_0 = \langle \mathbf{T}_d^{1/2}\psi_S, \mathbf{e}_0\rangle = \frac{\psi_S^T \mathbf{T}_d \mathbf{1}_n}{\sqrt{\mathrm{vol}(G_T)}} = \frac{\mathrm{vol}(S)}{\sqrt{\mathrm{vol}(G_T)}} \tag{3.13}$$

$$\sum_{j=0}^{n-1} a_j^2 = \langle \mathbf{T}_d^{1/2}\psi_S, \mathbf{T}_d^{1/2}\psi_S\rangle = \psi_S^T \mathbf{T}_d \psi_S = \mathrm{vol}(S) \tag{3.14}$$

where $\langle\cdot,\cdot\rangle$ *denotes the inner product [56] of two column vectors, then*

$$\langle \mathbf{A}_p\mathbf{T}_d^{1/2}\psi_S, \mathbf{A}_p\mathbf{T}_d^{1/2}\psi_S\rangle = \psi_S^T\mathbf{T}_d^{1/2}\mathbf{A}_p^T\mathbf{A}_p\mathbf{T}_d^{1/2}\psi_S \tag{3.15a}$$

$$= \sum_{j=0}^{n-1} a_j^2\mu_j \tag{3.15b}$$

$$\leq a_0^2 + (\sum_{j=1}^{n-1} a_j^2)\mu_1 \tag{3.15c}$$

$$= \frac{(\mathrm{vol}(S))^2}{\mathrm{vol}(G_T)} + \left(\mathrm{vol}(S) - \frac{(\mathrm{vol}(S))^2}{\mathrm{vol}(G_T)}\right)\mu_1 \tag{3.15d}$$

$$= (1-\mu_1)\frac{(\mathrm{vol}(S))^2}{\mathrm{vol}(G_T)} + \mu_1\mathrm{vol}(S) \tag{3.15e}$$

Furthermore,

$$\langle \mathbf{A}_p\mathbf{T}_d^{1/2}\psi_S, \mathbf{A}_p\mathbf{T}_d^{1/2}\psi_S\rangle = \sum_{u\in S}\sum_{v\in S} \sum_{\substack{y\,:\,(v,y)\,\in\,E \\ \text{and } (u,y)\,\in\,E}} \frac{1}{d_y} \tag{3.16a}$$

$$= \sum_{y\in N(S)} \left|\frac{N(y)\cap S}{\sqrt{d_y}}\right|^2 \tag{3.16b}$$

$$\geq \frac{\left(\sum_{y\in N(S)} \frac{|N(y)\cap S|}{\sqrt{d_y}}\sqrt{d_y}\right)^2}{\sum_{y\in N(S)} d_y} \tag{3.16c}$$

$$= \frac{(\mathrm{vol}(S))^2}{\mathrm{vol}(N(S))} \tag{3.16d}$$

where (3.16a) *and* (3.16b) *are generalized from [16, Page 97] and* (3.16c) *results from Cauchy-Schwartz inequality. Combining* (3.15e) *and* (3.16d),

$$(1-\mu_1)\frac{(\mathrm{vol}(S))^2}{\mathrm{vol}(G_T)}+\mu_1\mathrm{vol}(S)\geq\langle\mathbf{A}_p\mathbf{T}_d^{1/2}\psi_S,\mathbf{A}_p\mathbf{T}_d^{1/2}\psi_S\rangle\geq\frac{(\mathrm{vol}(S))^2}{\mathrm{vol}(N(S))} \tag{3.17}$$

and (3.11) *is the direct result. Similarly, we can prove* (3.11) *for* $S\subseteq Y_p$ *by using* $\mathbf{e}_i'$*'s and* $\mathbf{T}_d'$ *defined at* (3.10)*, and* $\psi_S'=(\psi_0',\psi_1',...\psi_{p-1}')^T$*, where* $\psi_i'=1$*, if* $y_i\in S$ *and* $\psi_i'=0$ *otherwise.* □

3.3 Lower Bounds on Variable Expansions of Tanner Graphs

3.3.1 Definitions

For a given bipartite graph $G_T=(X_n\cup Y_p,E)$, considering a subset $S_m\subseteq X_n$ (or Y_n) with size m, *i.e.,* $|S_m|=m$, its expansion is defined as:

$$\delta(S_m)=\frac{\text{the number of neighbors of } S_m}{\text{volume of } S_m}=\frac{|N(S_m)|}{\mathrm{vol}(S_m)} \tag{3.18}$$

It should be noted that, strictly speaking, what is defined in (3.18) is the"vertex" expansion. There is also an "edge" expansion defined in the literature of spectral graph theory. However, as we only consider vertex expansion throughout our work, we use the term expansion to refer to vertex expansion. It should also be noted that, to analyze the performance of iterative decoding, typically only subsets of variables, *i.e.,* subsets of X_n, are considered. Therefore, we only consider $S_m\subseteq X_n$ hereafter. However, the definition of expansion is not restrict to variables.

Furthermore, we can define:

$$\delta_{\min}(m) \quad = \min_{S_m} \delta(S_m) \tag{3.19}$$

$$\delta_{\max}(m) \quad = \max_{S_m} \delta(S_m) \tag{3.20}$$

$$\delta_{\text{avg}}(m) \quad = \frac{1}{\begin{pmatrix} n \\ m \end{pmatrix}} \sum_{S_m} \delta(S_m) \tag{3.21}$$

where $\begin{pmatrix} n \\ m \end{pmatrix}$ is the binomial coefficient.

3.3.2 Relations to previous results

In [17, 37, 51], stopping sets were used to determine the performance of iterative decoding on erasure channels. For $S_m \subseteq X_n$, we say that S_m is a stopping set if all vertices in the neighborhood of S_m, *i.e.,* vertices in $N(S_m)$, are connected to at least two different vertices in S_m. Thus, if S_m is a stopping set, $\delta(S_m) < \frac{1}{2}$. It should be noted that $\delta(S_m) < \frac{1}{2}$ is a necessary but not sufficient condition for S_m to be a stopping set.

Furthermore, stopping distance [17] was defined as the size of the smallest stopping sets. Let

$$m_\delta = \text{the smallest } m \text{ such that } \delta_{\min}(m) \geq \frac{1}{2} \tag{3.22}$$

then

$$\text{stopping distance } \geq m_\delta \tag{3.23}$$

In [55], the authors discussed iterative decoding of (d_v, d_c)-regular LDPC codes on binary symmetric channels, where all variables vertices have degree d_v and all parity-checks have degree d_c. They proved that Spielman's simple sequential decoding algorithm can correct any $\alpha n/2$ or fewer random errors if every variable subset of the size αn or less expands by a factor of at least $3d_v/4$, where n is the number of variable vertices. Translating into our notation, it is equivalent to say that Spielman's simple sequential decoding algorithm can correct any pattern of $m/2$ or fewer errors if $\delta_{\min}(i) \geq 3/4$ for $1 \leq i \leq m$. Similar results were obtained in [31, 32], where irregular LDPC codes were discussed, and [12], where Gallager's hard-decision decoding and soft-decision decoding (with clipping) algorithms were discussed. Thus, our goal in this chapter is to establish lower bounds on expansion properties for a given Tanner graph with these results in mind.

3.3.3 Lower bounds on expansion properties of variable subsets

Using Lemma 3.1, we now present lower bounds on expansion parameters defined in (3.18) and (3.21).

Theorem 3.1. *For any subset S_m of X_n,*

$$\delta(S_m) \geq \frac{1}{r_{\max}} \cdot \frac{\mathrm{vol}(G_T)}{\mu_1 \mathrm{vol}(G_T) + (1-\mu_1)\mathrm{vol}(S_m)} \tag{3.24}$$

where μ_1 is the second largest eigenvalue of $\mathbf{A}_p \mathbf{A}_p^T$.

Proof of Theorem 3.1. *Follows directly from Lemma 3.1 and* (3.18)*, using the fact that $N(S_m) \geq \mathrm{vol}(N(S_m))/r_{\max}$.* □

Theorem 3.2. *For subsets of X_n with size m,*

$$\delta_{\text{avg}}(m) \geq \frac{1}{r_{\max}} \frac{n}{m + (n-m)\mu_1} \tag{3.25}$$

where μ_1 is the second largest eigenvalue of $\mathbf{A}_p \mathbf{A}_p^T$.

Proof of Theorem 3.2. *To prove Theorem 3.2, we need to go back to* (3.17).

$$(1-\mu_1)\frac{(\text{vol}(S_m))^2}{\text{vol}(G_T)} + \mu_1 \text{vol}(S_m) \geq \frac{(\text{vol}(S_m))^2}{\text{vol}(N(S_m))} \tag{3.26a}$$

$$\Leftrightarrow (1-\mu_1)\frac{\text{vol}(S_m)}{\text{vol}(G_T)} + \mu_1 \geq \frac{\text{vol}(S_m)}{\text{vol}(N(S_m))} \geq \frac{\text{vol}(S_m)}{|N(S_m)| r_{\max}} = \frac{1}{\delta(S_m) r_{\max}} \tag{3.26b}$$

Summing both sides of the (3.26b) *over all $S_m \subseteq X_n$ such that $|S_m| = m$, and noting the fact that $\sum_{S_m} \text{vol}(S_m) = \begin{pmatrix} n-1 \\ m-1 \end{pmatrix} \text{vol}(G_T)$ and $\sum_{S_m} 1 = \begin{pmatrix} n \\ m \end{pmatrix}$,*

$$\sum_{S_m} \mu_1 + (1-\mu_1) \sum_{S_m} \frac{\text{vol}(S_m)}{\text{vol}(G_T)} \geq \sum_{S_m} \frac{1}{\delta(S_m) r_{\max}} \tag{3.27a}$$

$$\Leftrightarrow \begin{pmatrix} n \\ m \end{pmatrix} \mu_1 + \begin{pmatrix} n-1 \\ m-1 \end{pmatrix} (1-\mu_1) \geq \frac{1}{r_{\max}} \sum_{S_m} \frac{1}{\delta(S_m)} \tag{3.27b}$$

Also, using Cauchy-Schwartz inequality, the right side of (3.27b) *satisfies*

$$\sum_{S_m} \frac{1}{\delta(S_m)} = \sum_{S_m} \frac{1}{\delta(S_m)} \frac{\sum_{S_m} \delta(S_m)}{\sum_{S_m} \delta(S_m)} = \sum_{S_m} \frac{1}{\delta(S_m)} \frac{\sum_{S_m} \delta(S_m)}{\begin{pmatrix} n \\ m \end{pmatrix} \delta_{\text{avg}}(m)} \geq \begin{pmatrix} n \\ m \end{pmatrix} \frac{1}{\delta_{\text{avg}}(m)}$$

Therefore,

$$\begin{pmatrix} n \\ m \end{pmatrix} \mu_1 + \begin{pmatrix} n-1 \\ m-1 \end{pmatrix} (1-\mu_1) \geq \frac{1}{r_{\max}} \begin{pmatrix} n \\ m \end{pmatrix} \frac{1}{\delta_{\text{avg}}(m)} \tag{3.28}$$

and (3.25) *follows.* □

However, Theorem 3.2 usually provides relatively weak lower bound on the average expansion of variable subsets because only the second largest eigenvalue of $\mathbf{A}_p\mathbf{A}_p^T$ is used. This can be observed from Figure 3.1. In the following theorem, another lower bound on $\delta_{\text{avg}}(m)$ is derived where all the eigenvalues and corresponding eigenvectors are used.

Theorem 3.3. *Considering a Tanner Graph* $G_T = (X_n \bigcup Y_p, E)$ *with largest variable degree of* L *and* $|X_n| = n$*, let* n_l *be the number of variable nodes of degree* l*,* $1 \leq l \leq L$*, and* d_l *be an integer such that* $0 \leq d_l \leq n_l$*, then*

$$\delta_{\text{avg}}(m) \geq \frac{\left(\underbrace{\sum^{n_1} .. \sum^{n_L}}_{d_1+..+d_L=m} \begin{pmatrix} n_1 \\ d_1 \end{pmatrix} ... \begin{pmatrix} n_L \\ d_L \end{pmatrix} \sqrt{\sum_l l d_l} \right)^2}{r_{\max} \begin{pmatrix} n \\ m \end{pmatrix} \sum_{j=0}^{n-1} \mu_j \widetilde{a}_j^2} \tag{3.29}$$

where μ_j's and $\mathbf{e}_j$'s are eigenvalues and the corresponding eigenvectors of $\mathbf{A}_p\mathbf{A}_p^T$ and

$$\widetilde{a}_j^2 = \begin{cases} \left[\begin{pmatrix} n-1 \\ \\ m-1 \end{pmatrix} - \begin{pmatrix} n-2 \\ \\ m-2 \end{pmatrix}\right] \mathbf{e}_0^T \mathbf{T}_d \mathbf{e}_0 + \begin{pmatrix} n-2 \\ \\ m-2 \end{pmatrix} \mathrm{vol}(G_T) & \text{if j} = 0, \\ \left[\begin{pmatrix} n-1 \\ \\ m-1 \end{pmatrix} - \begin{pmatrix} n-2 \\ \\ m-2 \end{pmatrix}\right] \mathbf{e}_j^T \mathbf{T}_d \mathbf{e}_j & \text{otherwise.} \end{cases} \tag{3.30}$$

Proof of Theorem 3.3. *Let $S_m \subseteq X_n$, $|S_m| = m$ and define a $n \times 1$ column vector ψ_{S_m} as $(\psi_0, \psi_1, ...\psi_{n-1})^T$, where $\psi_j = 1$, if $x_j \in S_m$ and $\psi_j = 0$, otherwise. Let*

$$a_j(S_m) = \langle \mathbf{T}_d^{1/2} \psi_{S_m}, \mathbf{e}_j \rangle \tag{3.31a}$$

$$= \psi_{S_m}^T \left(\mathbf{T}_d^{1/2}\right)^T \mathbf{e}_j \tag{3.31b}$$

$$= \psi_{S_m}^T \mathbf{T}_d^{1/2} \mathbf{e}_j \tag{3.31c}$$

$$= \mathbf{e}_j^T \mathbf{T}_d^{1/2} \psi_{S_m} \tag{3.31d}$$

combining (3.15b) *and* (3.16d),

$$\sum_{j=0}^{n-1} \left(a_j(S_m)\right)^2 \mu_j \geq \frac{(\mathrm{vol}(S_m))^2}{\mathrm{vol}(N(S_m))} \tag{3.32}$$

Summing the left side of the (3.32) *over all* $S_m \subseteq X_n$ *such that* $|S_m| = m$,

$$\sum_{S_m} \sum_{j=0}^{n-1} \left(a_j(S_m)\right)^2 \mu_j = \sum_{j=0}^{n-1} \mu_j \sum_{S_m} \left(a_j(S_m)\right)^2 \tag{3.33a}$$

$$= \sum_{j=0}^{n-1} \mu_j \sum_{S_m} |\langle \mathbf{T}_d^{1/2} \psi_{S_m}, \mathbf{e}_j \rangle|^2 \tag{3.33b}$$

$$= \sum_{j=0}^{n-1} \mu_j \sum_{S_m} \mathbf{e}_j^T \mathbf{T}_d^{1/2} \psi_{S_m} \psi_{S_m}^T \mathbf{T}_d^{1/2} \mathbf{e}_j \tag{3.33c}$$

$$= \sum_{j=0}^{n-1} \mu_j \mathbf{e}_j^T \mathbf{T}_d^{1/2} \left(\sum_{S_m} \psi_{S_m} \psi_{S_m}^T \right) \mathbf{T}_d^{1/2} \mathbf{e}_j \tag{3.33d}$$

$$= \sum_{j=0}^{n-1} \mu_j \widetilde{a}_j^2 \tag{3.33e}$$

From the definition of ψ_{S_m}, *it can be shown that* $\psi_{S_m} \psi_{S_m}^T$ *is a* $n \times n$ *binary symmetric matrix, where the entry at the intersection of the* i*-th row and the* j*-th column is* 1 *if and only if both* x_i *and* x_j *are in* S_m. *Then, it can be shown that,*

$$\sum_{S_m} \psi_{S_m} \psi_{S_m}^T = \begin{bmatrix} \binom{n-1}{m-1} & \binom{n-2}{m-2} & \cdots & \binom{n-2}{m-2} \\ \binom{n-2}{m-2} & \binom{n-1}{m-1} & \cdots & \binom{n-2}{m-2} \\ \vdots & \vdots & \ddots & \vdots \\ \binom{n-2}{m-2} & \binom{n-2}{m-2} & \cdots & \binom{n-1}{m-1} \end{bmatrix}$$

$$= \left[\binom{n-1}{m-1} - \binom{n-2}{m-2} \right] \mathbf{I}_n + \binom{n-2}{m-2} \mathbf{1}_n \mathbf{1}_n^T \tag{3.34}$$

where $\mathbf{I}_n$ *is the* $n \times n$ *identity matrix and* $\mathbf{1}_n$ *is the* $n \times 1$ *all one column vector. Noting that* $\mathbf{e}_0 = \frac{\mathbf{T}_d^{1/2}\mathbf{1}_n}{\sqrt{\text{vol}(G_T)}}$, i.e., (3.9), *and* $\mathbf{e}_j$, $1 \leq j \leq n-1$, *are the orthonormal eigenvectors of* $\mathbf{A}_p\mathbf{A}_p^T$,

$$\widetilde{a}_j^2 = \mathbf{e}_j^T \mathbf{T}_d^{1/2} \left(\sum_{S_m} \psi_{S_m} \psi_{S_m}^T \right) \mathbf{T}_d^{1/2} \mathbf{e}_j \tag{3.35a}$$

$$= \left[\begin{pmatrix} n-1 \\ m-1 \end{pmatrix} - \begin{pmatrix} n-2 \\ m-2 \end{pmatrix} \right] \mathbf{e}_j^T \mathbf{T}_d \mathbf{e}_j + \begin{pmatrix} n-2 \\ m-2 \end{pmatrix} \mathbf{e}_j^T \mathbf{T}_d^{1/2} \mathbf{1}_n \mathbf{1}_n^T \mathbf{T}_d^{1/2} \mathbf{e}_j \tag{3.35b}$$

$$= \left[\begin{pmatrix} n-1 \\ m-1 \end{pmatrix} - \begin{pmatrix} n-2 \\ m-2 \end{pmatrix} \right] \mathbf{e}_j^T \mathbf{T}_d \mathbf{e}_j + \begin{pmatrix} n-2 \\ m-2 \end{pmatrix} \text{vol}(G_T) \mathbf{e}_j^T \mathbf{e}_0 \mathbf{e}_0^T \mathbf{e}_j \tag{3.35c}$$

$$= \begin{cases} \left[\begin{pmatrix} n-1 \\ m-1 \end{pmatrix} - \begin{pmatrix} n-2 \\ m-2 \end{pmatrix} \right] \mathbf{e}_0^T \mathbf{T}_d \mathbf{e}_0 + \begin{pmatrix} n-2 \\ m-2 \end{pmatrix} \text{vol}(G_T) & \text{if } \mathrm{j} = 0, \\ \left[\begin{pmatrix} n-1 \\ m-1 \end{pmatrix} - \begin{pmatrix} n-2 \\ m-2 \end{pmatrix} \right] \mathbf{e}_j^T \mathbf{T}_d \mathbf{e}_j & \text{otherwise.} \end{cases} \tag{3.35d}$$

On the other hand, if summing the right side of the (3.32) *over all* $S_m \subseteq X_n$,

$$\sum_{S_m} \frac{(\mathrm{vol}(S_m))^2}{\mathrm{vol}(N(S_m))} \geq \sum_{S_m} \frac{(\mathrm{vol}(S_m))^2}{|N(S_m)| r_{\max}} \tag{3.36a}$$

$$= \frac{1}{r_{\max}} \sum_{S_m} \frac{\mathrm{vol}(S_m)}{\delta(S_m)} \tag{3.36b}$$

$$= \frac{1}{r_{\max}} \sum_{S_m} \frac{\mathrm{vol}(S_m)}{\delta(S_m)} \frac{\sum_{S_m} \delta(S_m)}{\begin{pmatrix} n \\ m \end{pmatrix} \delta_{\mathrm{avg}}(m)} \tag{3.36c}$$

$$\geq \frac{1}{r_{\max}} \frac{1}{\begin{pmatrix} n \\ m \end{pmatrix} \delta_{\mathrm{avg}}(m)} (\sum_{S_m} \sqrt{\frac{\mathrm{vol}(S_m)}{\delta(S_m)}} \sqrt{\delta(S_m)})^2 \tag{3.36d}$$

$$= \frac{1}{r_{\max}} \frac{1}{\begin{pmatrix} n \\ m \end{pmatrix} \delta_{\mathrm{avg}}(m)} (\sum_{S_m} \sqrt{\mathrm{vol}(S_m)})^2 \tag{3.36e}$$

Combining (3.35d) *and* (3.36e), *we have*

$$\delta_{\mathrm{avg}}(m) \geq \frac{(\sum_{S_m} \sqrt{\mathrm{vol}(S_m)})^2}{r_{\max} \begin{pmatrix} n \\ m \end{pmatrix} \sum_{j=0}^{n-1} \mu_j \widetilde{a}_j^2} \tag{3.37}$$

As the final step, using d_l *and* n_l, *it is easy to show that*

$$\sum_{S_m} \sqrt{\mathrm{vol}(S_m)} = \underbrace{\sum_{}^{n_1} .. \sum_{}^{n_L}}_{d_1 + .. + d_L = m} \begin{pmatrix} n_1 \\ d_1 \end{pmatrix} ... \begin{pmatrix} n_L \\ d_L \end{pmatrix} \sqrt{\sum_l l d_l} \tag{3.38}$$

and (3.29) *follows.* □

3.3.4 Example application of the bounds

Considering $[15, 7, 5]$ cyclic BCH code [30, Ch. 6] as an example, it has cyclic parity-check matrix

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.39}$$

and bounds provided in Theorem 3.2 and Theorem 3.3 are evaluated, as shown in Figure 3.1. It can be seen that Theorem 3.3 provides a much better bound than Theorem 3.2. Though calculating (3.29) needs more computation, it is much less complex than calculating $\delta_{\text{avg}}(m)$, which is exponential in n. This exact computation is also shown in Figure 3.1.

3.4 Summary

In this chapter, using some techniques well-developed in spectral graph theory, we derived lower bounds on the expansion properties of subsets of variables. Specifically, for any given Tanner graph represented by an incidence matrix, we showed that the expansion

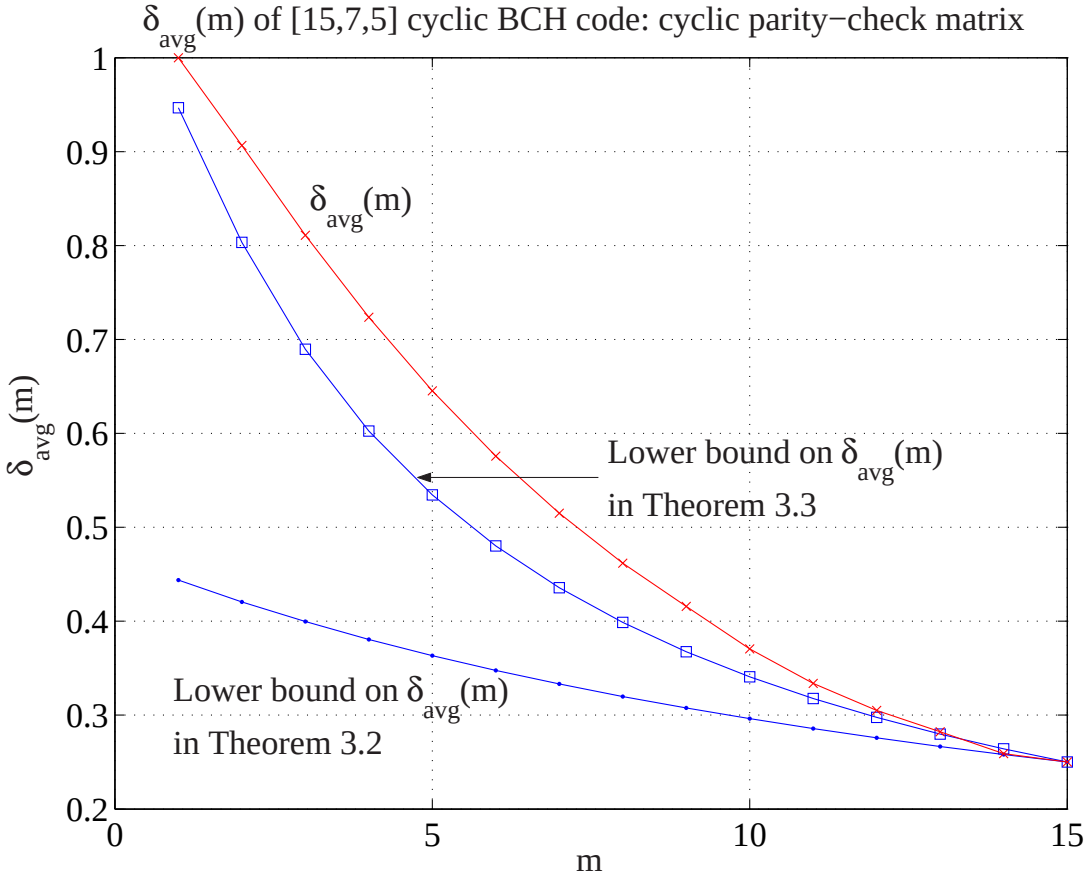

Figure 3.1: Lower bounds on $\delta_{\text{avg}}(m)$ and $\delta_{\text{avg}}(m)$ as functions of m: [15,7,5] cyclic BCH code

of subsets of variables, and the average expansion over all subsets of variables of the same size m, can be lower bounded by functions of the eigenvalues and corresponding eigenvectors of the normalized incidence matrix. These theoretical results have been verified using one simple example. However, as there are more advanced methods in the literature and there is still noticeable gap between our best bound and the actual value of the average expansion, it is an interesting question whether the expansion properties can be bounded by other quantities.

Also, results in this chapter will be used in following chapters to discuss topics related to the performance of iterative decoding on various channels, because the connection between the expansion property of subsets of variables and the performance of iterative decoding is generally believed.

Chapter 4

Bounds on Stopping Distance and Stopping Redundancy

4.1 Introduction

One of the most important and interesting open problems to the communication society is to determine the performance of iterative message passing algorithms on loopy graphs. Some recent work addresses this problem for loopy Tanner graphs on erasure channels [18].

Luby and his colleagues viewed the iterative decoding over Binary Erasure Channel (BEC) as a discrete random process and modelled its evolution by a system of differential equations [31, 32]. Using the assertion that the related random variables do not deviate too much from their expected value over the whole period of decoding, they explicitly solved the differential equations. Using these results, efficient encoding and decoding of capacity-approaching codes were also discussed.

Luby's results were also verified by Richardson and Urbanke using a technique called *density evolution* under the assumption that messages passed along different edges are independent [47]. Along with Schokrollahi, they also proposed some search strategies and provided search results of degree distributions of irregular LDPC codes, capacities of

which are very close to the Shannon bound [46]. It should be noted that the application of Richardson's work is not limited to erasure channels and details related to other channels will be discussed in the following chapter.

The reason that many researchers are interested in the BEC is that the performance of iterative decoding on Tanner graphs on the BEC is completely determined by its stopping sets [17]. The size of the smallest stopping sets was defined as *stopping distance* [37, 51]. Focusing on Tanner graph ensembles, Orlitsky, Viswanathan and Zhang [37] derived several results on the asymptotic behavior of stopping sets. Specifically, they have demonstrated a relation between degree distribution of the Tanner graph ensembles and the likely size of the smallest stopping sets, and argued that the size of these stopping sets is linear in the block length if certain conditions are satisfied by the degree distribution. They have also provided bounds on the average block error probability as a function of the erasure probability of the channel. On the other hand, Pishro-Nik and Fekri [43] used finite-length analysis to derive bounds on the maximum-likelihood (ML) capacity on BEC and presented an improved iterative decoding algorithm.

Generally, it is believed that by adding extra check nodes to the graphical representation, the performance of the iterative MPAs may be improved. Though various simulations [75] suggest that it is a plausible conjecture, it has not been proven in general. Most recent work by Schwartz and Vardy [51] solved this problem for loopy Tanner graphs on the BEC. They have demonstrated that, by carefully adding extra parity-checks, the stopping distance of the Tanner graph can be increased, which will improve the performance of the iMPAs correspondingly. They define the *stopping redundancy* as

the smallest number of parity-checks, which are needed so that the stopping distance of a Tanner graph representation of the code equals the minimum distance of the code.

Previous investigations have considered the properties of an ensemble of linear codes. In contrast, we focus on the parameters of an arbitrary linear code and will analyze eigenvalues and eigenvectors of the *"normalized" incidence matrix* representing the code. Using this technique, we will derive two lower bounds on the stopping distance. Since the stopping distance is always no larger than d_{min}, the minimum distance of the code, these lower bounds are also lower bounds on the minimum distance. In particular, if the graph is regular, they are Tanner's bit-oriented bound and parity-oriented bound [60] respectively, *i.e.,* we demonstrate that Tanner's bounds are actually lower bounds on stopping distance instead of d_{min} of regular Tanner graphs.

Furthermore, this technique can also be used to derive bounds on stopping redundancy, which was denoted as $\rho(\mathcal{C})$ in [51]. Previously, Schwartz and Vardy [51] proved that stopping redundancy is well defined and provided bounds on $\rho(\mathcal{C})$ for the family of binary Reed-Muller codes, extended Golay Codes and maximum distance separable (MDS) codes. In this work, we will provide an upper bound on $\rho(\mathcal{C})$ for the family of *simple difference-set codes*, *i.e.,*$\rho(\mathcal{C}) \leq n$, where n is the length of the code. It has been noted that similar results were obtained by Vontobel, Smarandache, Kiyacash, Teutsch and Vukobratovic [67], where minimal codewords and minimum pseudo-codewords of the families of codes derived from finite geometries were studied.

Using the graph analysis techniques discussed in Chapter 3 lower bounds on stopping distance for linear codes are derived in Section 4.2, which will also lead to Tanner's bit-oriented bound and parity-oriented bound on d_{min} for regular LDPC [22] codes. We

continue in Section 4.3 to show connections between our work and the work of Schwartz and Vardy by providing upper bounds on stopping redundancy of the difference-set codes. After a short discussion of possible directions for future research 4.4, conclusions are drawn in Section 4.5.

4.2 Lower Bounds of Stopping Distance

Considering an $[n, k, d_{\min}]$ binary linear code $\mathcal{C}$ specified by a $p \times n$ *incidence matrix* $\mathbf{H}_p$ with columns representing *bit variables*, rows representing *parity-checks* and $p \geq n - k = p_0$, the corresponding Tanner graph [59] G_T is:

$$G_T = (X_n \cup Y_p, E) = (\{x_0, x_1, ..., x_{n-1}\} \cup \{y_0, y_1, ..., y_{p-1}\}, E) \tag{4.1}$$

where X_n is the set of variables, Y_p is the set of single parity-check constraints and $E = \{(x, y) : \ x \in X_n, \ y \in Y_p\}$ is the set of edges. It can be shown that the correspondence between $\mathbf{H}_p$ and the traditional *parity-check matrix* representing $\mathcal{C}$ is one-to-one. When $p = p_0$, $\mathbf{H}_p$ is the standard parity-check matrix for the code. For $p \geq p_0$, there are redundant parity-checks and we refer to $\mathbf{H}_p$ as a redundant parity-check matrix. In either case, $\mathbf{H}_p$ can be interpreted as both a parity-check matrix and the incidence matrix for the corresponding bipartite graph.

Considering $S \subseteq X_n$, define bit variables in S as *active bits* and parity-checks in the neighborhood of S as *active parity-checks* [60], respectively, and S is called a *stopping set* if all the neighbors of S, *i.e.,* all active parity-checks, are connected to S at least twice. It is known that, for BEC, the performance of iterative decoding on G_T is completely

determined by its stopping sets [17]. The size of the smallest stopping set was defined as the *stopping distance* [51], which is usually denoted as $s(\mathbf{H}_p)$ to emphasize that it is a function of the specific (redundant) parity-check matrix representing the code.

Using Lemma 3.1,we will derive lower bounds on the stopping distance of linear codes. Since $s(\mathbf{H}_p) \leq d_{\min}$, these lower bounds are also lower bounds on $d_{\min}$. In particular, they lead to Tanner's results [60] when the underlying Tanner graph is regular.

Theorem 4.1. *For the $[n, k, d_{\min}]$ linear code $\mathcal{C}$ defined by the Tanner graph $G_T = (X_n \cup Y_p, E)$ with $p \times n$ incidence matrix $\mathbf{H}_p$, define $c_{\max}$, $c_{\min}$ and $r_{\max}$ as in (3.7), the followings are true:*

$$d_{\min} \geq s(\mathbf{H}_p) \geq \frac{(2/r_{\max}) - \mu_1}{1 - \mu_1} \cdot \frac{\mathrm{vol}(G_T)}{c_{\max}} = B(\mathbf{H}_p) \tag{4.2}$$

$$d_{\min} \geq s(\mathbf{H}_p) \geq \frac{1 + (2c_{\min} - 2)/r_{\max} - \mu_1 c_{\max}}{(1 - \mu_1)c_{\max}} \cdot \frac{2\mathrm{vol}(G_T)}{c_{\max} r_{\max}} = P(\mathbf{H}_p) \tag{4.3}$$

where $\mathrm{vol}(G_T)$ is the volume of G_T as defined in (3.6), $B(\mathbf{H}_p)$ and $P(\mathbf{H}_p)$ denote the bit-oriented bound and parity-oriented bound obtained using $\mathbf{H}_p$, respectively, μ_1 is the second largest eigenvalue of $\mathbf{A}_p^T \mathbf{A}_p$ and $\mathbf{A}_p$ as defined in (3.8).

Using Tanner's terminology, we call (4.2) and (4.3) bit-oriented bound and parity-oriented bound, respectively. The bit-oriented bound, *i.e.,* (4.2), becomes meaningless if $\mu_1 > 2/r_{\max}$. However, $1 + (2c_{\min} - 2)/r_{\max} - \mu_1 c_{\max}$ may still be positive which makes parity-oriented bound meaningful.

Proof of Theorem 4.1. *Since stopping distance is always no larger than minimum distance [51], we only need to prove the second inequalities of (4.2) and (4.3).*

Let $S_1 \subseteq X_n$ be a smallest stopping set, $N(S_1)$ is then the set of active parity-checks and $s(\mathbf{H}_p) = |S_1|$. Applying Lemma 3.1,

$$\frac{|N(S_1)|r_{\max}}{\mathrm{vol}(S_1)} \geq \frac{\mathrm{vol}(N(S_1))}{\mathrm{vol}(S_1)} \geq \frac{\mathrm{vol}(G_T)}{\mu_1 \mathrm{vol}(G_T) + (1-\mu_1)\mathrm{vol}(S_1)} \tag{4.4}$$

where μ_1 is the second largest eigenvalue of $\mathbf{A}_p^T \mathbf{A}_p$. Since any active parity-check in $N(S_1)$ must be connected to at least two active bits, $|N(S_1)| \leq \frac{1}{2}\mathrm{vol}(S_1)$. Therefore,

$$\frac{r_{\max}}{2} \geq \frac{\mathrm{vol}(G_T)}{\mu_1 \mathrm{vol}(G_T) + (1-\mu_1)\mathrm{vol}(S_1)} \tag{4.5}$$

$$\Rightarrow s(\mathbf{H}_p) = |S_1| \geq \frac{\mathrm{vol}(S_1)}{c_{\max}} \geq \frac{2/r_{\max} - \mu_1}{1-\mu_1} \cdot \frac{\mathrm{vol}(G_T)}{c_{\max}} \tag{4.6}$$

To prove (4.3)*, let $S_2 \subseteq Y_p$ be the set of active parity-checks of a smallest stopping set,*

$$\frac{|N(S_2)|c_{\max}}{\mathrm{vol}(S_2)} \geq \frac{\mathrm{vol}(N(S_2))}{\mathrm{vol}(S_2)} \geq \frac{\mathrm{vol}(G_T)}{\mu_1 \mathrm{vol}(G_T) + (1-\mu_1)\mathrm{vol}(S_2)} \tag{4.7}$$

Considering $N(S_2)$, it contains all active bits of the stopping set and some other bits that are not in the stopping set. For those active bits, all their neighbors are included in the set of S_2, and for the rest bits, some of their neighbors are in S_2 but others are not. Therefore, let $c_{avg}(N(S_2))$ be the average number of edges connected to $N(S_2)$ that are counted in $\mathrm{vol}(S_2)$, i.e., *$|N(S_2)|c_{avg}(N(S_2)) = \mathrm{vol}(S_2)$, then*

$$(4.7) \Rightarrow \frac{c_{\max}}{c_{avg}(N(S_2))} \geq \frac{\mathrm{vol}(G_T)}{\mu_1 \mathrm{vol}(G_T) + (1-\mu_1)\mathrm{vol}(S_2)} \tag{4.8}$$

Also, among the r_i neighbors of any node $y_i \in S_2$, at least 2 of them are active bits and the remaining $r_i - 2$ bits have at least one edge connected to S_2. In other words, assuming the r_i neighbors of y_i are $x_1, x_2, .., x_{r_i}$, among which x_1 and x_2 are active bits and $x_3, ..x_{r_i}$ each has at least one edge connected to S_2, it can be shown that at least $(c_1 + c_2 + r_i - 2)/r_i = 1 + (c_1 + c_2 - 2)/r_i \geq 1 + (2c_{\min} - 2)/r_{\max}$ edges connected to a neighbor of y_i are counted in vol(S_2) on average. Thus,

$$c_{avg}(N(S_2)) \geq 1 + (2c_{\min} - 2)/r_{\max} \tag{4.9}$$

$$\begin{aligned} (4.8) \quad &\Rightarrow \quad \frac{c_{\max} r_{\max}}{2c_{\min} + r_{\max} - 2} \geq \frac{\mathrm{vol}(G_T)}{\mu_1 \mathrm{vol}(G_T) + (1 - \mu_1)\mathrm{vol}(S_2)} \\ &\Leftrightarrow \quad \mathrm{vol}(S_2) \geq \frac{1 + (2c_{\min} - 2)/r_{\max} - \mu_1 c_{\max}}{(1 - \mu_1)c_{\max}} \cdot \mathrm{vol}(G_T) \end{aligned} \tag{4.10}$$

Noting that $s(\mathbf{H}_p)c_{\max} \geq 2|S_2| \geq 2\mathrm{vol}(S_2)/r_{\max}$, (4.3) is obtained. □

Lower bounds on minimum distance and stopping distance when the underlying graph is regular can be considered as a special case of Theorem 4.1, which is summarized in the following corollary.

Corollary 4.1. *The $d_{\min}$ of regular LDPC codes defined by the $p \times n$ parity-check matrix $\mathbf{H}_p$ satisfies*

$$d_{\min} \quad \geq \quad s(\mathbf{H}_p) \geq \frac{n(2c - \eta_1)}{cr - \eta_1} \tag{4.11}$$

$$d_{\min} \quad \geq \quad s(\mathbf{H}_p) \geq \frac{2n(2c + r - 2 - \eta_1)}{r(cr - \eta_1)} \tag{4.12}$$

where $n = |X_n|$ *and* $\eta_1 = \mu_1 cr$ *is the second largest eigenvalue of* $\mathbf{H}_p^T \mathbf{H}_p$.

Proof of Corollary 4.1. *It can be shown that, if* $\mathbf{H}_p$ *is regular,* i.e., $c_0 = ... = c_{n-1} = c$ *and* $r_0 = ... = r_{p-1} = r$, *the* $n \times n$ *square matrix* $\mathbf{H}_p^T \mathbf{H}_p$ *has* $\eta_0 = cr$ *as its unique largest single eigenvalue and* $\eta_1 = \mu_1 cr$ *as its second largest eigenvalue, where* μ_1 *is the second largest eigenvalue of* $\mathbf{A}_p^T \mathbf{A}_p$ *and* $\mathbf{A}_p$ *is the normalized incidence matrix defined in* (3.8). *The proof of Corollary 4.1 is then straightforward by plugging* $c_{\max} = c_{\min} = c$, $r_{\max} = r$, $\mathrm{vol}(G_T) = nc$ *and* $\eta_1 = \mu_1 cr$ *into* (4.2) *and* (4.3) *respectively.* □

It can be seen that the part of (4.11) and (4.12) corresponding to $d_{\min}$ coincide with Tanner's bit-oriented bound and parity-oriented bound for regular LDPC codes [60, Theorem 3.1,Theorem 4.1], respectively. We have also noted that Shin [53] generalized Tanner's work by deriving lower bounds on $d_{\min}$ for block-wise irregular LDPC codes, where some degree of regularity is still necessary. Our main contributions are the derivation of low bounds for general LDPC codes and demonstrating that Tanner's bounds are indeed lower bounds on stopping distance, and an immediate result of this is the explanation why Tanner's bounds on $d_{\min}$ are not tight.

Considering Gallager's $(20, 3, 4)$ regular LDPC code [22, Figure 2.1], it has $d_{\min} = 6$, and the given redundant parity-check matrix has stopping distance of 6, $r = 4$, $c = 3$ and

$\mu_1 = 0.5$. The bit-oriented bound does not apply, the parity-oriented bound is, however, 4. The parity-check matrix for this code is given as

$$\mathbf{H}_{15} = \begin{bmatrix}
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}$$

4.3 The Difference-set Codes: an Upper Bound on Stopping Redundancy

Stopping redundancy was introduced by Schwartz and Vardy [51]. Lower and upper bounds were also provided for binary and ternary extended Golay codes, the family

of Reed-Muller codes and Maximum-Distance Separable (MDS) codes. In this section, we will provide an upper bound on stopping redundancy of the family of difference-set codes, which is also known as the type-I 2-D projective geometry LDPC, or PG-LDPC, codes [27]. Specifically, assuming $\mathcal{C}$ is a difference-set code of length n and minimum distance $d_{\min}$, we will show that there exists a $n \times n$ redundant parity-check matrix $\mathbf{H}_n$ such that $s(\mathbf{H}_n) = d_{\min}$, therefore $\rho(\mathcal{C}) \leq n$.

Though there are relatively few codes in the family of difference-set codes, they are nearly as powerful as the best known cyclic codes in the range of practical interest [30]. Furthermore, several recent experiments [27, 33, 77] suggested that this family of codes can perform very well under iterative decoding. It should also be noted that, in [67], *pseudo-weight enumerators of pseudo-codewords* of both type-I 2-D PG-LDPC and type-I 2-D Euclidean geometry LDPC, EG-LDPC, were discussed, and stopping redundancy of these two families of codes can be derived from their pseudo-weight enumerator as well.

4.3.1 A lemma on cyclic parity-check matrices

To analyze the algebraic properties of cyclic codes, the components of a row vector[1] $\mathbf{v} = (v_0, v_1, ..., v_{n-1})$ are usually treated as coefficients of a polynomial, *i.e.,* $\mathbf{v}(X) = v_0 + v_1 X + v_2 X^2 + ... + v_{n-1} X^{n-1}$. Since the correspondence between $\mathbf{v}$ and $\mathbf{v}(X)$ is one-to-one, we use the terms "row vector" and "polynomial" interchangeably hereafter.

[1] Different from previous sections, where column vectors are used, row vectors are used here.

It is known that a cyclic code is uniquely specified by its *parity polynomial* [30], which is of degree k and defined as:

$$\mathbf{h}(X) = 1 + h_1 X + h_2 X^2 + ... + h_{k-1} X^{k-1} + X^k \tag{4.13}$$

The corresponding parity-check matrix can be written as:

$$\mathbf{H}_{p_0} = \begin{bmatrix} \mathbf{h}^*(X) \bmod (X^n+1) \\ X\ \mathbf{h}^*(X) \bmod (X^n+1) \\ . \\ . \\ X^{p_0-1}\mathbf{h}^*(X) \bmod (X^n+1) \end{bmatrix}_{p_0 \times n} = \begin{bmatrix} \mathbf{h}^*(X) \\ X\ \mathbf{h}^*(X) \\ . \\ . \\ X^{p_0-1}\mathbf{h}^*(X) \end{bmatrix}_{p_0 \times n} = \begin{bmatrix} \mathbf{h}_0^* \\ \mathbf{h}_1^* \\ . \\ . \\ \mathbf{h}_{p_0-1}^* \end{bmatrix}_{p_0 \times n} \tag{4.14}$$

where $p_0 = n - k$, $\mathbf{h}^*(X) = X^k \mathbf{h}(X^{-1})$ is the reciprocal of $\mathbf{h}(X)$ and $\mathbf{h}_i^*$, $0 \le i \le p_0 - 1$, are row vectors. A parity-check matrix of this form is called a *cyclic parity-check matrix* because $\mathbf{h}_i^*$ is the i-th cyclic shift of $\mathbf{h}_0^*$ to the right, $1 \le i \le p_0 - 1$.

It is also known that the parity-check matrix for a given cyclic code is usually not unique. One interesting result is the following lemma.

Lemma 4.1. *Assuming that* $\mathbf{h}(X)$ *is the parity polynomial of an [n,k,*$d_{\min}$*] cyclic code* $\mathcal{C}$*, if there exists another polynomial* $\mathbf{z}(X) = \mathbf{h}(X)\mathbf{f}(X)$ *such that:*

- $\mathbf{f}(X)$ *is a non-zero polynomial of degree* $f < p_0 = n - k$*;*
- *the greatest common divisor of* $\mathbf{f}(X)$ *and* $X^n + 1$ *is* 1, i.e., $\mathrm{GCD}(\mathbf{f}(X), X^n + 1) = 1$*;*

then,

$$\mathbf{H}_{p_0}(\mathbf{z}) = \begin{bmatrix} \mathbf{z}^*(X) \bmod (X^n+1) \\ X\ \mathbf{z}^*(X) \bmod (X^n+1) \\ . \\ . \\ X^{p_0-1}\mathbf{z}^*(X) \bmod (X^n+1) \end{bmatrix}_{p_0\times n} = \begin{bmatrix} \mathbf{z}_0^* \\ \mathbf{z}_1^* \\ . \\ . \\ \mathbf{z}_{p_0-1}^* \end{bmatrix}_{p_0\times n} \tag{4.15}$$

is also a cyclic parity-check matrix for $\mathcal{C}$*, where*

$$\mathbf{z}^*(X) = X^{k+f}\mathbf{z}(X^{-1}) = X^k\mathbf{h}(X^{-1})X^f\mathbf{f}(X^{-1}) = \mathbf{h}^*(X)\mathbf{f}^*(X) \tag{4.16}$$

is the reciprocal of $\mathbf{z}(X)$*.*

Proof of Lemma 4.1. *To show* $\mathbf{H}_{p_0}(\mathbf{z})$ *is a valid parity-check matrix of* $\mathcal{C}$*, it suffices to show that its row vectors belong to the row space of* $\mathbf{H}_{p_0}$ *and they are linearly independent. Since the row space of* $\mathbf{H}_{p_0}$ *is of dimension* p_0*, the row space of* $\mathbf{H}_{p_0}(\mathbf{z})$ *is then the same as the row space of* $\mathbf{H}_{p_0}$*. Therefore,* $\mathbf{H}_{p_0}(\mathbf{z})$ *is a cyclic parity-check matrix for* $\mathcal{C}$*.*

Noting that $\mathbf{z}^*(X) = \mathbf{h}^*(X)\mathbf{f}^*(X)$ *and* $\mathrm{GCD}(\mathbf{f}^*(X), X^n+1) = 1$*,* $\mathbf{z}_0^*$ *is a non-zero row vector and is a linear combination of the row vectors of* $\mathbf{H}_{p_0}$*. Also, the cyclic property of the row space of* $\mathbf{H}_{p_0}$ *guarantees that it contains all the cyclic shift of* $\mathbf{z}_0^*$ *to the right.*

To prove part two, assuming that the row vectors of $\mathbf{H}_{p_0}(\mathbf{z})$ *are linearly dependent, thus there exist a set of variables* $\alpha_i \in \{0,1\}$*,* $0 \le i \le p_0-1$*, such that not all of them are zero and*

$$\alpha_0\mathbf{z}_0^* \oplus \alpha_1\mathbf{z}_1^* \oplus ... \oplus \alpha_{p_0-1}\mathbf{z}_{p_0-1}^* = \mathbf{0} \tag{4.17}$$

where $\oplus$ *is modulo-2 addition and* $\mathbf{0}$ *is a zero row vector. Equivalently,*

$$\alpha_0\mathbf{z}^*(X) \oplus \alpha_1 X\mathbf{z}^*(X) \oplus ... \oplus \alpha_{p_0-1}X^{p_0-1}\mathbf{z}^*(X) \equiv 0 \bmod (X^n+1) \quad (4.18a)$$

$$\mathbf{f}^*(X)\left[\alpha_0\mathbf{h}^*(X) \oplus \alpha_1 X\mathbf{h}^*(X) \oplus ... \oplus \alpha_{p_0-1}X^{p_0-1}\mathbf{h}^*(X)\right] \equiv 0 \bmod (X^n+1) \quad (4.18b)$$

Noting that $\mathrm{GCD}(\mathbf{f}^*(X), X^n+1) = 1$*, thus*

$$[\alpha_0 \oplus \alpha_1 X \oplus ... \oplus \alpha_{p_0-1}X^{p_0-1}]\mathbf{h}^*(X) \equiv 0 \bmod (X^n+1) \quad (4.19a)$$

$$\Leftrightarrow \alpha_0\mathbf{h}_0^* \oplus \alpha_1\mathbf{h}_1^* \oplus ... \oplus \alpha_{p_0-1}\mathbf{h}_{p_0-1}^* = \mathbf{0} \quad (4.19b)$$

contradicts with the fact that row vectors of $\mathbf{H}_{p_0}$ *are linearly independent. Thus, row vectors of* $\mathbf{H}_{p_0}(\mathbf{z})$ *are linearly independent.*

4.3.2 An upper bound on stopping redundancy of the difference-set codes

Definition 4.1. *[30][Ch.5] Let* $D = \{d_0, d_1, ..., d_q\}$ *be a set of* $q+1$ *non-negative integers such that* $0 \le d_0 < d_1 < ... < d_q \le q(q+1)$*, and for each* $0 < t < q(q+1)$*, there exist one and only one ordered pair* $0 \le i \ne j \le q$ *such that* $d_i - d_j \equiv t \bmod q(q+1)$*, then* D *is a perfect simple* ***difference set*** *of order* q*.* □

It can be shown that, if D is perfect simple difference set, $D' = \{0, d_1 - d_0, d_2 - d_0, ..., d_{q-1} - d_0, d_q - d_0\}$, $\overline{D} = \{q(q+1) - d_q, q(q+1) - d_{q-1}, ..., q(q+1) - d_1, q(q+1) - d_0\}$ and $\overline{D}' = \{0, d_q - d_{q-1}, ..., d_q - d_1, d_q - d_0\}$ are also perfect simple difference sets. It is also known that perfect simple difference sets exist for order $q = \alpha^\beta$, where α is prime and β

is any positive integer. However, the case of $q = 2^\beta$ corresponds to the most commonly studied difference-set codes.

Definition 4.2. *[30][Ch.5] Let $D = \{0, d_1, ..., d_q\}$ be a perfect simple difference set of order $q = 2^\beta$, define the polynomial $\mathbf{z}(X) = 1+X^{d_1}+X^{d_2}+...+X^{d_q}$. Let $n = q(q+1)+1 = 2^{2\beta} + 2^\beta + 1$, $k = 2^{2\beta} + 2^\beta - 3^\beta$ and $\mathbf{h}(X)$ be the greatest common divisor of $\mathbf{z}(X)$ and $X^n + 1$, i.e., $\mathbf{h}(X) = \mathrm{GCD}(\mathbf{z}(X), X^n + 1)$, the cyclic code defined by the parity-check matrix with $p_0 = n - k$,*

$$\mathbf{H}_{p_0} = \begin{bmatrix} \mathbf{h}^*(X) \\ X\ \mathbf{h}^*(X) \\ . \\ . \\ X^{p_0-1}\mathbf{h}^*(X) \end{bmatrix}_{p_0 \times n} = \begin{bmatrix} \mathbf{h}_0^* \\ \mathbf{h}_1^* \\ . \\ . \\ \mathbf{h}_{p_0-1}^* \end{bmatrix}_{p_0 \times n} \tag{4.20}$$

*is an $[n, k, d_{\min} = q + 2]$ **difference-set code**, where $\mathbf{h}^*(X)$ is the reciprocal of $\mathbf{h}(X)$.* □

Theorem 4.2. *The stopping redundancy of an $[n,k,d_{\min}]$ difference-set code is less than or equal to n, where $n = q^2 + q + 1$, $k = q^2 + q - 3^\beta$, $d_{\min} = q + 2$, $q = 2^\beta$ and β is any positive integer.*

Proof of Theorem 4.2. *Since* $\mathbf{h}(X) = \mathrm{GCD}(\mathbf{z}(X), X^n + 1)$*, where polynomial* $\mathbf{z}(X)$ *corresponds to the perfect simple difference set* D *of order* $q = 2^\beta$*, there exists a polynomial* $\mathbf{f}(X)$ *such that* $\mathbf{z}(X) = \mathbf{h}(X)\mathbf{f}(X)$ *and* $\mathrm{GCD}(\mathbf{f}(X), X^n + 1) = 1$*. Using Lemma 4.1,*

$$\mathbf{H}_{p_0}(\mathbf{z}) = \begin{bmatrix} \mathbf{z}^*(X) \bmod (X^n+1) \\ X\ \mathbf{z}^*(X) \bmod (X^n+1) \\ . \\ . \\ X^{p_0-1}\mathbf{z}^*(X) \bmod (X^n+1) \end{bmatrix}_{p_0\times n} = \begin{bmatrix} \mathbf{z}_0^* \\ \mathbf{z}_1^* \\ . \\ . \\ \mathbf{z}_{p_0-1}^* \end{bmatrix}_{p_0\times n} \tag{4.21}$$

is a parity-check matrix of $\mathcal{C}$*. By adding row vectors corresponding to* $X^i\mathbf{z}^*(X) \bmod (X^n+1)$*,* $p_0 \le i \le n-1$*, to* $\mathbf{H}_{p_0}(\mathbf{z})$*, a* $n \times n$ *redundant parity-check matrix* $\mathbf{H}_n(\mathbf{z})$ *is formed,*

$$\mathbf{H}_n(\mathbf{z}) = \begin{bmatrix} \mathbf{z}^*(X) \bmod (X^n+1) \\ . \\ X^{p_0-1}\mathbf{z}^*(X) \bmod (X^n+1) \\ . \\ X^{n-1}\mathbf{z}^*(X) \bmod (X^n+1) \end{bmatrix}_{n\times n} = \begin{bmatrix} \mathbf{z}_0^* \\ . \\ \mathbf{z}_{p_0-1}^* \\ . \\ \mathbf{z}_{n-1}^* \end{bmatrix}_{n\times n} \tag{4.22}$$

which has both rows and columns weight $q+1$*. Then*

$$\mathbf{A}_n(\mathbf{z}) = \frac{1}{(q+1)^2}\mathbf{H}_n(\mathbf{z}) \tag{4.23}$$

Furthermore, $1 = \widetilde{\mu}_0 > \widetilde{\mu}_1 = \widetilde{\mu}_2 ... \widetilde{\mu}_{n-1} = \frac{q}{(q+1)^2}$ *are eigenvalues of* $\mathbf{A}_n(\mathbf{z})^T \mathbf{A}_n(\mathbf{z})$*, which has diagonal entries of* $\frac{1}{q+1}$ *and off-diagonal entries of* $\frac{1}{(q+1)^2}$*,* i.e.,

$$\mathbf{A}_n(\mathbf{z})^T \mathbf{A}_n(\mathbf{z}) = \begin{bmatrix} \frac{1}{q+1} & \frac{1}{(q+1)^2} & \frac{1}{(q+1)^2} & \cdot & \cdot & \cdot & \frac{1}{(q+1)^2} \\ \frac{1}{(q+1)^2} & \frac{1}{q+1} & \cdot & \cdot & \cdot & \cdot & \frac{1}{(q+1)^2} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \frac{1}{(q+1)^2} & \frac{1}{(q+1)^2} & \frac{1}{(q+1)^2} & \cdot & \cdot & \cdot & \frac{1}{q+1} \end{bmatrix} \tag{4.24}$$

Then, the bit-oriented bound is

$$s(\mathbf{H}_n(\mathbf{z})) \geq \frac{\frac{2}{q+1} - \frac{q}{(q+1)^2}}{1 - \frac{q}{(q+1)^2}} (q^2 + q + 1) = q + 2 = d_{\min} \tag{4.25}$$

Therefore, the stopping distance of $\mathbf{H}_n(\mathbf{z})$ *equals* $d_{\min}$ *of the code, and the stopping redundancy of the family of difference-set codes,* $\rho(\mathcal{C}) \leq n$ *= the length of the code.* □

It should be noted that the class of difference-set codes is a subset of a larger class of linear codes known as one-step majority-logic decodable code, so is the class of Reed-Muller codes which was used in [51] as an example. Therefore, its stopping redundancy must satisfy another upper bound, which is derived in Appendix B for one-step majority-logic decodable codes. However, upper bounds in B are too weak to be useful here.

Furthermore, for redundant parity-check matrix $\mathbf{H}_n(\mathbf{z})$, we can not only show that its stopping distance equals the minimum distance, but also the number of smallest stopping sets equals the number of minimum weight codewords, *i.e.,*

Theorem 4.3. *For the family of $[n, k, d_{\min}]$ difference-set codes,*

$$A_d[d_{\min}, \mathcal{C}] = A_s[s(\mathbf{H}_n(\mathbf{z})), \mathbf{H}_n(\mathbf{z})] \tag{4.26}$$

where

$$A_d[\omega, \mathcal{C}] = \text{number of weight } \omega \text{ codewords} \tag{4.27}$$

$$A_s[|S|, \mathbf{H}_p] = \text{number of size } |S| \text{ stopping sets} \tag{4.28}$$

Proof of Theorem 4.3. *To show* (4.26)*, it suffices to show that, by letting variables in it be* 1 *and the rest be* 0*, every smallest stopping set corresponds to a minimum weight codeword. Without loss of generality, assuming that* $\{x_1, x_2, ..., x_{q+2}\}$ *forms a stopping set and* y_1*,* y_2*, ...* y_{q+1} *are neighbors of* x_1*, there exists at least one* x_j*,* $2 \leq j \leq q+2$*, such that* $y_i \in N(x_j)$ *because* $\{x_1, x_2, ..., x_{q+2}\}$ *is a stopping set. However, as* $|N(x_1) \bigcap N(x_2)| = ... = |N(x_1) \bigcap N(x_{q+2})| = 1$ *and* $|N(x_1)| = q + 1$*, it can be shown that there is only one such* x_j *for each* y_i *so that all neighbors of* x_1 *are of degree two. Similarly, we can prove this for* x_j*,* $2 \leq j \leq q + 2$*. Thus, let* $x_j = 1$ *for* $1 \leq j \leq q + 2$ *and* $x_j = 0$ *otherwise, a minimum weight codeword, which is of weight* $q + 2$*, is formed.* □

Using (4.26), we can argue that, when the erasure probability is small, the performance of the iterative message passing algorithm can be very close to that of the ML decoding. This can be verified using the $[21, 11, 6]$ difference-set code $\mathcal{C}_{21}$ derived from the difference set $D = \{0, 3, 4, 9, 11\}$, where $\mathbf{h}(X) = \mathbf{z}(X) = 1 + X^3 + X^4 + X^9 + X^{11}$, $d_{\min} = 6$ and $A_d[6, \mathcal{C}_{21}] = 168$. It can be shown that $\rho(\mathcal{C}_{21}) \leq 12$, and $A_s[s(\mathbf{H}_p(\mathbf{z})), \mathbf{H}_p(\mathbf{z})] = 168$ if

p	10	11	12	13	14	15	16..20	21
$s(\mathbf{H}_p(\mathbf{z}))$	5	5	6	6	6	6	6	6
$A_s[s(\mathbf{H}_p(\mathbf{z})), \mathbf{H}_p(\mathbf{z})]$	8	4	186	171	169	168	168	168

Table 4.1: $s(\mathbf{H}_p(\mathbf{z}))$ and $A_s[s(\mathbf{H}_p(\mathbf{z}), \mathbf{H}_p(\mathbf{z})]$ v.s. the number of rows of $\mathbf{H}_p(\mathbf{z})$

$p \geq 15$, where Theorem 4.2 and Theorem 4.3 provide bounds $\rho(\mathcal{C}_{21}) \leq 21$ and $p \geq 21$, respectively. These results are summarized in Table 4.1.

Figure 4.1 evaluates the performance of iterative decoding for $\mathcal{C}_{21}$ on the erasure channel as a function of p, the number of rows of the cyclic redundant parity-check matrix $\mathbf{H}_p(\mathbf{z})$ in the form similar to (4.22). The general belief, that the iterative decoder will perform better if redundant parity-checks are added to the Tanner Graph, is supported by this simulation. For example, when the channel erasure probability is 0.12, the probability of block error is 0.001 if $p = 10$, but this number is 0.00048 if $p = 15$ and 0.00047 when $p = 21$. The performance of ML decoding is also shown in 4.1 and is observed to be identical to that of the $p = 21$ iterative decoding algorithm.

Furthermore, Figure 4.2 evaluates the performance of iterative decoding for $\mathcal{C}_{21}$ on the AWGN channel as a function of p. It can be seen that, by increasing the number of parity-checks from 10 to 21, there is a performance gain of 0.5 dB (in E_b/N_0) and the curve corresponding the $p = 21$ is only 0.25 dB away from that of the optimal ML decoding. Also, it should be noted that the performance gain is not significant by increasing p from 14 to 21.

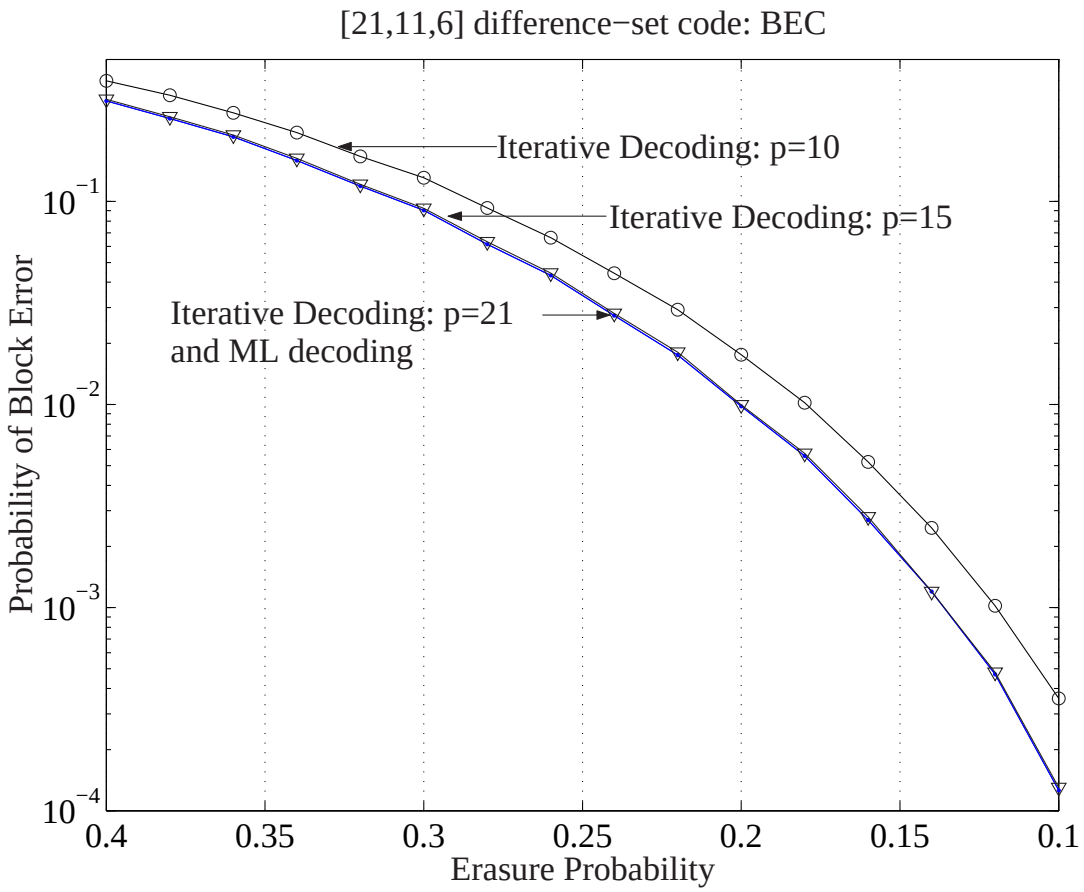

Figure 4.1: Performance of iterative decoder as a function of p and Maximum-Likelihood decoder for $[21, 11, 6]$ difference-set code on BEC. Note that the curve of ML decoding and iterative decoding with $p = 21$ coincide.

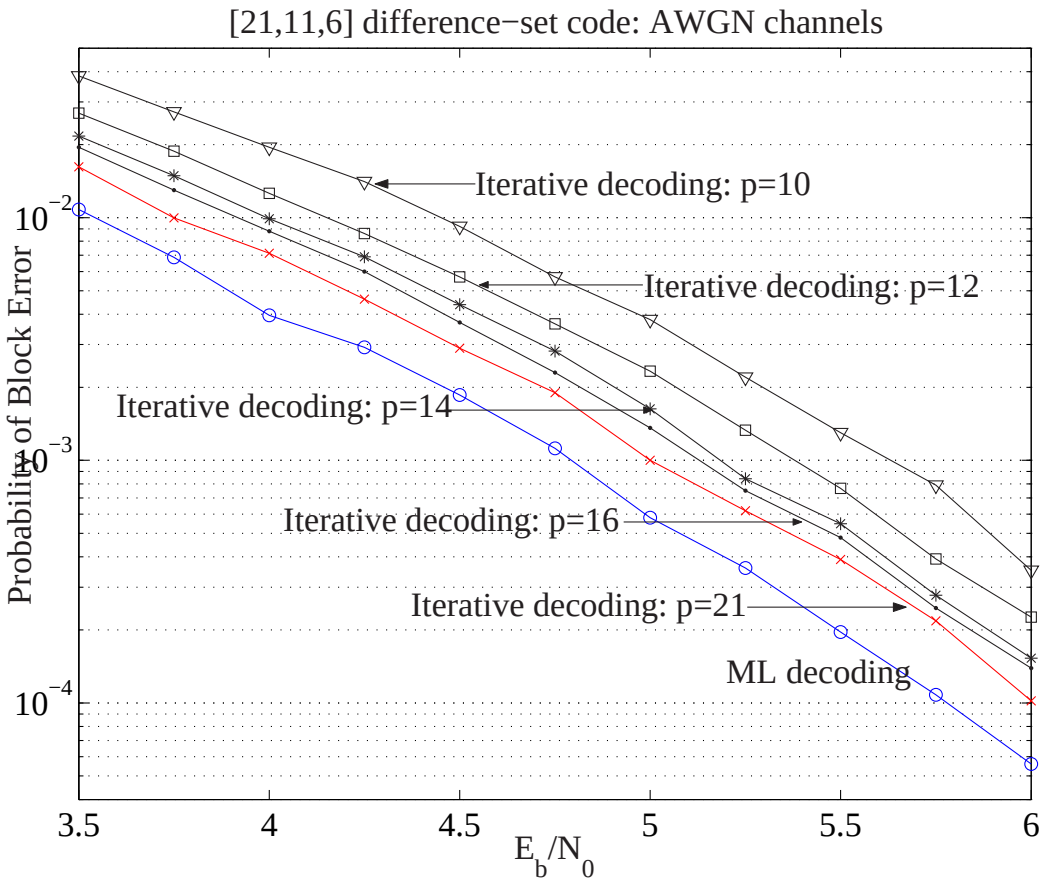

Figure 4.2: Performance of iterative decoder as a function of p and Maximum-Likelihood decoder for $[21, 11, 6]$ difference-set code on AWGN.

4.4 Discussions for future research

In the previous section, we have demonstrated that, by properly adding redundant parity-checks to a Tanner graph representing simple difference-set codes, the performance of the associated iMPA can be improved. Generally, however, an open issue is the development of methods to choose such redundant parity-checks in order to improve the performance of the iterative decoding. In other words, given a $p \times n$ parity-check matrix $\mathbf{H}_p$, the question is how to find a non-zero row vector $\mathbf{h}_{\text{opt}}$ from the *row space* [56] of $\mathbf{H}_p$ such that the iMPA associated with

$$\mathbf{H}_{p+1} = \begin{bmatrix} \mathbf{H}_p \\ \\ \mathbf{h}_{\text{opt}} \end{bmatrix}$$

performs better in the sense of block error rate (or bit error rate) than the iMPA associated with $\mathbf{H}_p$ and the iMPA associated with

$$\mathbf{H}'_{p+1} = \begin{bmatrix} \mathbf{H}_p \\ \\ \mathbf{h}' \end{bmatrix}$$

where $\mathbf{h}' \neq \mathbf{h}_{\text{opt}}$ is any non-zero vector belongs to the row space of $\mathbf{H}_p$.

It is believed that this problem is hard in general because the relation between properties of a Tanner graph and the performance of the associated iMPA is generally unknown. Even for the BEC case, where the performance of iterative decoding is determined by the stopping sets of the Tanner graph, the search for $\mathbf{h}_{\text{opt}}$ is hardly possible because finding

minimum stopping sets for any given parity-check matrix is as hard as finding the minimum distance of the code and the problem of finding $d_{\min}$ is known to be intractable for large codes [9, 62].

However, noting that the lower bounds in Theorem 4.1 are functions of the specific (redundant) parity-check matrix being used, a simple heuristic to this problem is to find some row vector $\mathbf{h}$ from the row space of $\mathbf{H}_p$ such that the lower bound obtained by (4.2) and/or (4.3) increases.

One problem with this approach is that Theorem 4.1 usually provides relatively weak lower bounds on stopping distance. Noting that the main reason for this is because only the largest and second largest eigenvalues of $\mathbf{A}_p^T\mathbf{A}_p$ are used, if adding $\mathbf{h}$ makes $\mathbf{A}_p^T\mathbf{A}_p$ have two non-zero eigenvalues, *i.e.,*one is the unique single eigenvalue $\mu_0 = 1$ and the other is eigenvalue μ_1 of multiplicity $p-1$ if $p \leq n$ or $n-1$ if $p > n$, the lower bounds in Theorem 4.1 can be tight.

Therefore, define $\overline{\mu}(p)$ and $\overline{\sigma^2}(p)$ as the average and variance of non-zero eigenvalues of $\mathbf{A}_p^T\mathbf{A}_p$ except $\mu_0 = 1$, respectively, *i.e.,*

$$\overline{\mu}(p) = \begin{cases} \frac{1}{p-1}\sum_{j=1}^{p-1}\mu_j = \frac{T_R(\mathbf{A}_p^T\mathbf{A}_p)-1}{p-1} & \text{if p } \leq \text{ n,} \\ \\ \frac{1}{n-1}\sum_{j=1}^{n-1}\mu_j = \frac{T_R(\mathbf{A}_p^T\mathbf{A}_p)-1}{n-1} & \text{otherwise.} \end{cases} \tag{4.29}$$

$$\overline{\sigma^2}(p) = \begin{cases} \frac{1}{p-1}\sum_{j=1}^{p-1}[\mu_j - \overline{\mu}(p)]^2 = \frac{T_R[(\mathbf{A}_p^T\mathbf{A}_p)^2]-1}{p-1} - [\overline{\mu}(p)]^2 & \text{if p } \leq \text{ n,} \\ \\ \frac{1}{n-1}\sum_{j=1}^{n-1}[\mu_j - \overline{\mu}(p)]^2 = \frac{T_R[(\mathbf{A}_p^T\mathbf{A}_p)^2]-1}{n-1} - [\overline{\mu}(p)]^2 & \text{otherwise.} \end{cases} \tag{4.30}$$

where $T_R(\mathbf{A}_p^T\mathbf{A}_p)$ denotes the *trace* [56] of the $n \times n$ matrix $\mathbf{A}_p^T\mathbf{A}_p$, and let $B(\mathbf{H}_p)$ be the bit-oriented bound as defined in (4.2), we propose the following search strategy,

1. For a given $[n, k, d_{\min}]$ linear code $\mathcal{C}$, the algorithm starts with its parity-check matrix $\mathbf{H}_p = \mathbf{H}_{n-k}$, $B(\mathbf{H}_p)$ and $\overline{\sigma^2}(p)$ are then obtained using (4.2) and (4.30), respectively;

2. Search for a row vector $\mathbf{h}$ from the row space of $\mathbf{H}_p$ such that $B(\mathbf{H}_{p+1}) > B(\mathbf{H}_p)$ and $\overline{\sigma^2}(p+1) < \overline{\sigma^2}(p)$, where

$$\mathbf{H}_{p+1} = \begin{bmatrix} \mathbf{H}_p \\ \mathbf{h} \end{bmatrix}$$

3. If $B(\mathbf{H}_{p+1}) = d_{\min}$, the search stops; otherwise, $p \longleftarrow p+1$ and go back to step 2.

We believe this algorithm may work because, by making $B(\mathbf{H}_p)$ monotonically increase as a function of p, $B(\mathbf{H}_p)$ may approach $d_{\min}$. Also, by making $\overline{\sigma^2}(p)$ monotonically decrease as a function of p, the obtained lower bound becomes tighter. However, it should be pointed out that this algorithm is not "optimal" in general because its goal is to find some redundant parity-check matrix of $\mathcal{C}$, the stopping distance of which equals $d_{\min}$, but not such matrix with the minimum number of rows. In other words, this algorithm may be useful for finding upper bounds on stopping redundancy for a given linear code, but not the stopping redundancy.

Furthermore, it should be noted that the convergence of the algorithm is not guaranteed and we leave this as one of the future research problems for interested readers. Another issue of this algorithm is that, it is possible that $B(\mathbf{H}_p) < d_{\min}$ but $\overline{\sigma^2}(p)$ is

very small which makes it very hard to find new $\mathbf{h}$. Therefore, it may be necessary for the algorithm to go back to some previous stage and restart the search. Considering the following hypothetical scenario: for a given $[n, k, d_{\min}]$ linear code $\mathcal{C}$ with parity-check matrix $\mathbf{H}_p$, the previous algorithm is carried of K times, and a series of K redundant parity-check matrices, $\mathbf{H}_{p+1}$,...$\mathbf{H}_{p+k}$,...$\mathbf{H}_{p+K}$, are obtained. However, it is observed that $B(\mathbf{H}_{p+K}) < d_{\min}$ and $\overline{\sigma^2}(p)$ is close to 0. To make further search possible, a number of the obtained matrices may be dropped, say, from $\mathbf{H}_{p+k+1}$ to $\mathbf{H}_{p+K}$, the search restarts with $\mathbf{H}_{p+k}$ and a new series of redundant parity-check matrices is obtained. How to decide and how many matrices that need to be dropped are two research topics related to this problem.

Another interesting application of Theorem 4.1 is whether it can help finding $d_{\min}$ of a given LDPC codes, which is known to be NP-hard in general [9, 62]. Noting that $B(\mathbf{H}_p)$ may converge to the minimum distance as p goes to infinity, if the proposed search algorithm can be carried several times with different initial conditions and $B(\mathbf{H}_p)$ always converges to the same number, it is conceivable that the minimum distance of the code is obtained. Again, this may be an interesting research topic for interested readers.

4.5 Summary

Using results from the previous chapter, we derived lower bounds on the stopping distance of linear codes defined by a given parity-check matrix, and pointed out the relationship between our bounds and Tanner's bit-oriented bound and parity-oriented bound on minimum distance for regular LDPC codes.

Furthermore, these lower bounds can lead an upper bound on stopping redundancy of the family of difference-set codes. Theoretical and simulation results also showed that, by properly adding redundant parity-checks to the parity-check matrix of difference-set codes, not only can we form redundant parity-check matrix with stopping distance equal minimum distance, but also the spectrum of stopping sets is close to that of the valid codeword. Therefore, the performance of iterative decoding is close the that of the ML decoding as well.

Chapter 5

Expansions of Tanner Graphs and Message-Passing Algorithms

5.1 Introduction

Though it is known that the performance of iterative decoding over Tanner graphs on erasure channels is determined by the spectrum of the stopping sets of the Tanner graphs, the same problem is still considered unsolved for binary symmetric channels (BSCs) and additive white Gaussian noise (AWGN) channels.

Using density evolution, three main results are obtained in [47]. Namely, let $P_e^n(l)$ be the expected (over the ensemble of the code, the choice of the message and the realization of the noise) fraction of incorrect messages passed in the l-th iteration for a LDPC code of length n, Richardson and Urbanke first proved that the actual fraction of errors in the l-th iteration for a particular instance among the ensemble converges to $P_e^n(l)$ as n goes to infinity. Then, they showed that $P_e^n(l)$ converges to the cycle-free case, *i.e.,*the graphical representation does not contain cycles of length $2l$ or less. Third, they demonstrated the threshold phenomenon, *i.e.,* letting the channel be characterized by a single parameter

σ, there exists a channel parameter σ^* such that $\lim_{n\to\infty\ l\to\infty} P_e^n(l) = 0$ if $\sigma < \sigma^*$, and this limit is always non-zero if $\sigma > \sigma^*$. Later, by computer search, some good degree distributions in the sense that their *design rates* approach the Shannon capacity were obtained [46].

Sipser and Spielman proposed another approach to this problem [55]. By introducing the expansion of variable nodes and defining several iterative decoding algorithms on BSCs, they proved that, for regular LDPC codes, their algorithms can correct a fraction of errors if the expansion properties of the underlying graphs is good enough. For example, as was pointed out in section 3.3.2, for Spielman's simple sequential decoding algorithm, any error pattern containing no more than $m/2$ random errors can be corrected if $\delta_{\min}(i) \geq 3/4$ for $1 \leq i \leq m$. Also, they argued that their codes are asymptomatically good because it can be proved that, with high probability, a randomly generated regular LDPC code will have relatively good expansion property. An extension of their approach to irregular LDPC codes was provided in [32].

The expander graph argument on regular LDPC codes was later generalized by Burshtein and Miller [12], where Gallager's hard-decoding and soft-decoding [22] were analyzed for irregular LDPC codes and similar results were obtained. However, their results should be considered as arguments suggesting why iterative decoding on randomly generated Tanner graphs can perform well, rather than rigorous proofs relating the variable expansions of the graphical representation with the performance of associated iMPAs.

In this chapter we propose a possible solution to a related problem, which can be stated as follows. Let G_1 and G_2 be two different graphical representations of a binary linear code $\mathcal{C}$, and let the same, in the sense of definition of messages, rule of updates,

schedule and stopping criterion, iterative message-passing algorithm runs on both graphs; is there a systematic way to determine which instance will have smaller block error rate by just analyzing the graphs?

Our criteria for this problem also use the expansion properties of the graphs. However, we use average expansions instead of minimum expansions, which were used in [55], because we believe that the former will relate the graphical representation and performance of the associated iMPAs more accurately. Also, we propose a practical method for calculating lower bounds on the average expansions, where results from Chapter 3 are used.

It should be noted that our theory has not been fully developed so that results in this chapter are preliminary. The main criteria is provided in section 5.2 and we argue that they are valid, which is also verified by examples.

It should also be noted that there are other approaches trying to address similar problems. By deriving algebraic eigenvalue-based lower bound and linear-programming-based lower bound on minimum distance, Tanner [60] suggested that the performance of iterative decoding on loopy Tanner graphs is related to these lower bounds. His results were generalized by Vontobel and Koetter [65] to derive lower bounds on minimum pseudo-weight of pseudo-codewords [20]. Also, as suggested by Gallager [21, 22], the girth and cycle structures are believed to be related to the performance of iMPAs on loopy graphs [2, 3, 25, 57, 57, 63]. More recently, Halford and Chugg [24] try to evaluate the performance of iMPAs on loopy graphs using the distribution of short cycles.

5.2 Average Expansions v.s. Performance of Associated iMPAs

Before we provide the two criteria that relate the average expansions of variable subsets of Tanner graphs with the performance of associated iMPAs running on them, several things must be clarified. Otherwise, the comparison would be unfair and the conclusion would be of limited use. Namely, we should clarify the channel model, how the decoder processes, and how the algorithm is evaluated.

The first criterion focuses on the performance of Spielman's simple sequential decoding [55] over "different" Tanner graphs on the BSCs. Assuming G_1 and G_2 are both Tanner graphs with n variables and p parity-checks, we consider them the "same " if one can be obtained from the other by simply changing the order of the variables and/or the order of the parity-checks. Otherwise, they are different Tanner graphs.

Iterative decoding on AWGN channels is discussed in the second criterion. Regarding the processing of this decoder, the "standard" soft-decision iMPA on Tanner graphs is used, which has been well developed and used by many researchers over the years [2, 3, 14, 28, 29, 39, 40, 70]. Details about this algorithm have been covered in Section 2.4 regarding the processing on Figure 2.2(b).

In both cases, block error rate is used to evaluate the performance of the algorithms.

5.2.1 System model

Consider the binary linear code $\mathcal{C}$ defined by a Tanner graph with n variable vertices and p parity-checks, as shown in Figure 5.1. Let $n \times 1$ column vector $\mathbf{x} = (x_0, x_1, .., x_k, .., x_{n-1})^T$

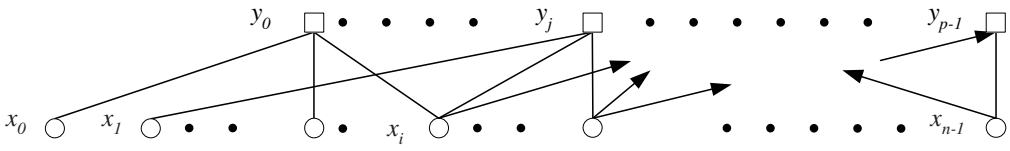

Figure 5.1: A Tanner Graph

represent a valid codeword of $\mathcal{C}$, where $x_i \in \{0, 1\}$, $0 \leq i \leq n-1$, and these n variables are transmitted to the receiver together, which is referred to as a *block*. At the receiver side, n observations, z_i, $0 \leq i \leq n-1$, are available, and a properly defined iterative decoding algorithm over the Tanner graph will try to recover $\mathbf{x}$ from $\mathbf{z} = (z_0, z_1, ..., z_k, ...z_{n-1})^T$. Let the output of the decoder be $\widetilde{\mathbf{x}}$, a block error is declared if $\widetilde{\mathbf{x}} \neq \mathbf{x}$.

The two signal models adopted in this chapter are the BSC and AWGN channel, respectively. For the BSC, $z_i \in \{0, 1\}$, and

$$P_r\{z_i = 1|x_i = 1\} = P_r\{z_i = 0|x_i = 0\} = 1 - \epsilon$$

$$P_r\{z_i = 0|x_i = 1\} = P_r\{z_i = 1|x_i = 0\} = \epsilon$$

where ϵ is defined as the *cross-over probability* of the channel. For AWGN, z_i's are real numbers and

$$z_i = \sqrt{E_s}(-1)^{x_i} + n_i \qquad 0 \leq \ i \ \leq n-1 \tag{5.1}$$

where $\sqrt{E_s}$ is the energy per transmitted symbol and n_i's are zero-mean and $N_0/2$-variance white Gaussian random variables.

5.2.2 The iterative decoder

Two iterative decoders are considered in Criterion 5.1 and Criterion 5.2, respectively.

The first one is Spielman's simple sequential decoding for BSCs defined in [55, pp. 1713]. Given n observations from the BSCs, $z_i \in \{0, 1\}$, $0 \leq i \leq n-1$, the p parity-checks are calculated. A parity-check is even if the sum of the connected variables are even, and it is odd otherwise. Spielman's sequential decoding algorithm checks the number of even parity-checks and the number of odd parity-checks that are in the neighborhood of each variable. The value of a variable is flipped if more odd parity-checks than even parity-checks are in its neighborhood. After the value of a variable has been flipped, all parity-checks are calculated again. This process is repeated until no such variable can be found.

The second decoder is the standard soft-decision iMPA. In short, the decoding process starts from the n variable vertices, where the initial symbol-level soft-decision channel information and messages [1] from connected parity-checks are available. Based on those provided information and the constraint that values of the bit inferred by the messages should be the same, the variables calculate *extrinsic information* [14] and pass them to the connected parity-checks. Similar processes are performed at the p parity-checks based on the constraint that module-2 sum of the bits inferred by the messages should be 0, and extrinsic information are passed back from a parity-check to its neighbors as well.

The process of passing messages from variables to the parity-checks and back from the parity-checks to the variables defines one iteration. This iterative processing is repeated

[1] During the first iteration of the decoding process, those messages are set to 0 if the processing is in minus-log metric domain or 1 if it is in the probability domain.

a fixed number of times and the final hard-decisions are made on the n variables. If the final decoded block is different from the transmitting one, a block error is declared. Unlike the Spielman's simple sequential decoding, a soft-decision iterative decoder does not terminate automatically.

5.2.3 The proposed criteria

Criterion 5.1. *For two different Tanner graphs, G_1 and G_2, representing the same binary linear code $\mathcal{C}$ of length n, both have n variables and p parity-checks and assume that the minimum distance of $\mathcal{C}$ is $d_{\min}$. If*

$$\delta_{\text{avg}}(m, G_1) \geq \delta_{\text{avg}}(m, G_2) \tag{5.2}$$

for every $1 \leq m \leq d_{\min}$, where $\delta_{\text{avg}}(m, G_1)$ and $\delta_{\text{avg}}(m, G_2)$ denote the average expansion for subsets of size m of G_1 and G_2 respectively, we claim that, with high probability, Spielman's simple sequential decoding on G_1 will outperform the one on G_2 in the sense that it can achieve lower block error rate. □

As Spielman's simple sequential decoding will terminate automatically, we compare the final achievable block error rate. Also, as not all error patterns beyond the minimum distance can not be corrected by maximum-likelihood (ML) decoding either, we can only consider average expansion of subset of the size no larger than $d_{\min}$.

Though we do not have a rigorous proof for this criterion, it be argued that this is correct with high probability. Considering a general Tanner graph $G_T = (X_n \cup Y_p, E)$, let S_m be a subset of X_n such that $|S_m| = m$. Without loss of generality, assuming that the

all-zero codeword is transmitted, and the received observations can be written in vector form as:

$$\mathbf{z} = \mathbf{x} + \psi_{S_m} = \mathbf{0} + \psi_{S_m} \tag{5.3}$$

where the $n \times 1$ column vectors $\psi_{S_m} = (\psi_0, \psi_1, ...\psi_{n-1})^T$, such that $\psi_j = 1$, if $x_j \in S_m$ and $\psi_j = 0$, otherwise, correspond to error patterns caused by the channel. Thus, the probability of block error is

$$P_e = P_r(\widetilde{\mathbf{x}} \neq \mathbf{0} | \mathbf{x} = \mathbf{0}) \tag{5.4a}$$

$$= 1 - P_r(\widetilde{\mathbf{x}} = \mathbf{0} | \mathbf{x} = \mathbf{0}) \tag{5.4b}$$

$$= 1 - \sum_{m=0}^{n} \sum_{S_m} \epsilon^m (1-\epsilon)^{n-m} P_r(\psi_{S_m} \text{ is a correctable error pattern}) \tag{5.4c}$$

$$\leq 1 - \sum_{m=0}^{d_{\min}} \sum_{S_m} \epsilon^m (1-\epsilon)^{n-m} P_r(\psi_{S_m} \text{ is a correctable error pattern}) \tag{5.4d}$$

where ϵ is the cross-over probability of the BSC.

Using the expander graph arguments [12, 55], we know that an error pattern ψ_{S_m} is correctable if the expansion of S_m is larger than some value, denoted as δ_{th} [2]. Therefore,

$$P_e \leq 1 - \sum_{m=0}^{d_{\min}} \sum_{S_m \,:\, \delta(S_m) > \delta_{th}} \epsilon^m (1-\epsilon)^{n-m} \tag{5.5}$$

because $\delta(S_m) > \delta_{th}$ is a sufficient condition that ψ_{S_m} corresponds to a correctable error pattern.

[2] It should be noted that this δ_{th} is determined by the decoding algorithm and the channel. For example, for erasure channel, $\delta_{th} = 1/2$; and for Spielman's simple sequential decoding on regular LDPC codes for the BSCs, $\delta_{th} = 3/4$.

As discussed in Section 3.3.2, previous work on this problem focused on the minimum expansion of subsets of variables. However, as suggested by (5.5), not only the expansion, but also the number of subsets with such expansion matters. Thus, the expansions of all subsets of size no larger than $d_{\min}$ can be computed, an upper bound on the block error rate can be obtained. For two different Tanner graphs $G_1 = \{X_n^{(1)} \bigcup Y_p^{(1)}, E^{(1)}\}$ and $G_2 = \{X_n^{(2)} \bigcup Y_p^{(2)}, E^{(2)}\}$ representing the same binary linear $[n, k, d_{\min}]$ code $\mathcal{C}$, and the same well defined δ_{th}, if it can be shown that, for ever $1 \leq m \leq d_{\min}$, the number of size m subsets of $X_n^{(1)}$ with expansion larger than δ_{th} is larger than the number of size m subsets of $X_n^{(2)}$ with expansion larger than δ_{th}, we can claim that the iterative decoder on G_1 will perform better with high confidence.

Unfortunately, exhaustively calculating expansions of all subsets is hardly possible for practical communication systems, when n is usually very large. However, we can argue that, with high probability, if $\delta_{\text{avg}}(m, G_1) \geq \delta_{\text{avg}}(m, G_2)$ then the number of size m subsets of $X_n^{(1)}$ with expansion larger than δ_{th} is larger than the number of size m subsets of $X_n^{(1)}$ with expansion larger than δ_{th}. The reason for this is that, for randomly generated parity-check matrices, it is conceivable that the distribution of δ_{S_m} concentrates on its average value.

Though we have not been able to prove Criterion 5.1, we believe that it suffices to take two steps to prove it. First, we may want to show that, for a given ϵ, approximation of P_e using (5.5) will be sufficiently close to the real block error rate. Then, we may want to show, for subsets of size m, the concentration of their expansions around the average expansion. It should be noted that similar methods have been used by several

researchers [31, 32, 46, 47] to derive the evolution of messages. Therefore, it is reasonable to believe this would work for our criterion as well.

As the noises presented in practical communication systems are usually modelled as zero-mean, white and Gaussian, our second criterion is summarized in the following.

Criterion 5.2. *For two different Tanner graphs, G_1 and G_2, representing the same binary linear code $\mathcal{C}$ of length n, both have n variables and p parity-checks and assume that the minimum distance of $\mathcal{C}$ is $d_{\min}$. If*

$$\delta_{\text{avg}}(m, G_1) \geq \delta_{\text{avg}}(m, G_2) \tag{5.6}$$

for every $1 \leq m \leq d_{\min}$, where $\delta_{\text{avg}}(m, G_1)$ and $\delta_{\text{avg}}(m, G_2)$ denote the average expansion for subsets of size m of G_1 and G_2 respectively, we claim that, with high probability, the standard soft-decision iMPA on G_1 will outperform the one on G_2 in the sense that it can achieve lower block error rate for the same number of iterations. □

Again, we do not have a rigorous proof for this claim, and we believe it would be much harder to prove it than to prove Criterion 5.1. Thus, we think this should be more properly considered as a heuristic.

At the end, we would like to present one example to argue the correctness of the criteria. Considering the $[15, 7, 5]$ cyclic BCH code as an example, it has parity-check matrix in cyclic form as

$$\mathbf{H}_{\text{cyc}} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.7}$$

and parity-check matrix in systematic form as

$$\mathbf{H}_{\text{sys}} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.8}$$

m	1	2	3	4	5
$\delta_{\text{avg}}(m, G_{\text{cyc}})$	1.0	0.91	0.81	0.72	0.64
$\delta_{\text{avg}}(m, G_{\text{sys}})$	1.0	0.90	0.79	0.69	0.60

Table 5.1: Average expansions of $[15, 7, 5]$ cyclic BCH code

Let G_{cyc} and G_{sys} be the Tanner graphs correspond to $\mathbf{H}_{\text{cyc}}$ and $\mathbf{H}_{\text{sys}}$ respectively, and Table 5.1 contain results of for the average expansions of both cases as a function of m. It can be seen from Table 5.1 that $\delta_{\text{avg}}(m, G_{\text{cyc}}) > \delta_{\text{avg}}(m, G_{\text{sys}})$ for every $1 \leq m \leq 5 = d_{\min}$, thus Criterion 5.2 would suggest that the iMPA on G_{cyc} have a lower block error rate. This has been verified using Figure 5.2. From left to right, the two curves are the block error rate for iMPA on G_{cyc}, and the block error rate for iMPA on G_{sys} respectively, where the and x-axis and y-axis correspond to E_b/N_0 and block error rate respectively.

5.2.4 Applying Theorem 3.3

In the previous section, we have provided two criteria suggesting that there may be relations between the average variable expansions of Tanner graphs and the performance of iterative decoding algorithms on them. However, calculating average variable expansions is not an easy task either. Thus, it is also conjectured that we can use the lower bounds on the average expansion, $\delta_{\text{avg}}(m)$, which was derived in Theorem 3.3. Again, use the $[15, 7, 5]$ cyclic BCH code as an example, it can be seen from Table 5.2 that the G_{cyc} has larger lower bound for every $1 \leq m \leq 5 = d_{\min}$, the performance of soft-decision iMPA on G_{cyc} is indeed better.

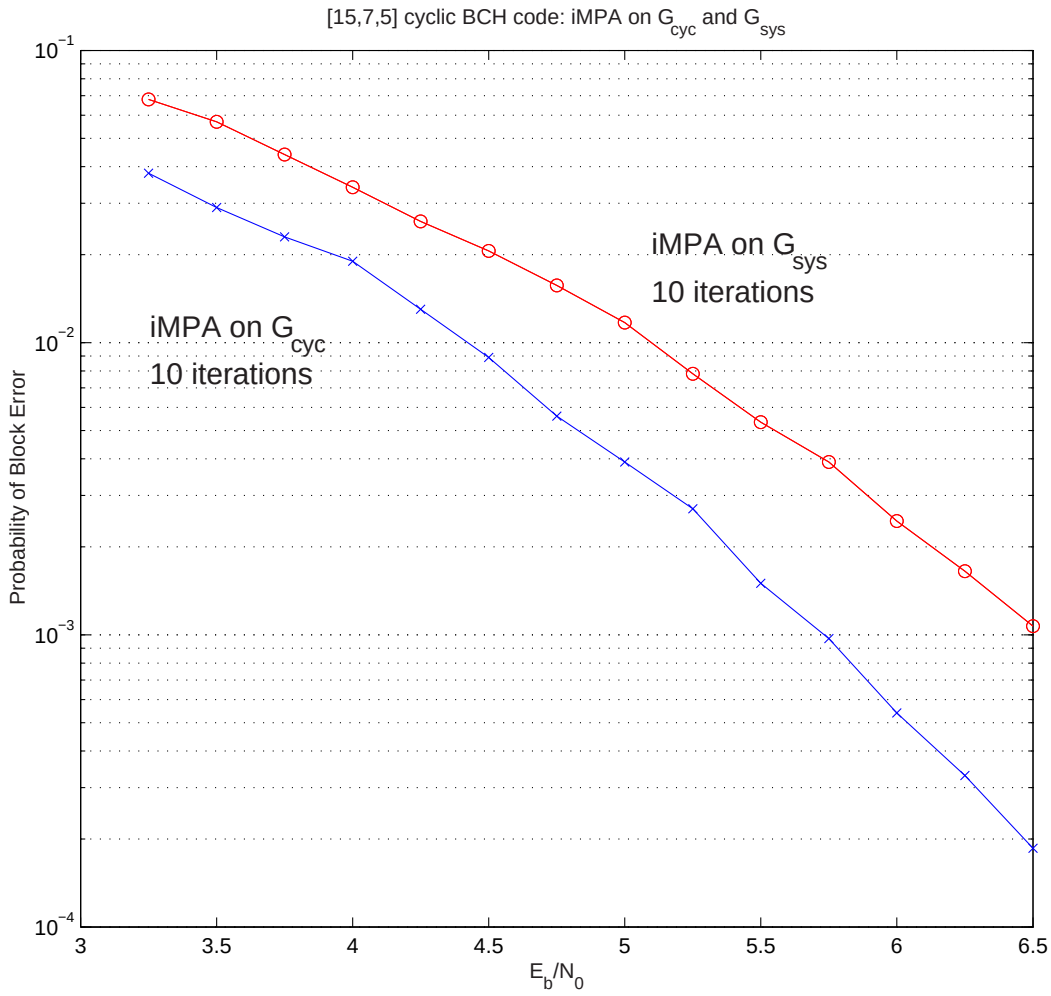

Figure 5.2: Block error rate of standard soft-decision iMPA on G_{cyc} and G_{sys} of $[15, 7, 5]$ cyclic BCH code

m	1	2	3	4	5
Lower Bound on $\delta_{\text{avg}}(m, G_{\text{cyc}})$ using Theorem 3.3	0.94	0.80	0.69	0.60	0.53
Lower Bound on $\delta_{\text{avg}}(m, G_{\text{sys}})$ using Theorem 3.3	0.70	0.59	0.50	0.43	0.38

Table 5.2: Lower bound on $\delta_{\text{avg}}(m)$ of $[15, 7, 5]$ cyclic BCH code

5.3 Summary

Two criteria have been provided in this chapter trying to connect the performance of iterative message-passing algorithms with the expansion property of the underlying graphical representations. Specifically, we have considered Spielman's simple sequential decoding for BSCs and the standard soft-decision algorithm for AWGN, and argued that these criteria are true with high probability. An example was also provided.

However, we have not been able to provide a rigorous proof for these criteria. It would be of great interest to prove these criteria, either theoretically or heuristically via more empirical evidence . Section 5.2.3 contains a short discuss suggesting a possible proof of Criterion 5.1.

Chapter 6

Conclusion and Future work

There are two main contributions of this work. First, we applied the well-known iterative decoding techniques to the code acquisition problem, which existed in the literature for a long time. As the applications of traditional methods are usually limited, we found, mostly by computer simulations, that the iterative MPA approach provides a promising solution to the code acquisition problem. Specifically, our approach approximate the full parallel search with the same order of acquisition time, but with significantly less complexity. This approach seems also promising to solve the PN code acquisition problems for the low duty cycle UWB systems, where extremely fast code acquisition is critical.

We also tried to use the techniques of eigenvalue analysis to relate the performance of iterative decoders on loopy Tanner graphs to the expansion properties of these graphs. Similar work has been done by many researchers for the binary erasure channels, and our approach can provide interesting results related to theirs. For more general binary symmetric channels and additive white Gaussian channels, we proposed two criteria suggesting the possible connections between the average variable expansions of Tanner graphs and the performance of associated iMPAs. However, as average variable expansions is not

easy to compute either. A practical heuristic was proposed as well by deriving an lower bound on the average variable expansions first, then suggested that it could be used as a replacement of the real average expansion.

It is believed by us that, in both cases, we have only scratched the surface of the problems. However, these two problems are closely related. The code acquisition motivates us to find good graphical representation, which is the problem being addressed by the eigenvalue analysis. Using the eigenvalue analysis and if the two criteria are true with high probability, we can find good graphical representation for the PN code acquisition problems.

Among the open problems, the most important ones will be to prove the correctness, or the confidence of the claims, of the criteria, either theoretically or heuristically. For Criterion 5.1, we have pointed out a possible route to reach the proof. For Criterion 5.2, though similar argument may still be true, we believe heuristic computer simulations may be much practical solution.

Also, we have derived lower bounds on the stopping distance and used them to derive an upper bound on the stopping redundancy of the difference-set codes. However, we believe that the applications of those results may be beyond that, and bounds on the stopping redundancy of other family of binary linear codes are interesting research problems as well. For example, in Appendix B, we provide a weak upper bound on the stopping redundancy of the family of one-step majority-logic decodable code, and these bounds may be significantly improved if more algebraic structures of the codes are brought into considerations.

Bibliography

[1] D. Agrawal. *GMD decoding of Euclidean-space codes and iterative decoding of turbo codes*. PhD thesis, University of Illinois at Urbana-Champaign, Urbana, IL, 1999.

[2] S. M. Aji. *Graphical Models and Iterative Decoding*. PhD thesis, California Institute of Technology, 1999.

[3] S. M. Aji and R. J. McEliece. The generalized distributive law. *IEEE Trans. Information Theory*, 46(2):325–343, March 2000.

[4] S. M. Aji and R. J. McEliece. The generalized distributive law and free enery minimization. In *Proc. Allerton Conf. Commun., Control,Comp.*, October 2001.

[5] W. K. Alem and C. L. Weber. Acquisition techniques of PN sequences. In *1977 NTC COnference Record*, pages 35:2–1–35:2–4, Los Angeles, CA, October 1977.

[6] S. Benedetto and G. Montorsi. Design of parallel concatenated convolutional codes. *IEEE Trans. Commununication*, 44:591–600, May 1996.

[7] S. Benedetto and G. Montorsi. Unveiling turbo codes: Some results on parallel concatenated coding schemes. *IEEE Trans. Information Theory*, 42:409–428, March 1996.

[8] C. L. Bennett and G. F. Ross. Time-domain electromagnetics and its applications. *Proc. IEEE*, 66, March 1978.

[9] E. Berlekamp, R. McEliece, and H. van Tilborg. On the inherent intractablity of certain coding problems. *IEEE Trans. Information Theory*, IT-24:384–386, May 1978.

[10] C. Berrou and A. Glavieux. Near optimum error correcting coding and decoding: turbo-codes. *IEEE Trans. Commununication*, 44(10):1261–1271, October 1996.

[11] C. Berrou, A. Glavieux, and P. Thitmajshima. Near shannon limit error-correcting coding and decoding: turbo-codes. In *Proc. International Conf. Communications*, pages 1064–1070, Geneva, Switzerland, May 1993.

[12] D. Burshtein and G. Miller. Expander graph arguments for message-passing algorithms. *IEEE Trans. Information Theory*, 47:782–790, February 2001.

[13] Xiaopeng Chen. *Iterative Data Detection: Complexity Reduction and Applications*. PhD thesis, University of Southern California, Los Angeles, CA, December 1999.

[14] K. M. Chugg, A. Anastasopoulos, and X. Chen. *Iterative Detection: Adaptivity, Complexity Reduction, and Applications*. Kluwer Academic Publishers, 2001.

[15] K. M. Chugg and M. R. Zhu. A new approach to rapid PN code acquisition using iterative message passing techniques. *IEEE J. Select. Areas Commun.*, pages 884–897, May 2005.

[16] F. R. K. Chung. *Spectral Graph Theory*. American Mathematical Society, 1997.

[17] C. Di, D. Proietti, I. E. Telatar, T. J. Richardson, and R. L. Urbanke. Finite-length analysis of low-density parity-check codes on the binary erarue channel. *IEEE Trans. Information Theory*, 48:1570 – 1579, June 2002.

[18] R. E. Elias. Coding for two noisy channels. In *Information Theory, 3rd London Symp.*, pages 61–76, 1955.

[19] G. D. Forney, Jr. Codes on graphs: normal realizations. *IEEE Trans. Information Theory*, 47:520–548, February 2001.

[20] G. D. Forney, Jr., R. Koetter, F. R. Kschischang, and A. Reznik. On the effecctive wieghts of pseufocodewords for codes defined on graphs with cycles. *Codes, Systems and Graphical Models (Minneapolis, MN, 1999) (B. Marcus and J. Rosenthal, eds.)*, 123 of IMA VOl. Math. Appl., 2001.

[21] R. G. Gallager. Low density parity check codes. *IEEE Trans. Information Theory*, 8:21–28, January 1962.

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

[23] Solomon W. Golomb. *Shift Register Sequences, revised edition*. Aegean Park Press, 1982.

[24] T. R. Halford and K. M. Chugg. On efficiently counting short cycles in bipartite graphs. *IEEE Trans. Information Theory. to appear in 2006*.

[25] X. Y. Hu, E. Elfrheriou, and D. M. Arnold. Progressive edge-growth tanner graphs. In *Proc. Globecom Conf.*, pages 995–1001, November 2001.

[26] C. C. Kilgus. Pseudonoise code acquisition using majority logic decoding. *IEEE Trans. Commununication*, 21:772–774, June 1973.

[27] Y. Kou, S. Lin, and P. C. Fossorier. Low-density parity-check codes based on finite geometries: A rediscoverty and new results. *IEEE Trans. Information Theory*, 47: 2711–2736, November 2001.

[28] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Trans. Information Theory*, IT-47, February 2001.

[29] F.R. Kschischang and B.J. Frey. Iterative decoding of compond codes by probability propagation in graphical models. *IEEE J. Select. Areas Commun.*, pages 219–231, February 1998.

[30] S. Lin and Jr. D. Costello. *Error Control Coding: Fundamentals and Applications*. Prentice-Hall, 1983.

[31] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman. Efficient erasure correcting codes. *IEEE Trans. Information Theory*, IT-47, February 2001.

[32] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman. Improved low-density parity-check codes using irregular graphs. *IEEE Trans. Information Theory*, IT-47, February 2001.

[33] R. Lucas, M. Fossorier, Y. Kou, and S. Lin. Iterative decoding of one-step majority logic decodable codes based on belief propagation. *IEEE Trans. Commununication*, 43:354–364, Feb–Apr. 1995.

[34] D. J. C. MacKay and R. M. Neal. Near Shannon limit performance of low density parity check codes. *IEE Electronics Letters*, 32(18):1645–1646, August 1996.

[35] R. J. McEliece, D. J. C. MacKay, and J. F. Cheng. Turbo decoding as an instance of Pearl's "belief propagation" algorithm. *IEEE J. Select. Areas Commun.*, 16:140–152, February 1998.

[36] I. D. O'Donnell, S. W. Chen, B. T. Wang, and R. W. Brodersen. An integrated, low power, ultra-wideband reansceriver architecture for low-rate, indoor wireless systems. *IEEE CAS Workshop on Wireless Communications and Networking*, September 2002.

[37] A. Orlitsky, K. Viswananathan, and J. Zhang. Stopping set distribution of low-density parity-check code ensembles. *IEEE Trans. Information Theory*, 51:929 – 953, March 2005.

[38] H. M. Pearce and M. P. Ristenblatt. The threshold decoding estimator for synchonization with binary linear recursive sequences. In *Proc. International Conf. Communications*, pages 43–25–43–30, Montreal, Canada, June 1971.

[39] J. Pearl. Fusion, propagation, and structuring in belief networks. *Artif. Intell.*, 29 (3):241–288, September 1986.

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

[41] L. Perez, J. Seghers, and D. Costello. A distance spectrum interpretation of turbo codes. *IEEE Trans. Information Theory*, 42:1698–1709, November 1996.

[42] Roger L. Peterson, Rodger E. Ziemer, and David E. Borth. *Introduction to Spread-Spectrum Communications*. Prentice-Hall, 1995.

[43] H. Pishro-Nik and F. Fekri. On decoding of low-density parity-check codes over the binary erasure channel. *IEEE Trans. Information Theory*, 50:439 – 454, March 2004.

[44] A. Polydoros and C. L. Weber. A unified approach to serial search spread-spectrum code acquisition. *IEEE Trans. Commununication*, vol. 32:542–560, May 1984.

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

[46] T. J. Richardson, M. A. Schokrollahi, and R. L. Urbanke. Desing of capacity-approaching irregular low-density parity-check codes. *IEEE Trans. Information Theory*, IT-47, February 2001.

[47] T. J. Richardson and R. L. Urbanke. The capacity of low-density parity-check codes under message-passing decoding. *IEEE Trans. Information Theory*, IT-47, February 2001.

[48] G. F. Ross. The transient analysis of certain TEM mode four-post networks. *IEEE Tran. Microwave Theory Tech.*, MTT-14, November 1966.

[49] G. F. Sage. Serial synchonization of psuedonoise systems. *IEEE Trans. Commununication*, 12:123–127, December 1965.

[50] R. A. Scholtz. Multiple access with time-hopping impulse modulation. In *Proc. IEEE Military Comm. Conf.*, 1993.

[51] M. Schwartz and A. Vardy. On the stopping distance and the stopping redundancy of codes. *Submitted to Information Theory*, 2005.

[52] M. H. Shin, J. S. Kim, and H. Y . Song. Minimum distance bounds of irregular qc-ldpc codes and their applications. In *Proc. IEEE Symposium on Information Theory*, page 311, 2004.

[53] M. H. Shin, J. S. Kim, and H. Y. Song. Generalization of tanner's minimum distance bounds for ldpc codes. *IEEE Communications Letters*, 9:240–242, March 2005.

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

[55] M. Sipser and D. A. Spielman. Expander codes. *IEEE Trans. Information Theory*, 42:1710 – 1722, November 1996.

[56] Gilbert Strang. *Linear Algebra and Its Applications*. Harcourt Brace Jovanovich College Publishers, 1986.

[57] J. S.Yedidia, W. T. Freeman, and Y. Weiss. Bethe free energy, kikuchi approximations and belief propagation algorithms. *Mitsubishi Electric Research Laboratories*, 2001. (avilable at www.merl.com/papers/TR2001-16/).

[58] J. S.Yedidia, W. T. Freeman, and Y. Weiss. Construction free energy approximations and generalized belief propagation algorithms. *Mitsubishi Electric Research Laboratories*, 2002. (avilable at www.merl.com/papers/TR2002-35/).

[59] R. M. Tanner. A recursive approach to low complexity codes. *IEEE Trans. Information Theory*, 27:533–547, September 1981.

[60] R. M. Tanner. Minimum-distance bounds by graph analysis. *IEEE Trans. Information Theory*, 47:808–820, February 2001.

[61] Durai. P. Thirupathi. *Synchronization in wide band systems with coding*. PhD thesis, University of Southern California, Los Angeles, CA, December 2004.

[62] A. Vardy. The intractability of computing the minimum distance of a code. *IEEE Trans. Information Theory*, IT-43:1757 – 1766, November 1997.

[63] A. Vardy. Which codes have cycle-free Tanner graphs. *IEEE Trans. Information Theory*, IT-45:2173 – 3006, September 1999.

[64] R. Vigoda, J. Dauwels, N. Gershenfeld, and H. A. Loeliger. Low-complexity lfsr synchronization by forward-only message passing. *IEEE Trans. Information Theory*, 2003. (submitted).

[65] P. O. Vontobel and R. Koetter. Lower bounds on the minimum pseudo-weight of linear codes. In *Proc. IEEE Symposium on Information Theory*, page 70, June 2004.

[66] P. O. Vontobel and R. Smarandache. On minimal pseudo-codewords of Tanner graphs from projective planes. In *Proc. Allerton Conf. Commun., Control,Comp.*, September 2005.

[67] P. O. Vontobel, R. Smarandache, N. Kiyavash, J. Teutsch, and D. Vukobratovic. On the minimal pseudo-codewords of codes from finite geometries. In *Proc. IEEE Symposium on Information Theory*, June 2005.

[68] R. B. Ward. Acquisition of pseudonoise signals by sequential estimation. *IEEE Trans. Commununication*, 13:475–483, December 1965.

[69] R. B. Ward and K. P. Yiu. Acquisition of pseudonoise signals by recursion aided sequential estimation. *IEEE Trans. Commununication*, 25:784–794, August 1977.

[70] N. Wiberg. *Codes and Decoding on General Graphs*. PhD thesis, Linköping University (Sweden), 1996.

[71] N. Wiberg, H.-A. Loeliger, and R. Kötter. Codes and iterative decoding on general graphs. In *Proc. IEEE Symposium on Information Theory*, page 468, 1995.

[72] M. Z. Win. *Ultra-Wide Bandwidth Spread-Spectrum Techniques for Wireless Multiple-Access Communications*. PhD thesis, usc, 1998.

[73] L. L. Yang and L. Hanzo. Iterative soft sequential estimation assisted acqusition of m-sequences. *IEE Electronics Letters*, pages 1550–1551, November 2002.

[74] L. L. Yang and L. Hanzo. Acquisition of m-sequences using recursive soft sequential estimation. *IEEE Trans. Commununication*, pages 199–204, February 2004.

[75] O. W. Yeung and K. M. Chugg. An iterative algorithm and low complexity hardwar architecture for fast acquisition of long PN codes in UWB systems. *Journal of VLSI Signal Processing*, 2005. (accepted).

[76] M. Zhu and K. M. Chugg. Iterative message passing techniques for rapid code acquisition. In *Proc. IEEE Military Comm. Conf.*, October 2003.

[77] M. R. Zhu and K. M. Chugg. Lower bounds on stopping distance and their applications. In *Proc. Allerton Conf. Commun., Control,Comp.*, September 2005.

Appendix A

Iterative MPA on Figure 2.2 (d)

In this appendix, we clarify the messages passed along edges in Figure 2.2(d). A check node from Figure 2.2(d) is redrawn in Figure A.1. 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 Figure A.1, specific labels are given to messages passed along these edges. The chip-level soft-decision channel information is

$$\mathrm{M}_{ch}[x_k] = \frac{2\sqrt{E_c} z_k (-1)^{x_k}}{N_0} \qquad x_k = 0, 1 \tag{A.1}$$

where z_k is the relevant real part of the observation in (2.3) and the local configuration metric is

$$\mathrm{M}[\tau_k] = \mathrm{F}_k[\sigma_k] + \mathrm{RI}_k[x_k] + \mathrm{LI}_k[x_{k-15}] + \mathrm{B}_{k+1}[\sigma_{k+1}] \tag{A.2}$$

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 (A.2), a compact way of expressing the message updating in min-sum form is[1]:

$$\begin{aligned}
\mathrm{F}_{k+1}[\sigma_{k+1}] &= \min\nolimits_{\tau_k:\sigma_{k+1}} \mathrm{M}[\tau_k] - \mathrm{B}_{k+1}[\sigma_{k+1}] && \sigma_{k+1} = 0, 1 && \text{(A.3)}\\
\mathrm{B}_k[\sigma_k] &= \min\nolimits_{\tau_k:\sigma_k} \mathrm{M}[\tau_k] - \mathrm{F}_k[\sigma_k] && \sigma_k = 0, 1 && \text{(A.4)}\\
\mathrm{LO}_k[x_{k-15}] &= \min\nolimits_{\tau_k:x_{k-15}} \mathrm{M}[\tau_k] - \mathrm{LI}_k[x_{k-15}] && x_{k-15} = 0, 1 && \text{(A.5)}\\
\mathrm{RO}_k[x_k] &= \min\nolimits_{\tau_k:x_k} \mathrm{M}[\tau_k] - \mathrm{RI}_k[x_k] && x_k = 0, 1 && \text{(A.6)}\\
\mathrm{MO}[x_k] &= \mathrm{LO}_{k+15}[x_k] + \mathrm{RO}_k[x_k] && x_k = 0, 1 && \text{(A.7)}\\
\mathrm{LI}_k[x_{k-15}] &= \mathrm{RO}_{k-15}[x_{k-15}] + \mathrm{M}_{ch}[x_{k-15}] && x_{k-15} = 0, 1 && \text{(A.8)}\\
\mathrm{RI}_k[x_k] &= \mathrm{LO}_{k+15}[x_k] + \mathrm{M}_{ch}[x_k] && x_k = 0, 1 && \text{(A.9)}
\end{aligned}$$

Similarly, min*-sum messages can be obtained by replacing min operators in equations (A.2)-(A.6) by min*.

[1]Since the term subtracted in each of (A.3)-(A.9) is constant over all terms in the minimization, each equation can be written in a form where (A.2) is simplified by cancelling that term priori to minimization.

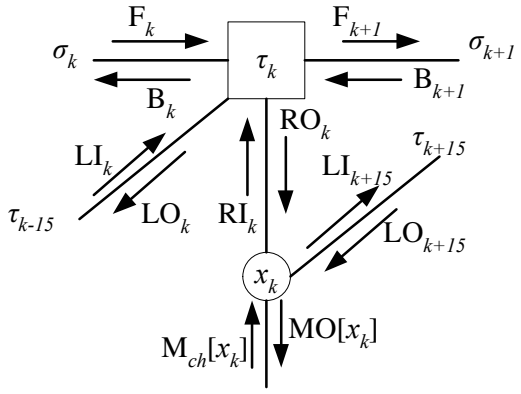

Figure A.1: Detailed notation of the input and output messages associated with one check node in Figure 2.2(d).

Pseudo-code of the proposed iMPA algorithm over Figure 2.2(d)

1. Initialization: $\mathrm{F}_0[\sigma_0], \mathrm{B}_M[\sigma_M], i \leftarrow 0, I \leftarrow$ Maximum Number of iterations;

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

2. Forward-backward algorithm: updating $\mathrm{F}_k[\sigma_k]$ and $\mathrm{B}_k[\sigma_k]$, $0 \leq k \leq M-1$, sequentially using (A.3) and (A.4) respectively. $\mathrm{F}_0[\sigma_0] \rightarrow \mathrm{F}_1[\sigma_1] \rightarrow ...\mathrm{F}_k[\sigma_k]... \rightarrow \mathrm{F}_M[\sigma_M]$; $\mathrm{B}_M[\sigma_M] \rightarrow ...\mathrm{B}_{k+1}[\sigma_{k+1}] \rightarrow \mathrm{B_k}[\sigma_k]... \rightarrow \mathrm{B}_0[\sigma_0]$.

3. Update $\mathrm{LO}_k[x_{k-15}]$ and $\mathrm{RO}_k[x_k]$: $0 \leq k \leq M-1$, using equation (A.5) and (A.6) respectively. Then, $i \longleftarrow i+1$, $\mathrm{LI}_k[x_{k-15}]$ and $\mathrm{RI}_k[x_k]$ are updated using equation (A.8) and (A.9) respectively.

4. Selecting candidate decisions: $\lfloor M/15 \rfloor$ non-overlap (intermediate) estimates of the initial state are obtained using $\mathrm{M}_k[x_k] = \mathrm{M}_{ch}[x_k] + \mathrm{MO}[x_k]$, $15i \leq k \leq 15i+14$, $i = 0, 1, .., \lfloor M/15 \rfloor$. 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.

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.

Appendix B

Upper bounds on the Stopping Redundancy of the one-step MLD codes

The goal of this appendix is to provide an upper bound on the stopping redundancy of the class of the one-step majority-logic decodable (MLD) codes.

B.1 One-step majority-logic decodable codes

For a given binary matrix $\mathbf{H}$, suppose that there exist I vectors in its row space,

$$\mathbf{w}_1 = (w_{10}, w_{11}, ..., w_{1,n-1})$$

$$\mathbf{w}_2 = (w_{20}, w_{21}, ..., w_{2,n-1})$$

$$. \quad .$$

$$\mathbf{w}_I = (w_{I0}, w_{I1}, ..., w_{I,n-1})$$

such that:

- $w_{1,n-1} = w_{2,n-1} = ...w_{I,n-1} = 1$
- For $j \neq n-1$, there exist at most on vector whose i-th component is a "1".

These I vectors are said that be orthogonal on the $(n-1)$-th digit. We call them *orthogonal vectors* [30].

Definition B.1. *[30][Ch.7] A cyclic code with minimum distance d_{min} is said to be completely orthogonalizable in one step if and only if it is possible to form $I = d_{min} - 1$ parity-checks orthogonal on an error digit.*

Definition B.2. *[30][Ch.7] A cyclic code is one-step majority-logic decodable (MLD) if it is completely orthogonalizable in one step.*

It should be noted that the majority-logic decoding was first introduced as an effective scheme for decoding certain classes of block codes. Therefore, in strict sense, every linear block code is one-step majority-logic decodable. However, it can be shown that one-step majority-logic decoding is most effective for cyclic codes that are completely orthogonalizable in one step. The term one-step majority-logic decodable code is usually

reserved for the class of codes that are completely orthogonalizable in one step only. Details of the majority-logic decoding and the class of majority-logic decodable codes can be found in [30][Ch.7,Ch.8]

B.2 Upper Bounds on the Stopping Redundancy of the one-step MLD codes

Referred to Section 3.1, for an $[n, k, d_{min}]$ linear code $\mathcal{C}$ specified by a $p \times n$ parity-check matrix $\mathbf{H}$, where $p \geq n - k$, the corresponding bipartite graph G_T is denoted as:

$$G_T = (B \cup Y, E) = (\{b_1, b_2, ..., b_n\} \cup \{y_1, y_2, ..., y_p\}, E) \tag{B.1}$$

It can be shown that, if we define

$$s(b_j, \mathbf{H}) = \text{size of the smallest stopping sets include } b_j \quad 1 \leq j \leq n \tag{B.2}$$

then the stopping distance of the graph

$$s(\mathbf{H}) = \min_j s(b_j, \mathbf{H}) \tag{B.3}$$

If we further know that $\mathcal{C}$ is an one-step majority-logic decodable code, there exists a $I \times n$ matrix

$$\mathbf{H}_0 = \begin{bmatrix} 1 & h_{11} & h_{12} & \cdot & h_{1,n-1} \\ 1 & h_{21} & h_{22} & \cdot & h_{2,n-1} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & h_{I1} & h_{I2} & \cdot & h_{I,n-1} \end{bmatrix} \tag{B.4}$$

which has row vectors orthogonal on the 0-th digit and $I = d_{min} - 1$. It can be shown that the j-th, $0 \leq j \leq n-1$ cyclic shift of $\mathbf{H}_0$ to the right

$$\mathbf{H}_0 = \begin{bmatrix} h_{1,n-j} & \cdot & h_{1,n-1} & 1 & h_{11} & \cdot & h_{1,n-j-1} \\ h_{2,n-j} & \cdot & h_{2,n-1} & 1 & h_{21} & \cdot & h_{2,n-j-1} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ h_{I,n-j} & \cdot & h_{I,n-1} & 1 & h_{I1} & \cdot & h_{I,n-j-1} \end{bmatrix} \tag{B.5}$$

contain the I row vectors orthogonal to the j-th digit. Then, we have the following theorem.

Theorem B.1. *The stopping redundancy of one-step majority-logic decodable codes is less than or equal to $(d_{min} - 1)n + n - k$, where n and d_{min} are the length, the dimension and minimum distance of the code respectively.*

Proof of Theorem B.1. *This theorem can be proved by using the redundant parity-check matrix,*

$$\widetilde{\mathbf{H}} = \begin{bmatrix} \mathbf{H} \\ \mathbf{H}_0 \\ \cdot \\ \mathbf{H}_i \\ \cdot \\ \mathbf{H}_{n-1} \end{bmatrix} \tag{B.6}$$

where $\mathbf{H}_j$*'s are defined in* (B.5)*. It can be shown that*

$$s(b_j, \widetilde{\mathbf{H}}) \geq I + 1 \quad 0 \leq \ j \leq \ n-1 \tag{B.7}$$

$$\Rightarrow s(\widetilde{\mathbf{H}}) = \min_j s(b_j, \widetilde{\mathbf{H}}) \geq I + 1 = d_{min} \tag{B.8}$$

However, since stopping distance is always no larger than d_{min}*,*

$$s(\widetilde{\mathbf{H}}) = d_{min} \tag{B.9}$$

$$\rho(\mathbf{C}) \leq \textit{number of rows in } \widetilde{\mathbf{H}} = I \cdot n + p = (d_{min} - 1) \cdot n + n - k \tag{B.10}$$

Bounds in Theorem B.1 can be improved a little by noting the fact, to make $\widetilde{\mathbf{H}}$ a valid parity-check matrix for the code, not all rows of $\mathbf{H}$ are needed. Since the I rows in $\mathbf{H}_0$ are linearly independent, we can safely remove I properly selected rows from the $\mathbf{H}$ in $\widetilde{\mathbf{H}}$ and still make row space of $\widetilde{\mathbf{H}}$ unchanged. Therefore, the improved upper bound

Theorem B.2.

$$\rho(\mathcal{C}) \leq (d_{min} - 1)n + n - k - d_{min} + 1 \tag{B.11}$$

where $\mathcal{C}$ *is an one-step majority-logic decodable code of length* n*, dimension* k *and minimum distance* d_{min}*.*

Consider the family of Reed-Muller codes as examples. Our Theorem B.1 applies since Reed-Muller codes are one-step majority-logic decodable codes. Also, Schwartz and Vardy [51] have provided an upper bound for Reed-Muller codes, specifically, for μ-th order RM code $RM(\mu, m)$ with parameters $n = 2^m - 1$, $k = \sum_{i=0}^{\mu} \begin{pmatrix} m \\ i \end{pmatrix}$, $d_{min} = 2^{m-\mu} - 1$, the upper bound is

$$\rho(\mathbf{C}) \leq d_{min} \cdot k/2 \tag{B.12}$$

Results are summarized in Table B.1. Unfortunately, our upper bound is very weak as compared to Schwartz and Vardy's work. However, this phenomenon does make sense as our work is on for a general one-step majority-logic decodable without any knowledge of the algebraic structure of the code.

$R(\mu, 7)$	$R(1,7)$	$R(2,7)$	$R(3,7)$	$R(4,7)$	$R(5,7)$
k	8	29	64	99	120
d_{min}	63	31	15	7	3
Schwartz-Vardy bound (B.12)	252	450	480	350	180
bound (B.11)	7921	3878	945	616	259

Table B.1: Upper bounds on stopping redundancy of the Reed-Muller codes