UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

ADAPTIVE CHANNEL AND SOURCE CODING USING APPROXIMATE INFERENCE

A DISSERTATION

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

Degree of

DOCTOR OF PHILOSOPHY

By

SHUANG WANG
Norman, Oklahoma
2011

ADAPTIVE CHANNEL AND SOURCE CODING
USING APPROXIMATE INFERENCE

A DISSERTATION APPROVED FOR THE
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

BY

Dr. Samuel Cheng, Chair

Dr. Pramode Verma

Dr. James Sluss

Dr. J. R. Cruz

Dr. William Ray

Acknowledgments

I would like to express my sincere gratitude to my advisor, Dr. Samuel Cheng, who has provided me valuable guidance and encouragement in my research work. His impressive patience and insightful instructions help me grow. Moreover, I would like to take this opportunity to thank Dr. Pramode Verma, Dr. James Sluss, Dr. J.R. Cruz and Dr. William Ray for serving on my committee and their kind review and suggestions of my dissertation.

I would extend my thanks to all those whom I have worked with in the past years. I want to thank Dr.Lina Stankovic and Dr.Vladimir Stankovic from the University of Strathclyde. I have benefited a lot from the joint work with them about distributed video/image coding. My sincere appreciation also goes to Dr. Qiang Wu, the leader of Soft Imaging LLC, for offering the internships to study and work in his company during the summer 2008 and 2009. Moreover, I am grateful to Dr. Xiaoqian Jiang, with whom I worked during the summer 2011 at UCSD. His strong presentation skills, insightful comments and suggestions help me to work out problems during the intern.

I want to thank all my group mates in our research group set up by Dr. Samuel Cheng. All my group mates were sources of learning, knowledge, inspiration, and friendship. It was sheer enjoyment to learn and discuss knowledge, share research life with them, Mouhammad Al-Akkoumi, Amin, Nafise, Siddharth, Bei Li, Anusha, Lijuan Cui. In addition, I want to give my special thanks to my friends for their help and being with me. Since arriving at OU, I have had friendship, support and encouragement from all the friends, old and new.

My deepest gratitude is for my family. I would like to express my heartfelt gratitude to my parents, Fuchen Wang and Suming Zhao. They raised me, supported me, taught me, and loved me. Furthermore, I wish to thank my wife, Lijuan Cui.

She accompanied me, helped me and support me in all the past years. To them, I dedicate this dissertation.

Table of Contents

Chapter Page

List of Figures

Figure **Page**

Abstract

Channel coding and source coding are two important problems in communications. Although both channel coding and source coding (especially, the distributed source coding (DSC)) can achieve their ultimate performance by knowing the perfect knowledge of channel noise and source correlation, respectively, such information may not be always available at the decoder side. The reasons might be because of the time-varying characteristic of some communication systems and sources themselves, respectively. In this dissertation, I mainly focus on the study of online channel noise estimation and correlation estimation by using both stochastic and deterministic approximation inferences on factor graphs.

In channel coding, belief propagation (BP) is a powerful algorithm to decode low-density parity check (LDPC) codes over additive white Gaussian noise (AWGN) channels. However, the traditional BP algorithm cannot adapt efficiently to the statistical change of SNR in an AWGN channel. To solve the problem, two common workarounds in approximate inference are stochastic methods (e.g. particle filtering (PF)) and deterministic methods (e.g. expectation approximation (EP)). Generally, deterministic methods are much faster than stochastic methods. In contrast, stochastic methods are more flexible and suitable for any distribution. In this dissertation, I proposed two adaptive LDPC decoding schemes, which are able to perform online estimation of time-varying channel state information (especially signal to noise ratio (SNR)) at the bit-level by incorporating PF and EP algorithms. Through experimental results, I compare the performance between the proposed PF based and EP based approaches, which shows that the EP based approach obtains the comparable estimation accuracy with less computational complexity than the PF based method for both stationary and time-varying SNR, and enhances the BP decoding performance simultaneously. Moreover, the EP estimator shows a very fast convergence speed, and the additional computational overhead of the proposed decoder is less than 10% of the standard BP decoder.

Moreover, since the close relationship between source coding and channel coding, the proposed ideas are extended to source correlation estimation. First, I study the correlation estimation problem in lossless DSC setup, where I consider both asymmetric and non-asymmetric SW coding of two binary correlated sources. The aforementioned PF and EP based approaches are extended to handle the correlation between two binary sources, where the relationship is modeled as a virtual binary symmetric channel (BSC) with a time-varying crossover probability. Besides, to handle the correlation estimation problem of Wyner-Ziv (WZ) coding, a lossy DSC setup, I design a joint bit-plane model, by which the PF based approach can be applied to tracking the correlation between non-binary sources. Through experimental results, the proposed correlation estimation approaches significantly improve the compression performance of DSC.

Finally, due to the property of ultra-low encoding complexity, DSC is a promising technique for many tasks, in which the encoder has only limited computing and communication power, e.g. the space imaging systems. In this dissertation, I consider a real-world application of the proposed correlation estimation scheme on the onboard low-complexity compression of solar stereo images, since such solutions are essential to reduce onboard storage, processing, and communication resources. In this dissertation, I propose an adaptive distributed compression solution using PF that tracks the correlation, as well as performs disparity estimation, at the decoder side. The proposed algorithm is tested on the stereo solar images captured by the twin satellites system of NASAs STEREO project. The experimental results show the significant PSNR improvement over traditional separate bit-plane decoding without dynamic correlation and disparity estimation.

CHAPTER 1

INTRODUCTION

The abilities of sending information between senders and receivers are increasingly demanded by the modern world. Given an input source (e.g. voice, video, image, etc.), the sender produces a signal such that it can tolerate the transmission error caused by noisy channel, as well as minimize channel bandwidth usage. Then, the receiver can recover the original input source with the highest fidelity. The first goal usually refers to the channel coding problem, in which redundancy is introduced by a channel encoder, so that the transmission error caused by the noisy channel can be corrected at the decoder side. In contrast, the compression of sources belongs to the source coding problem, in which the source redundancy is removed by the source encoder, so that the required transmission bandwidth can be minimized. Thus, both channel coding and source coding are two important problems in communications. However, both channel coding and source coding (especially, the distributed source coding (DSC)) can achieve their ultimate performance by knowing the perfect knowledge of channel noise and source correlation, respectively, such information may not be always available at the decoder side. The reasons might be because of the time-varying characteristic of some communication systems and sources themselves, respectively. In this dissertation, I mainly focus on the study of online channel noise estimation and correlation estimation by using both stochastic and deterministic approximation inferences on factor graphs. In this chapter, I briefly describe channel coding problems and DSC problems, and then show the contributions and organization of this dissertation.

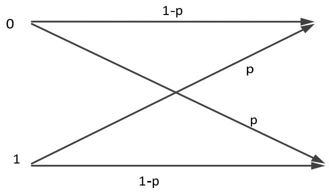

Figure 1.1: Binary symmetric channel $C = 1 - H(p)$ bits.

1.1 Channel Coding

In a point-to-point communication system, a source signal can be transmitted over a physical channel from a sender to a receiver. However, such a channel is generally imperfect and therefore leads to noisy output at the receiver side. To achieve a high-fidelity communication, a channel encoder is used to introduce helpful redundancy, with which the receiver can correct transmission errors caused by the noisy channel and reconstruct the original source. In this section, I will introduce some basic communication channels, such as the binary symmetric channel (BSC) and Gaussian channel, and derive their channel capacities.

1.1.1 Channel Capacity

Binary Symmetric Channel

A BSC is a channel with binary input and binary output, and each input produces an output with an error probability p. As shown in Fig.1.1, if the error occurs, the input symbol 0 will produce output symbol 1, and vice versa. The channel capacity of the BSC is evaluated by the mutual information as follows

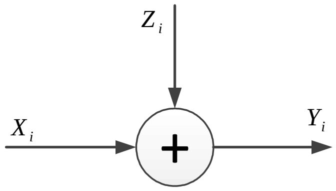

Figure 1.2: Gaussian channel.

$$
\begin{aligned}
I(X,Y) &= H(Y) - H(Y|X) \\
&= H(Y) - \sum p(x) H(Y|X=x) \\
&= H(Y) - \sum p(x) H(p) \\
&= H(Y) - H(p) \\
&\leq 1 - H(p)
\end{aligned}
\tag{1.1}
$$

Thus, the channel capacity of the BSC is $C = \max_{p(x)} I(X;Y) = 1 - H(p)$ bits.

Gaussian Channel

A Gaussian channel is one of the most important channels with continuous channel input and output. As shown in Fig.1.2, the Gaussian channel is presented by a series of discrete input X_i and discrete output Y_i, where i is the index of discrete time. The output Y_i is the sum of the input X_i and noise, Z_i, where $Z_i \sim N(0,v)$ satisfies a Gaussian distribution with zero mean and variance v and is assumed to be independent of input X_i. Thus,

$$Y_i = X_i + Z_i \quad Z_i \sim N(0,v). \tag{1.2}$$

For many practical communication channels, such as radio and satellite links, there are energy or power constraints on the input symbols. Usually, an average power constraint for any codeword $x_1, \cdots, x_n$ is defined as $\frac{1}{n}\sum_{i=1}^{n} x_i^2 \le P$. The channel capacity of a Gaussian channel with power constraint P is defined as

$$C = \max_{p(x):EX^2 \le P} I(X;Y). \tag{1.3}$$

To calculate the capacity, I first derive the mutual information

$$\begin{aligned} I(X;Y) &= h(Y) - h(Y|X) \\ &= h(Y) - h(X+Z|X) \\ &= h(Y) - h(Z|X) \\ &= h(Y) - h(Z) \end{aligned} \tag{1.4}$$

where $h(Z) = \frac{1}{2}\text{log}2\pi ev$. To obtain the higher bound of entropy of Y, I first calculate $E(Y^2) = E(X+Z)^2 = EX^2 + 2EXEZ + EZ^2 = P + v$, where $EZ = 0$. Given the variance $P+v$ for Y, the maximum entropy of Y is $h(Y) = \frac{1}{2}\text{log}2\pi e(P+v)$.

Then, the capacity of Gaussian channel is

$$\begin{aligned} C &= \text{max}I(X,Y) \\ &= \frac{1}{2}\text{log}2\pi e(P+v) - \frac{1}{2}\text{log}2\pi ev \\ &= \frac{1}{2}\text{log}(1+\frac{P}{v}). \end{aligned} \tag{1.5}$$

where capacity C achieves the maximum value when $X \sim N(0,P)$.

1.1.2 Channel Coding

During the past decades, numerous error-correcting codes have been proposed in the channel coding area. Among these existing codes, low-density parity check (LDPC) codes [3] have raised wide interests in the research community, because the performance of LDPC codes can make the data transmission rates achieve near Shannon's limit [4, 5]. As a type of error-correcting code, LDPC codes were first proposed by Gallager in the early 1960s [3] and revived by Mackay and Neal in 1996 [6]. LDPC codes can be decoded by using a powerful iterative algorithm known as the belief propagation (BP) algorithm [6]. However, the decoding performance of LDPC codes usually relies on the knowledge of channel noise statistics. Hence, a better initial estimate of the channel noise statistics, e.g. the noise variance of an additive white Gaussian noise (AWGN) channel or the crossover probability of a BSC, would generate a better decoding performance of LDPC codes.

1.2 Distributed Source Coding

Sensor technology has grown tremendously in recent decades. The main objective of a sensor network is to relay sensor observations back to a basestation efficiently. Given the high constraint on bandwidth and power of a mesh network, it is important to reduce the transmission load and still obtain all observations with high fidelity. However, in many scenarios, sensors can only transmit data to a basestation and do not have the capability to communicate with each other. Hence, joint encoding of sensor observations is often impossible (or impractical). Fortunately, as sensor observations are usually correlated, it is possible to reduce the communication loads by taking advantage of these correlations. The main enabling technology to achieve this is DSC [7, 8]. DSC is a technique to compress correlated remote sources separately and decompress them jointly. Generally, DSC problems can be classified into lossless and lossy setups, which are also known as Slepian-Wolf (SW) [9] problems and Wyner-Ziv (WZ) [10] problems, respectively.

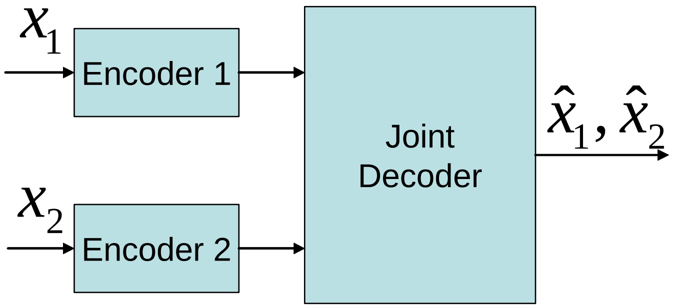

Figure 1.3: SW coding (lossless DSC) of two sources.

1.2.1 Lossless DSC (SW Coding) Design

Slepian and Wolf proved a very surprising result, namely that, generally speaking, it is possible to have no performance loss of separate encoding compared to the case when joint encoding is allowed [9]. As shown in Fig.1.3, two sensor outputs are separately compressed and transmitted to a basestation for joint decoding. Let R_1 and R_2 be the corresponding compression rates for a source pair x_1 and x_2 as shown in Fig.1.3. They show that lossless compression is possible if and only if

$$R_1 \geq H(x_1|x_2), R_2 \geq H(x_2|x_1), \text{and } R_1 + R_2 \geq H(x_1, x_2), \tag{1.6}$$

where $H(x)$ denotes the entropy of variable x. For example, when x_1 is compressed independently, we will need $R_1 = H(x_1)$ to achieve lossless compression. According to the SW Theorem, it is sufficient to have $R_2 = H(x_2|x_1)$. And thus the total rate is $R_1 + R_2 = H(x_1) + H(x_2|x_1) = H(x_1, x_2)$. This is equivalent to the rate required even when joint compression is allowed. We usually refer to this no-performance-loss feature as no rate loss.

Similar to Shannon's channel coding theorem [11], the proof of the SW Theorem is non-constructive in that Slepian and Wolf did not indicate how to implement DSC efficiently. The research on practical algorithms for DSC was stagnant until

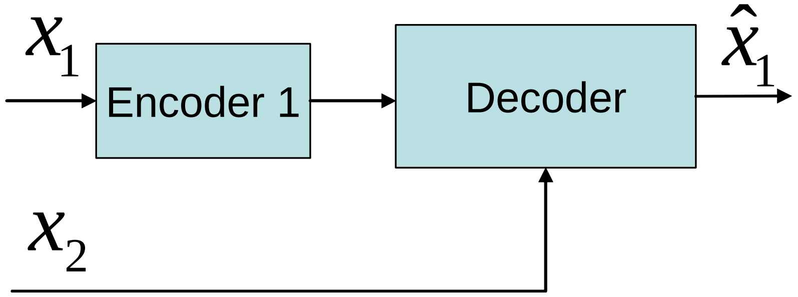

Figure 1.4: WZ coding (lossy source coding with side information at the decoder).

the publication of the work by Pradhan and Ramchandran in 1999 [12]. They rediscovered an early work by Wyner in which he suggested the use of channel codes for asymmetric SW coding [13]. Here, we describe a SW problem as asymmetric in the sense that only one of all correlated sources is compressed while the rest is transmitted uncompressed as side information. Practical syndrome-based schemes for SW coding using channel codes have been studied in [14–16, 16–25]. Notably, the Wyner approach is further extended to the non-asymmetric case (i.e., sources from more than one terminals will be compressed) [14, 26].

1.2.2 Lossy DSC (WZ Coding) Design

Wyner and Ziv later considered a rate distortion problem closely related to DSC [10, 27], namely, the rate distortion problem that occurs when side information is given to the decoder but not the encoder. Indeed, WZ coding can be treated as a degenerated case of DSC with sources at two different terminals. As shown in Fig.1.4, one source is transmitted directly to the decoder whereas the source at the other terminal is compressed and recovered with the help of the first source acting as side information. On the other hand, this problem generalizes the SW setup in which coding of the source is lossy with respect to a fidelity criterion, rather than lossless.

1.3 The Dissertation Contributions

In channel coding problems, knowing the signal to noise ratio (SNR) is necessary to achieve the best performance. Thus, in many previous studies, the SNR is assumed to be perfectly known prior to decoding. In reality, however, the perfect knowledge of the SNR may not always be available at the decoder, as the channel SNR may vary over time. Although one may argue that the actual SNR may be able to be obtained through a pilot signal or feedback channel under varying channel conditions, a fast varying channel implies potentially a large communication overhead if we want to take full advantage of the channel state information. Similarly, the performance of DSC depends on the knowledge of correlation between sources. Nevertheless, in many real applications, such as a sensor network which is widely used for environmental monitoring of temperature, pressure and humidity, or real-time area video surveillance, the correlation statistics among sensors cannot be obtained easily. In general, the correlations among sensors may vary over both space and time. Thus, the implementation of DSC suffers the same "trouble" from the inaccurate prior estimate of the correlation statistics as that of the channel coding. Since the decoding performances of channel coding and DSC highly rely on the knowledge of SNR and correlation, respectively, the design of an online estimation scheme of SNR for channel coding and correlation for sensor networks becomes a significant task both in theoretical study and practical applications.

The contributions of this dissertation can be divided into three portions:

1. Two adaptive LDPC decoding schemes were proposed, which enable one to perform online estimation of time-varying SNR at the bit-level by using the stochastic method (i.e., PF) and the deterministic method (i.e., EP). Through experimental results, I compare the performance between the proposed PF

based and EP based approaches, which shows that the proposed EP based approach obtains the comparable estimation accuracy with less computational complexity than PF based method for both stationary and time-varying SNR, and enhances the BP decoding performance simultaneously.

2. Due to the close relationship between source coding and channel coding, the proposed ideas are extended to source correlation estimation. First, I study the correlation estimation problem in lossless DSC setup, where I consider both asymmetric and non-asymmetric SW coding of two binary correlated sources. The aforementioned PF and EP based approaches are extended to handle the correlation between two binary sources, where the relationship is modeled as a virtual BSC with a time-varying crossover probability. Moreover, to handle the correlation estimation problem of WZ coding, a lossy DSC setup, I design a joint bit-plane model, by which the PF based approach can be applied to tracking the correlation between non-binary sources. Through experimental results, the proposed correlation estimation approaches significantly improve the compression performance of DSC.

3. A real-world application of the proposed correlation estimation scheme on the onboard low-complexity compression of solar stereo images was considered, since low-complexity compression solutions are essential to reduce onboard storage, processing, and communication resources. In this dissertation, I propose an adaptive distributed compression solution using PF that tracks correlation, as well as performs disparity estimation, at the decoder side. The proposed algorithm is tested on the stereo solar images captured by the twin satellites system of NASAs STEREO project. The experimental results show the significant PSNR improvement over traditional separate bit-plane decod-

ing without dynamic correlation and disparity estimation.

1.4 The Organization of The Dissertation

In Chapter 2, I will first review some background knowledge about probability and graphical models. In Chapter 3, I will introduce approximation inference, which includes both stochastic approximation and deterministic approximation. Then, the SNR estimation in channel coding with PF and EP will be presented in Chapter 4. Moreover, I will extend the ideas from channel coding problem to DSC problem in Chapter 5, in which both SW coding and WZ coding problems with correlation estimation are studied. In addition, I will present the implementation of the proposed scheme on a real-world application, the onboard solar stereo images compression in Chapter 6. Finally, I will draw our conclusion in Chapter 7.

Research in the dissertation has been published in several international journals and conferences. In Chapter 4, the work of PF based SNR estimation over AWGN channel has been published in IEEE Transactions on Communications [28]. The study of EP based SNR estimation over AWGN channel is a part of a conference paper in IEEE GLOBECOM 2011 [29] and a journal submission in IEEE Transactions on Communications [30]. The work on PF based correlation estimation in Chapter 5 has been published in IEEE Transactions on Communications [31], IEEE Transactions on TCSVT [32] and conferences [33–35]. Regarding the EP based correlation estimation work, one journal paper has been accepted by IEEE Communications Letter [36]. Furthermore, Chapter 6 has been published in 2011 IEEE International Conference on Image Processing [37] and submitted to IEEE Transactions on Image Processing [38].

CHAPTER 2

GRAPHICAL MODELS

This chapter starts from the introduction to Bayesian probability theory, because it plays a significant role in the field of probabilistic inference and is also the foundation of the rest of this dissertation. Then, I discuss a powerful representation of probabilistic models, called graphical models, which offer a great flexibility for problem solving and system modeling (especially factor graph for LDPC codes). This chapter is a concise review of the aforementioned concepts. For more detail on graphical models, I direct readers to these references [39–41].

2.1 Bayesian Probabilities

Bayesian probability theory provides a theoretical framework for reasoning under a probability. It has been widely used in many engineering disciplines such as communications, artificial intelligence, etc. In this section, I discuss the basics of Bayesian probability theory and provide a foundation of my later discussion.

2.1.1 Bayes' Rule

In a probabilistic model, suppose $\mathbf{y} = \{y_1, y_2, \cdots, y_N\}$ is an observed data set, and $\mathbf{x} = \{x_1, x_2, \cdots, x_{N'}\}$ is a set of hidden variables/parameters, where N and N' are some positive integer numbers. Then Bayes's rule can be expressed as

$$p(\mathbf{x}|\mathbf{y}) = \frac{p(\mathbf{y}, \mathbf{x})}{p(\mathbf{y})} = \frac{p(\mathbf{x})p(\mathbf{y}|\mathbf{x})}{p(\mathbf{y})}. \tag{2.1}$$

In (2.1), the probability $p(\mathbf{x}|\mathbf{y})$, also called the *posterior distribution*, allows us to evaluate the probability distribution of $\mathbf{x}$ given the observed data $\mathbf{y}$. The probability

$p(\mathbf{y}, \mathbf{x})$ is called the *joint distribution*, and $p(\mathbf{x})$ is called the *prior distribution*, which captures the probability of the hidden variables in $\mathbf{x}$ before observing the data. The quantity $p(\mathbf{y}|\mathbf{x})$ measures how likely the observed data set $\mathbf{y}$ is for different $\mathbf{x}$ and is called the *likelihood function*. Moreover, $p(\mathbf{y})$ can be interpreted as a *normalization constant* and can be evaluated through the marginalization step as follows

$$p(\mathbf{y}) = \int p(\mathbf{x})p(\mathbf{y}|\mathbf{x})d\mathbf{x}. \tag{2.2}$$

2.1.2 Bayesian Inference

In a nutshell, Bayesian inference is a statistical inference method in which any unknown is expressed in terms of probability. The fundamental differences between Bayesian and non-Bayesian paradigms are the ways they use the hidden variables in $\mathbf{x}$. The non-Bayesian paradigm always considers fixed variables in $\mathbf{x}$, whose values are determined by some estimators, e.g. *maximum likelihood* (ML) estimator. Then, the statistical accuracy of estimates (i.e. error bar) can be obtained by evaluating different data sets sampled from $p(\mathbf{y})$. By contrast, the Bayesian paradigm captures the hidden variables in $\mathbf{x}$ through its posterior distribution directly. One advantage of Bayesian inference is the inclusion of prior knowledge, which can avoid some unreasonable conclusions made by the ML estimator. Moreover, based on new observed data, Bayesian inference also offers an easier way to make predictions by taking the 'old' posterior distribution as 'new' prior knowledge. Then, by using Bayesian inference, one can decouple models from observed data.

2.2 Graphical Models

As mentioned in the previous section, Bayesian probability theory has played an important role in the modern machine learning and the probabilistic inference.

In this section, I discuss a powerful representation of probabilistic models, called graphical models, which offer a great flexibility for problem solving and system modeling (especially factor graph for LDPC codes).

A graph $G(V, E)$ is defined by a set of nodes V and a set of edges E connecting these nodes, where each node represents a random variable or a group of random variables, and each edge represents the statistical dependency between the connecting variables. Then the decomposition of the joint probability over all the random variables can be expressed by a graph, where the complex global algebraic calculation can be replaced by local graphical manipulations. In probabilistic graphical models, there are mainly three different kinds of graphs, i.e. directed graph, undirected graph and factor graph, where the directed and the undirected graphical models are also known as Bayesian networks and Markov random fields, respectively. In this dissertation, I focus on the discussion of factor graphs, since converting both directed and undirected graphs into factor graphs is often the canonical way for solving inference problems.

2.2.1 Factor Graph

A factor graph is a bipartite graph, which comprises two different kinds of nodes (i.e. factor node and variable node). In factor graphs, each edge must connect a factor node and a variable node. Moreover, each factor node represents one of the factors over subsets of some variables in a decomposed joint distribution. Each variable node expresses a random variable. Suppose that the decomposition of the joint distribution over a set of random variables has the form of a product of factors

$$p(\mathbf{x}) = \prod_{s} f_s(\mathbf{x}_s), \tag{2.3}$$

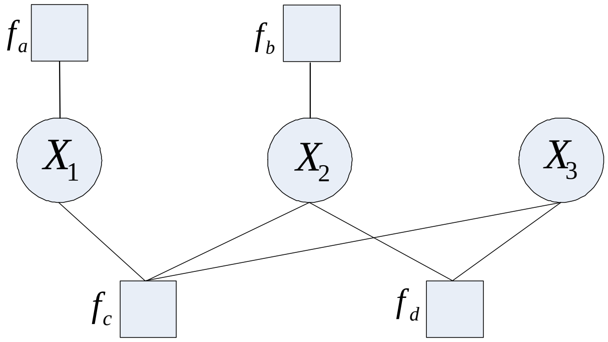

Figure 2.1: Example of a factor graph.

where the bold $\mathbf{x}$ is a set of variables in the joint distribution, $\mathbf{x}_s$ is a subset of variables and $f_s(\mathbf{x}_s)$ is a function of all variables in $\mathbf{x}_s$. Moreover, x_i denotes each individual variable in the joint distribution.

For example, let us consider the factorization of a joint distribution $p(\mathbf{x})$ as

$$p(\mathbf{x}) = f_a(x_1)f_b(x_2)f_c(x_1, x_2, x_3)f_d(x_2, x_3). \tag{2.4}$$

Then the factorization of $p(\mathbf{x})$ can be expressed by the factor graph in Fig 2.1. In this dissertation, please note that I use circles and squares to represent variable nodes and factor nodes in the factor graph, respectively.

2.2.2 The Sum-product Algorithm

The sum-product algorithm (or BP algorithm) is an efficient and exact inference algorithm for computing local marginals over variables on tree-structured graphs. For graphs with loops, a lot of applications (e.g. LDPC decoding [6] and image processing [42]) show that BP algorithm (or loopy BP algorithm) still provides a good performance.

To introduce the sum-product algorithm, let us take a look at the following example first. Suppose that a system has a set of hidden variables $\mathbf{x} =$

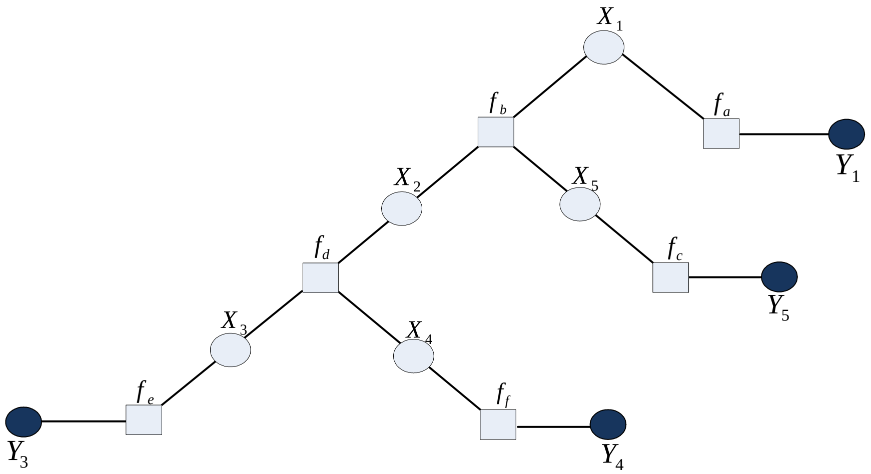

Figure 2.2: The factor graph for the factorization of joint probability in (2.6).

$\{x_1, x_2, x_3, x_4, x_5\}$ and a set of observations $\mathbf{y} = \{y_1, y_3, y_4, y_5\}$. We are interested in the estimate of each hidden variable x_i, $i = 1, 2, \cdots, 5$, given the observed data $\mathbf{y}$. Thus, the estimate $\hat{x}_i$ can be expressed as

$$\begin{aligned} \hat{x}_i &= \arg\max_{x_i} p(x_i|\mathbf{y}) \\ &= \arg\max_{x_i} \frac{p(x_i, \mathbf{y})}{p(\mathbf{y})} \\ &= \arg\max_{x_i} p(x_i, \mathbf{y}). \end{aligned} \tag{2.5}$$

(2.5) require us to compute the marginal distribution $p(x_i, \mathbf{y})$ out of the joint distribution $p(\mathbf{x}, \mathbf{y})$. For this toy problem, we can compute the marginal distributions for each variable independently. However, for a large-scale problem with hundred or even thousand hidden variables, it is infeasible to independently marginalize each variable, since the computational burden is very expensive. Fortunately, the BP algorithm on factor graphs provides an efficient way to compute marginal distributions over hidden variables.

Suppose that the factorization of joint probability $p(\mathbf{x}, \mathbf{y})$ takes the form

$$\begin{aligned} p(\mathbf{x}, \mathbf{y}) &= p(y_1|x_1)p(x_1, x_2, x_5)p(y_5|x_5)p(x_3, x_4|x_2)p(y_3|x_3)p(y_4|x_4) \\ &= f_a(x_1, y_1)f_b(x_1, x_2, x_5)f_c(x_5, y_5)f_d(x_3, x_4, x_2)f_e(x_3, y_3)f_f(x_4, y_4), \end{aligned} \tag{2.6}$$

where each function $f_s(\mathbf{x_s})$ corresponds to a factor with the same variables in the joint distribution. According the above factorization, we can construct the corresponding factor in Fig. 2.2. For example, let us compute the marginal probability of the discrete variable x_1 as follows

$$p(x_1, \mathbf{y}) = \underbrace{f_a(x_1, y_1)}_{m_{f_a \to X_1}(x_1)} \underbrace{\sum_{x_1, x_5} f_b(x_1, x_2, x_5) \underbrace{f_c(x_5, y_5)}_{m_{f_c \to X_5}(x_5)} \underbrace{\sum_{x_3, x_4} f_d(x_3, x_4, x_2) \underbrace{f_e(x_3, y_3)}_{m_{f_e \to X_3}(x_3)} \underbrace{f_f(x_4, y_4)}_{m_{f_f \to X_4}(x_4)}}_{m_{f_d \to X_2}(x_2)}}_{m_{f_b \to X_1}(x_1)}, \tag{2.7}$$

where $m_{f_s \to X_i}(x_i)$ denotes a message sent from a factor node f_s to a variable node X_i. Moreover, let us introduce $m_{X_i \to f_s}(x_i)$ as the message sent from a variable node X_i to a factor node f_s. By inspecting (2.7), we can conclude that the variable node message update rule as

$$m_{X_i \to f_s}(x_i) \propto \prod_{s' \in \mathcal{N}(X_i) \backslash s} m_{f_{s'} \to X_i}(x_i), \tag{2.8}$$

and the factor node update rule as

$$m_{f_s \to X_i}(x_i) \propto \sum_{\mathbf{x}_s \backslash x_i} \left(f_s(\mathbf{x}_s) \prod_{j \in \mathcal{N}(f_s) \backslash i} m_{X_j \to f_s}(x_j) \right), \tag{2.9}$$

where $\mathcal{N}(X_i) \backslash s$ denotes the set of all neighbors' indices of node X_i excluding the index s of the factor node f_s; $f_s(\mathbf{x}_s)$ is the factor function for the factor node f_s;

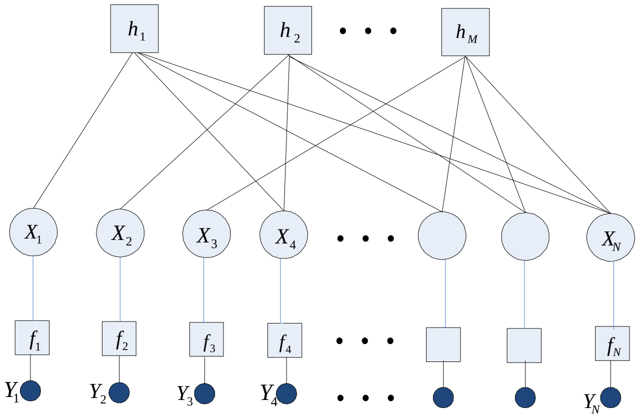

Figure 2.3: Factor graph of LDPC decoding.

$\sum\limits_{\mathbf{x}_s \backslash x_i}$ denotes a sum over all the variables in $\mathbf{x}_s$, that are arguments of $f_s(\mathbf{x}_s)$, except x_i. Loosely speaking, $m_{f_s \to X_i}(x_i)$ and $m_{X_i \to f_s}(x_i)$ can be interpreted as the beliefs of node X_i taking the value x_i transmitting from node f_s to X_i and vice versa. Finally, the sum-product algorithm computes the marginal probability of variable x_i, also called the belief $b(x_i)$ at a variable node X_i, as follows

$$b(x_i) \propto \prod_{s \in \mathcal{N}(X_i)} m_{f_s \to X_i}(x_i). \tag{2.10}$$

So far, we suppose that all of the variables are discrete, so the marginalization is computed by summation. However, the sum-product algorithm is also applicable to linear-Gaussian models by replacing summation by integration, e.g. Kalman filtering.

2.2.3 LDPC Decoding using Sum-product Algorithm

I shall show how to make use of factor graphs and the sum-product algorithm to perform LDPC decoding. Suppose that $\mathbf{x} = \{x_1, x_2, \cdots, x_N\}$ is a codeword of length N and transmitted over a memoryless noise channel (e.g. AWGN or BSC channel) with corresponding output $\mathbf{y} = \{y_1, y_2, \cdots, y_N\}$. Then in Bayesian viewpoint, the channel decoding problem is equivalent to the evaluation of the posterior probability $p(x_i|\mathbf{y})$ over each variable. Fig. 2.3 shows the factor graph of LDPC decoding problem. LDPC codes are usually defined on an $M \times N$ sparse parity check matrix H as

$$H\mathbf{x}^T = 0. \tag{2.11}$$

Thus, factor nodes h_k, $k = 1, 2, \ldots, M$, connect the bit variable nodes X_i, $i = 1, 2, \ldots, N$, and take into account the constraints imposed by the LDPC codes (see (2.11)). Here x_i, the variable of node X_i, is a candidate value of the i-th bit of the channel input (coded message). The corresponding factor function $h_k(\mathbf{x}_k)$ is given by

$$h_k(\mathbf{x}_k) = \begin{cases} 1 & \text{if there are even number of 1's in arguments} \\ 0 & \text{otherwise,} \end{cases} \tag{2.12}$$

where $\mathbf{x}_k$ indicates all variables connecting to factor node h_k, that is $\mathbf{x}_k = (x_i | i \in \mathcal{N}(h_k))$, and $\mathcal{N}(h_k)$ represents the set of neighbors' indices for a node h_k. Moreover, factor node f_i connects the bit variable node X_i and the observation variable node Y_i, where y_i, the variable of node Y_i, is the i-th bit of the channel output. The factor node f_i plays a role of providing a predetermined likelihood $p(y_i|x_i, \theta)$ to variable node X_i for LDPC decoding, where θ is the channel parameter, e.g. the noise variance for AWGN channels or the crossover probability for BSC. The corresponding factor function of f_i is defined as

$$f_i(y_i; x_i, \theta) = \begin{cases} \frac{1}{\sqrt{2\pi\theta}} \exp\left(-\frac{(y_i - x_i)^2}{2\theta}\right) & \text{for AWGN channel} \\ \theta^{x_i \oplus y_i} (1-\theta)^{1 \oplus x_i \oplus y_i} & \text{for BSC channel,} \end{cases} \tag{2.13}$$

where $\oplus$ is a binary sum operator. With the factor functions and an accurate channel parameter θ in place, the performance of BP based LDPC decoding scheme can be very close to the theoretical limit.

CHAPTER 3

APPROXIMATE INFERENCE

As shown in the previous chapter, the sum-product algorithm is a powerful probabilistic inference technique for efficiently computing posterior probabilities over discrete variables of small alphabet sizes or continuous variables of linear Gaussian distributions. However, it cannot handle a discrete variable with a medium alphabet size as the computational complexities increase exponentially with the alphabet size. Moreover, for continuous variables with non-linear Gaussian distribution, sum-product is also infeasible as the integration may not have a closed-form solution. Due to this fatal weakness of the sum-product algorithm, I cannot directly use it in my research for channel or correlation estimation, since the channel statistics or the source correlation is usually continuous and cannot be expressed by a linear Gaussian model. Two common workarounds in approximate inference are either to discretize the variable through sampling techniques or to parametrize the variables through variational inference, where the sampling and the variational methods are also known as stochastic and deterministic approximation schemes. In this chapter, I will describe the stochastic approximation and deterministic approximation.

3.1 Stochastic Approximation

Stochastic approximation is an approximate technique based on numerical sampling, which is also known as Monte Carlo techniques. In many applications, posterior distributions are employed to make predictions by evaluating expectations. For example, considering some function $f(\mathbf{x})$ with continuous variable $\mathbf{x}$, the expectation of function $f(\mathbf{x})$ respects to probability distribution $p(\mathbf{x})$ can be written as

$$\mathrm{E}[f] = \int f(\mathbf{x})p(\mathbf{x})d\mathbf{x}, \tag{3.1}$$

where the integral can be replaced by summation for discrete variables. However, for arbitrary functions and probability distributions, the exact evaluation over the integral may not be alway tractable. One workaround is to resort to stochastic approximation method. The general idea of sampling method is to independently draw a set of samples $\mathbf{x}^l$, $l = 1, 2, \cdots, L$ from the distribution $p(\mathbf{x})$, and then (3.1) can be approximated by a finite sum

$$\hat{f} = \frac{1}{L}\sum_{l=1}^{L} f(\mathbf{x}^l), \tag{3.2}$$

where the estimator $\hat{f}$ has the correct mean $\mathrm{E}[\hat{f}] = \mathrm{E}[f]$, and the variance $\mathrm{var}[\hat{f}] = \frac{1}{L}\mathrm{E}[(f - \mathrm{E}[f])^2]$. Since the estimator does not depend on the dimensionality of x, usually a relatively small number (e.g. ten or twenty) of independent samples may achieve a sufficient accuracy in principle. However, the problem is that samples might not be alway independent in practice, thus the effective sample size might be much smaller than the actual sample size. Therefore, to achieve sufficient accuracy, a relatively large sample size is usually required in stochastic approximation technique. This is why stochastic approximation methods are highly computational demanding.

In the rest of this section, I will concisely review some sampling strategies (i.e., important sampling and MCMC), with which I will develop my proposed algorithm. This section starts from the review of importance sampling. Then I discuss a general and powerful technique, MCMC, which could effectively avoid some limitations within the important sampling method.

3.1.1 Importance Sampling Methods

The general idea of important sampling is to draw samples from an easily sampling proposal distribution $q(\mathbf{x})$ instead of the distribution $p(\mathbf{x})$. Then the expectation form based on a finite sum in (3.2) can be rewritten as

$$\begin{aligned} \mathrm{E}[f] &= \int f(\mathbf{x})p(\mathbf{x})d\mathbf{x} \\ &= \int f(\mathbf{x})\frac{p(\mathbf{x})}{q(\mathbf{x})}q(\mathbf{x})d\mathbf{x} \\ &\simeq \frac{1}{L}\sum_{l=1}^{L}\frac{p(\mathbf{x}^l)}{q(\mathbf{x}^l)}f(\mathbf{x}^l) \\ &= \frac{1}{L}\sum_{l=1}^{L} r_l f(\mathbf{x}^l) \end{aligned} \tag{3.3}$$

where $r_l = p(\mathbf{x}^l)/q(\mathbf{x}^l)$ are important weight.

Considering the evaluation of $p(\mathbf{x})$ and $q(\mathbf{x})$ in terms of normalization constant, (3.3) can be written as

$$\begin{aligned} \mathrm{E}[f] &= \int f(\mathbf{x})p(\mathbf{x})d\mathbf{x} \\ &= \frac{Z_q}{Z_p}\int f(\mathbf{x})\frac{\tilde{p}(\mathbf{x})}{\tilde{q}(\mathbf{x})}q(\mathbf{x})d\mathbf{x} \\ &\simeq \frac{Z_q}{Z_p}\frac{1}{L}\sum_{l-1}^{L}\tilde{r}_l f(\mathbf{x}^l), \end{aligned} \tag{3.4}$$

where $\tilde{r_l} = \tilde{p}(\mathbf{x})/\tilde{q}(\mathbf{x})$, and the ratio Z_q/Z_p can also be evaluated through the same samples

$$\frac{Z_p}{Z_q} = \frac{1}{Z_q}\int \tilde{p}(\mathbf{x})d\mathbf{x} = \int \frac{\tilde{p}(\mathbf{x})}{\tilde{q}(\mathbf{x})}q(\mathbf{x})d\mathbf{x} \simeq \frac{1}{L}\sum_{l-1}^{L}\tilde{r}_l. \tag{3.5}$$

Thus,

$$\mathrm{E}[f] \simeq \sum_{l=1}^{L} w_l f(\mathbf{x}^l) \tag{3.6}$$

with

$$w_l = \frac{\tilde{r}_l}{\sum_m \tilde{r}_m} = \frac{\tilde{p}(\mathbf{x}^l)/\tilde{q}(\mathbf{x}^l)}{\sum_m \tilde{p}(\mathbf{x}^m)/\tilde{q}(\mathbf{x}^m)}. \tag{3.7}$$

Since important sampling technology introduces a proposal distribution $q(\mathbf{x})$, from which it is much easer to draw samples, the expectation accuracy depends on the selection of the proposal distribution. In the case that the $p(\mathbf{x})$ has much of their mass concentrated on a small region of $\mathbf{x}$ space, while the proposal distribution is small or zero in the regions where $p(\mathbf{x})$ has large weight, the expectation accuracy will be severely wrong.

3.1.2 Markov Chain Monte Carlo

MCMC is a more general and powerful sampling technique, which overcomes the limitation within the important sampling, and works for a large class of distributions. Before I talk about MCMC framework, I first review some general properties of Markov chains.

Markov Chains

Let us denote by $\mathbf{x}^1, \cdots, \mathbf{x}^t$ a series of random values. Then we say that the random variables are treated as first-order Markov chain, as long as the following conditional independence property holds

$$p(\mathbf{x}^{t+1}|\mathbf{x}^1, \cdots, \mathbf{x}^t) = p(\mathbf{x}^{t+1}|\mathbf{x}^t). \tag{3.8}$$

The transition probability $p(\mathbf{x}^{t+1}|\mathbf{x}^t)$ indicates that the future state of a random variable only depends on the current state. Moreover, if the transition probabilities are the same for all the states, the Markov chain is called *homogeneous*.

Markov Chain Monte Carlo

Similar to importance sampling methods, MCMC also chooses a proposal distribution, with which one can draw samples much easier. However, in MCMC, the proposal distribution $q(\mathbf{x}) = q(\mathbf{x}|\mathbf{x}^t)$ depends on the current state $\mathbf{x}^t$, which satisfies the first-order Markov chain. In addition, the candidate sample drawn from the proposal distribution is accepted with a probability. For example, in Metropolis algorithm, where the proposal distribution is symmetric, the probability of accepting a candidate sample can be expressed as

$$a(\mathbf{x}^*, \mathbf{x}^t) = \min\left(1, \frac{\tilde{p}(\mathbf{x}^*)}{\tilde{p}(\mathbf{x}^t)}\right). \tag{3.9}$$

If a candidate sample is accepted, then sample at the next state will take the candidate sample value $\mathbf{x}^{t+1} = \mathbf{x}^*$; otherwise, the sample at the next state will take the current state sample value $\mathbf{x}^{t+1} = \mathbf{x}^t$. Then the next candidate sample can be drawn from the updated distribution $q(\mathbf{x}|\mathbf{x}^{t+1})$.

3.2 Deterministic Approximation

The existing deterministic approximation methods include Laplace approximation (LA), variational Bayes (VB), Expectation Propagation (EP) and so on. In this section, I will focus on the discussion of EP, since EP usually shows a higher accuracy comparing with other deterministic approximation methods, if the selected model matches the problem. This section contains three parts. First, I will review some preliminaries for EP. Then, EP algorithm will be described in the next section. Finally, the relationship between EP and other variational methods will be discussed.

3.2.1 Preliminaries

Exponential Family

The exponential family of distributions over $\mathbf{x}$ is a set of distributions with the form

$$p(\mathbf{x};\boldsymbol{\theta}) = h(\mathbf{x})g(\boldsymbol{\theta})\mathrm{exp}\left(\boldsymbol{\theta}^T\mathbf{u}(\mathbf{x})\right), \tag{3.10}$$

where measurement $\mathbf{x}$ may be scalar or vector, discrete or continuous, $\boldsymbol{\theta}$ are parameters of the distribution, $h(\mathbf{x})$ and $\mathbf{u}(\mathbf{x})$ are some functions of $\mathbf{x}$, and the function $g(\boldsymbol{\theta})$ is a normalization factor as

$$g(\boldsymbol{\theta})\int h(\mathbf{x})\mathrm{exp}\left(\boldsymbol{\theta}^T\mathbf{u}(\mathbf{x})\right)d\mathbf{x} = 1. \tag{3.11}$$

In addition, if the variables are discrete, just simply replace the integration with summation.

Exponential family has many properties, which may simplify computations. For example, if a likelihood function is one of the members in the exponential family, the posterior can be expressed in a closed-form expression by choosing a conjugate prior within the exponential family. Moreover, exponential family has a wide range of members such as Gaussian, Bernoulli, discrete multinomial, Poisson and so on, thus it is applicable to many different inference models.

Kullback-Leibler Divergence

Kullback-Leibler (KL) divergence [43] is a measure to quantify the difference between a probabilistic distributions $p(\mathbf{x})$ and an approximate distribution $q(\mathbf{x})$. For the distributions $p(\mathbf{x})$ and $q(\mathbf{x})$ over continuous variables, KL divergence is defined as

$$D_{KL}(p(\mathbf{x})\|q(\mathbf{x})) = \int p(\mathbf{x})\mathrm{log}\frac{p(\mathbf{x})}{q(\mathbf{x})}d\mathbf{x}, \tag{3.12}$$

where for discrete variables, just replace integration with summation. Moreover, KL divergence is non-symmetric, which means $D_{KL}(p(\mathbf{x})\|q(\mathbf{x})) \neq D_{KL}(q(\mathbf{x})\|p(\mathbf{x}))$. To give readers an intuitive view about the difference between the above two forms of KL divergence, we assume that the true distribution $p(\mathbf{x})$ is multimodal and the candidate distribution $q(\mathbf{x})$ is unimodal. By minimizing $D_{KL}(q(\mathbf{x})\|p(\mathbf{x}))$, the approximate distribution $q(\mathbf{x})$ will pick one of the modes in $p(\mathbf{x})$, which is usually used in variational Bayes method. However, the best approximate distribution $q(\mathbf{x})$ obtained by minimizing $D_{KL}(p(\mathbf{x})\|q(\mathbf{x}))$ will be the average of all modes. The later case is used in the approximate inference procedure of EP. Since this section focus on the review of EP algorithm, I will study the property of minimizing $D_{KL}(p(\mathbf{x})\|q(\mathbf{x}))$ first. Regarding the difference between minimizing $D_{KL}(p(\mathbf{x})\|q(\mathbf{x}))$ and $D_{KL}(q(\mathbf{x})\|p(\mathbf{x}))$, I will discuss it later in this chapter.

To ensure a tractable solution for minimizing KL divergence $D_{KL}(p(\mathbf{x})\|q(\mathbf{x}))$, the approximate distribution $q(\mathbf{x})$ is usually restricted within a member of the exponential family. Thus, according to (3.10), $q(\mathbf{x})$ can be written as

$$q(\mathbf{x};\boldsymbol{\theta}) = h(\mathbf{x})g(\boldsymbol{\theta})\mathrm{exp}\left(\boldsymbol{\theta}^T\mathbf{u}(\mathbf{x})\right), \tag{3.13}$$

where $\boldsymbol{\theta}$ are the parameters of the given distribution.

By substituting $q(\mathbf{x};\boldsymbol{\theta})$ into the KL divergence $D_{KL}(p(\mathbf{x})\|q(\mathbf{x}))$, we get

$$D_{KL}(p(\mathbf{x})\|q(\mathbf{x})) = -\ln g(\boldsymbol{\theta}) - \boldsymbol{\theta}^T\mathbb{E}_{p(\mathbf{x})}[\mathbf{u}(\mathbf{x})] + \text{const}, \tag{3.14}$$

where the const represents all the terms that are independent of parameters $\boldsymbol{\theta}$. To minimize KL divergence, take the gradient of $D_{KL}(p(\mathbf{x})\|q(\mathbf{x}))$ with respect to $\boldsymbol{\theta}$ to zero, we get

$$-\bigtriangledown \ln g(\boldsymbol{\theta}) = \mathbb{E}_{p(\mathbf{x})}[\mathbf{u}(\mathbf{x})]. \tag{3.15}$$

Moreover, for (3.11), taking the gradient of both sides respect to $\boldsymbol{\theta}$, we get

$$\bigtriangledown g(\boldsymbol{\theta}) \int h(\mathbf{x})\exp\left\{\boldsymbol{\theta}^T\mathbf{u}(\mathbf{x})\right\} d\mathbf{x} + g(\boldsymbol{\theta}) \int h(\mathbf{x})\exp\left\{\boldsymbol{\theta}^T\mathbf{u}(\mathbf{x})\right\} \mathbf{u}(\mathbf{x})d\mathbf{x} = 0. \tag{3.16}$$

Then by rearranging and reusing (3.11) again, we get

$$-\bigtriangledown \ln g(\boldsymbol{\theta}) = \mathbb{E}_{q(\mathbf{x})}[\mathbf{u}(\mathbf{x})]. \tag{3.17}$$

By comparing (3.15) and (3.17), we obtain

$$\mathbb{E}_{p(\mathbf{x})}[\mathbf{u}(\mathbf{x})] = \mathbb{E}_{q(\mathbf{x})}[\mathbf{u}(\mathbf{x})]. \tag{3.18}$$

Thus, from (3.18), we see that the minimization of KL divergence is equivalent to matching the expected sufficient statistics. For example, for minimizing KL divergence with a Gaussian distribution $q(\mathbf{x};\boldsymbol{\theta})$, we only need to find the mean and covariance of $q(\mathbf{x};\boldsymbol{\theta})$ that are equal to the mean and covariance of $p(\mathbf{x};\boldsymbol{\theta})$, respectively.

Assumed-density Filtering (ADF)

ADF is a technique to construct a tractable approximation to complex probability distribution. EP can be viewed as an extension on ADF. Thus, I first provide a concise review of ADF and then extend it to EP algorithm.

Let us consider the same problem studied in Section 2.1.1. In this case, we suppose that the factorization of (2.1) has the following form

$$\begin{aligned} p(\mathbf{x}|\mathbf{y}) &= \frac{p(\mathbf{x})p(\mathbf{y}|\mathbf{x})}{p(\mathbf{y})} \\ &= \frac{1}{Z}p_0(\mathbf{x})\prod_{i=1}^{N}p(y_i|\mathbf{x}), \\ &= \frac{1}{Z}\prod_{i=0}^{N}p_i(\mathbf{x}), \end{aligned} \tag{3.19}$$

where Z is a normalization constant, $p_i(\mathbf{x})$ is a simplified notation of each corresponding factor in (3.19), where $p_0(\mathbf{x}) = p_0(\mathbf{x})$ and $p_i(\mathbf{x}) = p_i(y_i|\mathbf{x})$ for $i > 0$. If we assume that the likelihood function $p(y_i|\mathbf{x})$ has a complex form, the direct evaluation of the posterior distribution would be infeasible. For example, if each likelihood function is a mixture of two Gaussian distributions and there is total $N = 100$ number of observed data. Then to get the posterior distribution, we need to evaluate mixture of 2^{100} Gaussians.

To solve this problem, approximate inference methods try to seek an approximate posterior distribution that can be very close to the true posterior distribution $p(\mathbf{x}|\mathbf{y})$. Usually, the approximate distributions are chosen within the exponential family to ensure the computational feasibility. Then the best approximate distribution can be found by minimizing KL divergence:

$$\boldsymbol{\theta}^* = \arg\min_{\theta} D_{KL}(p(\mathbf{x})\|q(\mathbf{x};\boldsymbol{\theta})). \tag{3.20}$$

However, we can see that it is difficult to solve (3.20) directly. ADF solves this problem by iteratively including each factor function in the true posterior distribution. Thus, at first, ADF chooses $q(\mathbf{x};\boldsymbol{\theta}^0)$ to best approximate factor function $p_0(\mathbf{x})$ through

$$\boldsymbol{\theta}^0 = \arg\min_{\theta} D_{KL}(p_0(\mathbf{x})\|q(\mathbf{x};\boldsymbol{\theta})). \tag{3.21}$$

Then ADF will update the approximation by incorporating the next factor function $p_i(y_i|\mathbf{x})$ until all the factor functions have been involved, which gives us the following update rule

$$\boldsymbol{\theta}^i = \arg\min_{\theta} D_{KL}(p_i(\mathbf{x})q(\mathbf{x};\boldsymbol{\theta}^{i-1})\|q(\mathbf{x};\boldsymbol{\theta})). \tag{3.22}$$

As shown in Section 3.2.1, if $q(\mathbf{x};\boldsymbol{\theta})$ is chosen from the exponential family, the optimal solution of (3.22) is matching the expected sufficient statistics between the approximate distribution $q(\mathbf{x};\boldsymbol{\theta}^i)$ and the target distribution $p_i(\mathbf{x})q(\mathbf{x};\boldsymbol{\theta}^{i-1})$. Moreover, according to (3.22), we can see that the current best approximation is based on the previous best approximation. For this reason, the estimation performance of ADF may be sensitive to the process order of factor functions, which may produce an extremely poor approximation in some cases. In the next section, I will provide another perspective of the ADF update rule, which results in the EP algorithm and provides a way to avoid the drawback associated with ADF algorithm.

3.2.2 Expectation Propagation

By taking another perspective, ADF can be seen as sequentially approximating the factor function $p_i(\mathbf{x})$ by the approximate factor function $\tilde{p_i}(\mathbf{x})$, which is restricted within the exponential family, and then exactly updating the approximate distribution $q(\mathbf{x};\boldsymbol{\theta})$ by multiplying these approximate factor functions. This alternative view of ADF can be described as:

$$\tilde{p}_i(\mathbf{x}) \propto \frac{q(\mathbf{x};\boldsymbol{\theta}^i)}{q(\mathbf{x};\boldsymbol{\theta}^{i-1})}, \tag{3.23}$$

which also produces the EP algorithm. EP algorithm initializes each factor function $p_i(\mathbf{x})$ by a corresponding approximate factor function $\tilde{p}_i(\mathbf{x})$. Then, at later iterations, EP revisits each approximated factor function $\tilde{p}_i(\mathbf{x})$ and refined it by multiplying all the current best estimate but one true factor function $p_i(\mathbf{x})$ of the revisiting term. After multiple iterations, the approximation is obtained according to (3.24).

$$q(\mathbf{x};\boldsymbol{\theta}^*) \propto \prod_i \tilde{p}_i(\mathbf{x}). \tag{3.24}$$

Since this procedure does not depend on the process order of the factor function, EP provides a more accurate approximation than ADF.

The general process of EP is given as follows:

1. Initialize the term approximation $\tilde{p}_i(\mathbf{x})$, which can be chosen from one of members in the exponential family based on the problem.

2. Compute the approximate distribution

   $$q(\mathbf{x};\boldsymbol{\theta}) = \frac{1}{Z}\prod_i \tilde{p}_i(\mathbf{x}), \tag{3.25}$$

   where $Z = \int \prod_i \tilde{p}_i(\mathbf{x}) d\mathbf{x}$.

3. Until all $\tilde{p}_i(\mathbf{x})$ converge:

   (a) Choose $\tilde{p}_i(\mathbf{x})$ to refine the approximate.

(b) Remove $\tilde{p}_i(\mathbf{x})$ from the current approximated distribution $q(\mathbf{x};\boldsymbol{\theta})$ with a normalization factor:

$$q(\mathbf{x};\boldsymbol{\theta}^{\setminus i}) \propto \frac{q(\mathbf{x};\boldsymbol{\theta})}{\tilde{p}_i(\mathbf{x})}. \tag{3.26}$$

(c) Update $q(\mathbf{x};\boldsymbol{\theta})$, where we first combine $q(\mathbf{x};\boldsymbol{\theta}^{\setminus i})$, current $p_i(\mathbf{x})$ and a normalizer Z_i, and then minimize the KL divergence through moment matching projection (3.18) (i.e. the $\mathbf{Proj}(\cdot)$ operator):

$$q(\mathbf{x};\boldsymbol{\theta}) = \mathbf{Proj}\left(\frac{1}{Z_i} q(\mathbf{x};\boldsymbol{\theta}^{\setminus i}) p_i(\mathbf{x})\right). \tag{3.27}$$

(d) Update $\tilde{p}_i(\mathbf{x})$ as

$$\tilde{p}_i(\mathbf{x}) = Z_i \frac{q(\mathbf{x};\boldsymbol{\theta})}{q(\mathbf{x};\boldsymbol{\theta}^{\setminus i})}. \tag{3.28}$$

4. Get the final approximate distribution through

$$p(\mathbf{x}) \approx q(\mathbf{x};\boldsymbol{\theta}^*) \propto \prod_i \tilde{p}_i(\mathbf{x}). \tag{3.29}$$

Relationship With BP

This section shows that BP algorithm is a special case of EP, where EP provides an improved approximation for models in which BP is generally intractable.

Let us first take a quick review of BP algorithm discussed in Section 2.2.2. The procedure of BP algorithm is iteratively updating all variables nodes, then updating all factor nodes through sending messages in parallel, and finally update the belief of each variable after each iteration. By taking another viewpoint, BP can be viewed as updating the belief over a variable x_i by incorporating one factor node at each time. Under this perspective, the belief of variable x_i will be updated as

$$b(x_i) = \frac{m_{X_i \to f_s}\left(x_i\right) m_{f_s \to X_i}\left(x_i\right)}{Z_i}, \tag{3.30}$$

where $Z_i = \int m_{X_i \to f_s}\left(x_i\right) m_{f_s \to X_i}\left(x_i\right) dx_i$ is the normalization factor. Moreover, we can loosely interpret $m_{X_i \to f_s}\left(x_i\right)$ and $m_{f_s \to X_i}\left(x_i\right)$ as the prior and likelihood message, respectively.

Let us suppose that each likelihood message $m_{f_s \to X_i}\left(x_i\right)$ has a complex form, e.g. a mixture of multiple Gaussian distributions. Then the computational complexity to evaluate the exact beliefs over all variables would be infeasible. Instead of propagating exact likelihood message $m_{f_s \to X_i}\left(x_i\right)$, EP passes an approximate message $\tilde{m}_{f_s \to X_i}(x_i)$, where $\tilde{m}_{f_s \to X_i}(x_i)$ is obtained by using the projection operation as shown in the general process of EP. Moreover, $\tilde{m}_{f_s \to X_i}(x_i)$ is usually chosen from exponential family to make the problem tractable. Thus, the approximate belief in EP has the following form

$$b(x_i) \approx q(x_i) \propto \prod_{s \in N(X_i)} \tilde{m}_{f_s \to X_i}(x_i). \tag{3.31}$$

To show BP as a special case of EP, we further define the partial belief of a variable node as

$$b(x_i)^{\backslash f_s} = \frac{b(x_i)}{\tilde{m}_{f_s \to X_i}\left(x_i\right)} \propto \prod_{s' \in N(X_i) \backslash s} \tilde{m}_{f_{s'} \to X_i}\left(x_i\right), \tag{3.32}$$

and the partial belief of a factor node as

$$b(f_s)^{\backslash X_i} = \frac{b(f_s)}{\tilde{m}_{X_i \to f_s}\left(x_i\right)}, \tag{3.33}$$

where $b(f_s) = \prod_{j \in N(f_s)} \tilde{m}_{X_j \to f_s}\left(x_j\right)$ is define as the belief of the factor node f_s. By comparing to (3.27) and (3.28), the factor node message updating rule in EP can be written as

$$\begin{aligned}\tilde{m}_{f_s \to X_i}(x_i) &= \frac{\mathbf{Proj}\left(b(x_i)^{\backslash f_s} m_{f_s \to X_i}(x_i)\right)}{b(x_i)^{\backslash f_s}} \\ &= \frac{\mathbf{Proj}\left(b(x_i)^{\backslash f_s} \int_{\mathbf{x}_s \backslash x_i} f_s(\mathbf{x}_s)\, b(f_s)^{\backslash X_i}\right)}{b(x_i)^{\backslash f_s}}\end{aligned} \tag{3.34}$$

where the integral works over continuous variable. For discrete variable, one can simply replace integral with summation. Furthermore, the new belief $b(x_i)$ will be approximated as

$$b(x_i) \approx q_i(x_i) = \frac{b(x_i)^{\backslash f_s} \tilde{m}_{f_s \to X_i}(x_i)}{Z_i}, \tag{3.35}$$

where $Z_i = \int_{x_i} b(x_i)^{\backslash f_s} \tilde{m}_{f_s \to X_i}(x_i)$.

Now, if the integral in (3.34) is tractable (e.g. a linear Gaussian model) even without using the projection to approximate $m_{f_s \to X_i}$. Then $b(x_i)^{\backslash f_s}$ in (3.34) can be canceled. Finally, the factor node message update rule in EP reduces to the standard BP case.

3.2.3 Relationship With Other Variational Inference Methods

In this section, I will describe the relationship between EP and other variational inference algorithms, e.g. VB. The problem considered here is the same as that in Sections 2.1.1, where $\mathbf{y}$ are the observed data and $\mathbf{x}$ are some hidden variables. The Bayesian probabilistic model specifies the joint distribution $p(\mathbf{x}, \mathbf{y})$, where all the hidden variables in $\mathbf{x}$ are given prior distributions. The goal is to find the best approximation for the posterior distribution $p(\mathbf{x}|\mathbf{y})$. Let us take a look at the decomposition of the log joint distribution as follows

$$\log p(\mathbf{x}, \mathbf{y}) = \log p(\mathbf{x}|\mathbf{y}) + \log p(\mathbf{y}). \tag{3.36}$$

By rearranging (3.36) and taking the integral of the both sides of the rearranged equation with respect to a given distribution $q(\mathbf{x})$, we get the log model evidence

$$\begin{aligned}\log p(\mathbf{y}) &= \int q(\mathbf{x}) \log(p(\mathbf{y})) d\mathbf{x} \\ &= \int q(\mathbf{x}) \log(p(\mathbf{x}, \mathbf{y})) - \int q(\mathbf{x}) \log(p(\mathbf{x}|\mathbf{y})) d\mathbf{x},\end{aligned} \tag{3.37}$$

where $\int q(\mathbf{x}) d\mathbf{x} = 1$. Then, by reformatting (3.37), we get

$$\log p(\mathbf{y}) = \mathcal{L}(q(\mathbf{x})) + D_{KL}(q(\mathbf{x})||p(\mathbf{x})), \tag{3.38}$$

where we define

$$\mathcal{L}(q(\mathbf{x})) = \int q(\mathbf{x}) \log(\frac{p(\mathbf{x}, \mathbf{y})}{q(\mathbf{x})}) d\mathbf{x}, \tag{3.39}$$

$$D_{KL}(q(\mathbf{x})||p(\mathbf{x})) = \int q(\mathbf{x}) \log(\frac{q(\mathbf{x})}{p(\mathbf{x}|\mathbf{y})}) d\mathbf{x}. \tag{3.40}$$

Since $D_{KL}(q(\mathbf{x})||p(\mathbf{x}))$ is a nonnegative functional, $\mathcal{L}(q(x))$ gives the lower bound of $\log p(\mathbf{y})$. Then the maximization of the lower bound $\mathcal{L}(q(x))$ with respect to the distribution $q(\mathbf{x})$ is equivalent to minimizing $D_{KL}(q(\mathbf{x})||p(\mathbf{x}))$, which happens when $q(\mathbf{x}) = p(\mathbf{x}|\mathbf{y})$. However, working with the true posterior distribution $p(\mathbf{x}|\mathbf{y})$ may be intractable. Thus, we assume that the elements of $\mathbf{x}$ can be partitioned into M disjoint groups $\mathbf{x}_i$, $i = 1, 2, \cdots, M$. We then further assume that the factorization of the approximate distribution $q(\mathbf{x})$ with respect to these groups has the form

$$q(\mathbf{x}) = \prod_{i}^{M} q_i(\mathbf{x}_i). \tag{3.41}$$

Please note that the factorized approximation corresponds to the mean filed theory, which was developed in physics. Given the aforementioned assumptions, we now try to find any possible distribution $q(\mathbf{x})$ over which the lower bound $\mathcal{L}(q(\mathbf{x}))$ is largest. Since the direct maximization of (3.39) with respect to $q(\mathbf{x})$ is difficult, we instead to optimize (3.39) with respect to each of the factors in (3.41). By substituting (3.41) into (3.39), we get

$$\begin{aligned}\mathcal{L}(q(\mathbf{x})) &= \int q_j(\mathbf{x}_j)\left(\int \log\left(p(\mathbf{x},\mathbf{y})\right)\prod_{i\neq j} q_i(\mathbf{x}_i)d\mathbf{x}_i\right)d\mathbf{x}_j - \int q_j(\mathbf{x}_j)\log(q_j(\mathbf{x}_j))d\mathbf{x}_j + \text{const}\\ &= -\int q_j(\mathbf{x}_j)\log(\frac{q_j(\mathbf{x}_j)}{\tilde{p}(\mathbf{x}_j,\mathbf{y})})d\mathbf{x_j} + \text{const}\\ &= -D_{KL}(q_j(\mathbf{x}_j)||\tilde{p}(\mathbf{x}_j,\mathbf{y})) + \text{const},\end{aligned} \tag{3.42}$$

where we define $\tilde{p}(\mathbf{x}_j,\mathbf{y})$ as

$$\begin{aligned}\tilde{p}(\mathbf{x}_j,\mathbf{y}) &= \exp\left(\int \log\left(p(\mathbf{x},\mathbf{y})\right)\prod_{i\neq j} q_i(\mathbf{x}_i)d\mathbf{x}_i\right)\\ &= \exp\left(\mathbb{E}_{i\neq j}[\log p(\mathbf{x},\mathbf{y})]\right).\end{aligned} \tag{3.43}$$

Thus, if we keep all the factors $q_i(\mathbf{x}_i)$ for $i \neq j$ fixed, then the maximization of (3.42) with respect to $q_j(\mathbf{x}_j)$ is equivalent to the minimization of $D_{KL}(q_j(\mathbf{x}_j)||\tilde{p}(\mathbf{x}_j,\mathbf{y}))$. In practices, we need to initialize all the factors $q_i(\mathbf{x}_i)$ first, and then iteratively update each of the factor $q_j(\mathbf{x}_j)$ by minimizing the $D_{KL}(q_j(\mathbf{x}_j)||\tilde{p}(\mathbf{x}_j,\mathbf{y}))$, until the algorithm convergences.

Now we can see the key difference between EP and VB is the way to minimizing the KL divergence. The advantage of VB is that it provides a lower bound during each optimizing step, thus the convergence is guaranteed. However, VB may cause under-estimate for variance. In EP, minimizing $D_{KL}(p(\mathbf{x})||q(\mathbf{x}))$ is equivalent to the "moment matching", but convergence is not guaranteed. However, EP has a fix point and if it does converge, the approximation performance of EP usually outperforms VB.

CHAPTER 4

SNR ESTIMATION IN CHANNEL CODING

In this chapter, I will describe the two proposed adaptive LDPC decoding schemes, which can perform online SNR estimation, by incorporating PF and EP algorithm. First, I will introduce the proposed PF based approach, which extend BP algorithm to handle continues variables. Moreover, the proposed method was denoted by particle based BP (PBP) algorithm. Then, I will introduce the EP based approach.

4.1 Time-varying SNR Estimation using PBP over AWGN Channels

4.1.1 Noise Adaptive LDPC Decoding

The main idea of noise adaptive LDPC decoding with PF is illustrated in the factor graph of Fig. 4.1 with three regions, where all circle nodes denote variable nodes and all square nodes denote factor nodes. If an accurate estimation of the noise variance $v = \sigma^2$ is given, the standard BP algorithm can obtain a good decoding performance by exchanging messages within Region III. In Region III, factor node f_l^{III} with $l = 1, 2, \ldots, M$ connects the bit variable nodes X_i, $i = 1, 2, \ldots, N$ and takes into account the constraints imposed by the LDPC codes. The corresponding factor function of f_l^{III} is given by

$$f_l^{III}\left(\mathbf{x}_{f_l^{III}}\right) = \begin{cases} 1 & \text{if even number of 1's in arguments} \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

where $\mathbf{x}_{f_l^{III}}$ indicate all variables connecting to factor node f_l^{III}, that is $\mathbf{x}_{f_l^{III}} = (x_i | i \in \mathcal{N}(f_l^{III}))$, and $\mathcal{N}(f_l^{III})$ represents the set of neighbors' indices for a node f_l^{III}.

Now, assuming that the noise variance is changing with time, I can model this

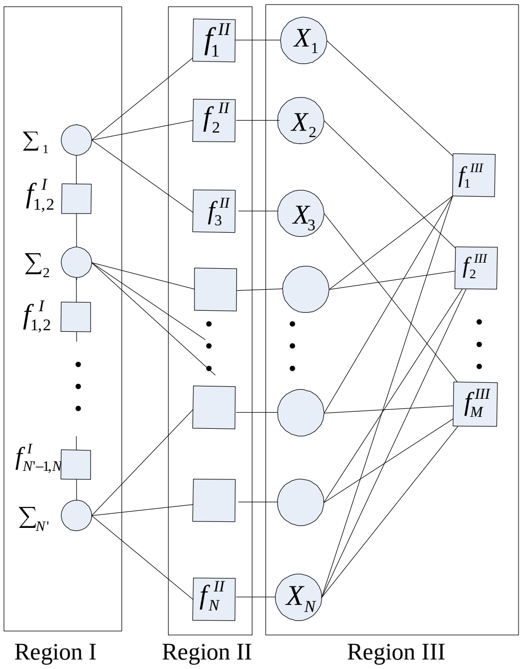

Figure 4.1: Factor graph of adaptive LDPC decoding based on PBP.

using extra variable nodes corresponding to $\Sigma_1, \Sigma_2, \ldots, \Sigma_{N'}$, which are shown as circles in Region I. For each variable $\sigma_{i'}$, the realization of variable node $\Sigma_{i'}$ in Region I, I model it with K particles, which are labeled as $\sigma^1_{i'}, \ldots, \sigma^K_{i'}$. Then these particles are used to estimate the noise variance with the PBP algorithm (see details in Section 4.1.2). Additionally, Region I and Region III are connected by factor nodes f^{II}_i in Region II. I call the number of factor nodes f^{II}_i connecting to each $\Sigma_{i'}$ the connection ratio, which is equal to three in Fig.4.1. The higher the connection ratio, the simpler the model and the fewer the number of hidden parameters [1]. The factor functions of f^{II}_i are defined as

$$f^{II}_i\left(\hat{x}_i, \sigma^k_{i'}; y_i\right) = \frac{1}{\sigma^k_{i'}} e^{-\frac{(y_i-\hat{x}_i)^2}{\left(\sigma^k_{i'}\right)^2}} \tag{4.2}$$

where y_i and $\hat{x}_i$ are the i-th input codeword and candidate codeword respectively,

[1] To estimate a constant noise variance, one can set the connection ratio equal to the code length N.

and the variables $i = 1, 2, \cdots, N$ and $k = 1, 2, \cdots, K$. Furthermore, the correlation between adjacent variable nodes is represented by additional factor node $f^I_{i',i'+1}$ in Region I, where the corresponding factor function is defined as

$$f^I_{i',i'+1}\left(\sigma^k_{i'}, \sigma^k_{i'+1}\right) = e^{-\frac{\left(\sigma^k_{i'+1} - \sigma^k_{i'}\right)^2}{\lambda_{i',i'+1}}} \tag{4.3}$$

where $\lambda_{i',i'+1}$ is a parameter to reflect the correlation between adjacent variable nodes. Generally, a small $\lambda_{i',i'+1}$ means a strong correlation, while a large $\lambda_{i',i'+1}$ reflects a weak/independent correlation. Moreover, the value of $\lambda_{i',i'+1}$ can be estimated simultaneously with decoding (see details in Section 4.1.4).

4.1.2 PBP Algorithm

In the standard BP algorithm [4], (2.9) is generally intractable when variables are continuous or the alphabet sizes of variables are large, since the summation in (2.9) will have an infinite number of terms. Thus, I introduce a PBP algorithm to solve this problem by combining BP with particle methods. The key idea of PBP is to model each continuous variable (or variable with large alphabet sizes) with K number of particles with associated weights, which just corresponds to K number of labels in the standard BP. Note that in standard BP only the belief of each label will be updated after each iteration, but in PBP both the value (i.e., location) of each label (i.e., particle) and the corresponding belief of each label will be updated after each iteration. Please note that these changes do not affect the sum-product message update rules described in the standard BP algorithm.

By introducing a distribution $W_{i'}(\sigma_{i'})$, which corresponds to the particle weights, we can rewrite (2.9) as an expectation form, which can be considered as importance-sampling transform of (2.9), as follows [2].

$$
\begin{aligned}
m_{f_i^{II} \to X_i}(x_i) &\propto \sum_{\sigma_{i'}} f_i^{II}(\sigma_{i'}, x_i) \frac{m_{\Sigma_{i'} \to f_i^{II}}(\sigma_{i'})}{W_{i'}(\sigma_{i'})} W_{i'}(\sigma_{i'}) \\
&\propto E\left[f_i^{II}(\sigma_{i'}, x_i) \frac{m_{\Sigma_{i'} \to f_i^{II}}(\sigma_{i'})}{W_{i'}(\sigma_{i'})} \right],
\end{aligned}
\tag{4.4}
$$

where E is the expectation with respect to the distribution $W_{i'}(\sigma_{i'})$. Then, the above message can be approximated by a list of K particles as

$$
\hat{m}_{f_i^{II} \to X_i}(x_i) \propto \frac{1}{K} \sum_{k=1}^{K} f_i^{II}\left(\sigma_{i'}^{(k)}, x_i\right) \frac{\hat{m}_{\Sigma_{i'} \to f_i^{II}}\left(\sigma_{i'}^{(k)}\right)}{W_{i'}(\sigma_{i'}^{(k)})}. \tag{4.5}
$$

Moreover, the distribution $W_{i'}(\sigma_{i'})$ can be chosen from the marginal distribution of variable $\sigma_{i'}$, which corresponds to the belief of this variable (see (2.10)). Additionally, locations and corresponding weights of particles have to be adjusted over time. This is achieved by using systematic resampling [44] and Metropolis-Hastings (MH) [45] random walk perturbation after each message update. The MH algorithm efficiently reduces the number of simulation iterations by half when comparing to the standard Gaussian random walk. In the following, the workflow of the PBP algorithm is described.

1. First, the weight of a particle $\sigma_{i'}^{(k)}$ will be computed as $b(\sigma_{i'}^{(k)})$, the belief of $\sigma_{i'}^{(k)}$ from standard BP, where $k = 1, 2, \cdots, K$.

2. Then K new samples, $\tilde{\sigma}_{i'}^{(1)}, \cdots, \tilde{\sigma}_{i'}^{(K)}$, will be drawn with probabilities proportional to $b(\sigma_{i'}^{(k)})$ using systematic resampling [44]. As a result, some $\sigma_{i'}^{(k)}$ that have small probabilities will be likely to be discarded whereas those with high probability will be repeatedly drawn.

[2] For ease exposition, I consider a factor function $f_i^{II}(\sigma_{i'}, x_i)$ with only two variables in this analysis. It is easy to extend the analysis to a factor function with more variables.

3. To maintain the diversity of the particles, the particle locations will be perturbed by an MH [45] based Gaussian random walk, which consists of two basic stages. First, let the proposed new K particles at each iteration be $\hat{\sigma}_{i'}^{(k)} = \tilde{\sigma}_{i'}^{(k)} + Z_r$, that is the current value plus a Gaussian random variable $Z_r \sim N(0, \sigma_r^2)$. Second, decide whether the proposed values of new particles are rejected or retained by computing the acceptance probability $a\{\hat{\sigma}_{i'}^{(k)}, \tilde{\sigma}_{i'}^{(k)}\} = \min\{1, \frac{p\left(\hat{\sigma}_{i'}^{(k)}\right)}{p\left(\tilde{\sigma}_{i'}^{(k)}\right)}\}$, where $\frac{p\left(\hat{\sigma}_{i'}^{(k)}\right)}{p\left(\tilde{\sigma}_{i'}^{(k)}\right)}$ is the ratio between the proposed particle value and the previous particle value. When the proposed value has a higher posterior probability than the current value $\tilde{\sigma}_{i'}^{(k)}$, it is always accepted; otherwise, it is accepted with probability a.

4. Based on the new particles, update messages and beliefs using standard BP.

5. Iterate steps 2 to 4 unless the maximum number of iterations is reached or other exit conditions are satisfied.

4.1.3 Noise Model

As stated above, I consider the case that the noise variance σ^2 varies continuously over time. In [46], the authors assume that the noise variance varies sinusoidally. In many other scenarios [47, 48], such as OFDM and CDMA systems, the noise variance is changing as a random variable with a predetermined probability density function (PDF). Here, I assume that the noise variance is Chi-square distributed with R degrees of freedom, each of which is a Gaussian distribution with zero mean and variance σ_h^2. The PDF of the noise variance is:

$$p(\sigma^2) = \frac{2^{-R/2}\sigma_h^{-R}}{\Gamma(R/2)}(\sigma^2)^{\frac{R}{2}-1}e^{-\sigma^2/2\sigma_h^2}, \tag{4.6}$$

where $\Gamma(\cdot)$ denotes the Gamma function [49]. For various scenarios, the corresponding PDF of noise variance can be obtained by adapting different values of R and σ_h^2.

4.1.4 Estimation of Parameter $\lambda_{i',i'+1}$

Generally, $\lambda_{i',i'+1}$ is taken as a predetermined value to simplify the problem. It may be beneficial to estimate $\lambda_{i',i'+1}$ for each factor node $f^I_{i',i'+1}$ in Region I to improve decoding performance. In my study, I utilize a similar method used for σ^2 estimation (see section 4.1.2) to estimate $\lambda_{i',i'+1}$ by sampling K particles $\lambda^1_{i,i+1}, \cdots, \lambda^K_{i',i'+1}$, for each factor node $f^I_{i',i'+1}$ in Region I. Here, I suppose that the change in $\lambda_{i',i'+1}$ has the same trend as the difference $\Delta\sigma_{i',i'+1} = |\bar{\sigma}_{i'} - \bar{\sigma}_{i'+1}|$ between the variable node of $\sigma_{i'}$ and $\sigma_{i'+1}$, where $\bar{\sigma}_{i'} = \frac{1}{K}\sum_{k=1}^{K}\sigma^k_{i'}$ is the mean of all the particles in variable node of $\sigma_{i'}$. A larger $\Delta\sigma_{i',i'+1}$ means a greater probability of $\lambda^k_{i',i'+1}$ to take a larger value. Thus, the weight of particles sampled for factor node $f^I_{i',i'+1}$ in Region I is defined as

$$\omega_{i',i'+1}\left(\lambda^k_{i',i'+1}\right) \propto \exp\left(-\frac{(\lambda^k_{i',i'+1})^2}{(\bar{\sigma}_{i'} - \bar{\sigma}_{i'+1})^2}\right). \tag{4.7}$$

Then $\lambda_{i',i'+1} = \frac{1}{K}\sum_{k=1}^{K}\lambda^k_{i',i'+1}$ and $\sigma^k_{i'}$ can be estimated alternately. To increase the stability, I perform one $\lambda_{i',i'+1}$ estimation for every T number of iterations whereas σ^2 estimation is performed at each iteration as described in Section 4.1.2.

4.2 Results and Discussion

In this section, the decoding performances (in terms of bits error rate (BER)) of standard BP and PBP decoders for LDPC codes were presented in the presence of a SNR mismatch. For the SNR mismatch, I considered two different scenarios, constant SNR mismatch and time varying SNR mismatch over an AWGN channel.

In my simulation, irregular LDPC codes with code rate of $\frac{1}{3}$ and degree profile $(\lambda^\star, \rho^\star)$ [50] were used, where the degree profile $(\lambda^\star, \rho^\star)$ was given by,

$$\begin{aligned}\lambda^\star = {} & 0.216724x^1 + 0.164615x^2 + 0.106047x^5 + 0.0935029x^6 \\ & + 0.000689685x^{12} + 0.0153518x^{13} + 0.0272307x^{14} \\ & + 0.00743584x^{15} + 0.0882668x^{16} + 0.0180324x^{32} \\ & + 0.0942067x^{33} + 0.000367395x^{40} + 0.16753x^{99}\end{aligned}$$

and

$$\rho^\star = 0.8x^6 + 0.2x^7.$$

Furthermore, the connection ratio is equal to 1. For each variable node in Region I, 16 particles were used. The initial value of $\lambda_{i',i'+1}$ was equal to 0.01, and then it was estimated online using the proposed algorithm, where the parameter T was equal to 10. All the results were obtained by averaging 10,000 different codewords and within 200 BP iterations.

In the experiments, first, I studied the decoding performance of the proposed PBP decoder, where the SNR was constant within each codeword block. In Fig. 4.2, the codeword block lengths of 10^3, 10^4, 10^5 were studied for LDPC codes, where the initial SNRs for BP and PBP decoders were the true SNR and $-2\ dB$ away from the true SNR, respectively. For LDPC codes, simulation results of different codeword block lengths showed no obvious degradation of performance between the proposed PBP decoder and the known SNR BP decoder. Also, the decoding performance of Turbo codes using online estimation [1] was compared with the proposed PBP decoder for LDPC codes. Fig. 4.2 showed that both of the online estimators for

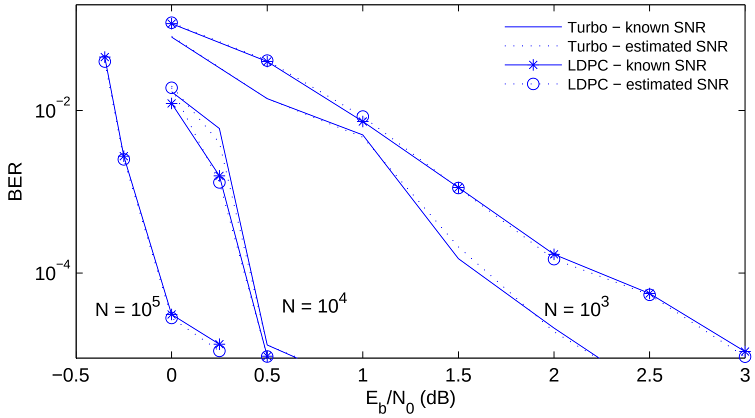

Figure 4.2: Performance of BER versus SNR for Turbo decoder and LDPC decoder using 1) the SNR determined by online estimation and 2) knowledge of the true SNR, where the SNR for each case was constant within a codeword block. The codeword block lengths of 10^3, 10^4, 10^5 were studied for LDPC codes. The results of Turbo codes were from [1].

Turbo codes and the proposed PBP decoder for LDPC codes manage to avoid the decoding performance degradation caused by SNR mismatch. Furthermore, the decoding performances of both Turbo and LDPC codes improves as the codeword block lengths are increased. When the codeword block length was larger than 10^4, the performances of LDPC codes became better than Turbo codes, which was also observed in [50].

Secondly, I studied the time varying SNR mismatch case. I assumed that the noise variance σ^2 satisfied a Chi-square distribution with $R = 2$ degrees of freedom and variance σ_h^2. Additionally, I assumed that every 100 successive bits in each codeword shared the same noise variance, which was sampled from a given Chi-square distribution. Then different Chi-square distributions could be obtained by varying σ_h^2. In Fig. 4.3, the solid line showed the sampled values of time varying noise variances for a codeword block with length 10^4, where σ_h^2 was equal to 1.6. The dotted line showed the estimation result using the proposed PBP decoder. Furthermore, the initial value $\hat{\sigma}$, used for estimation, was always equal to the mean

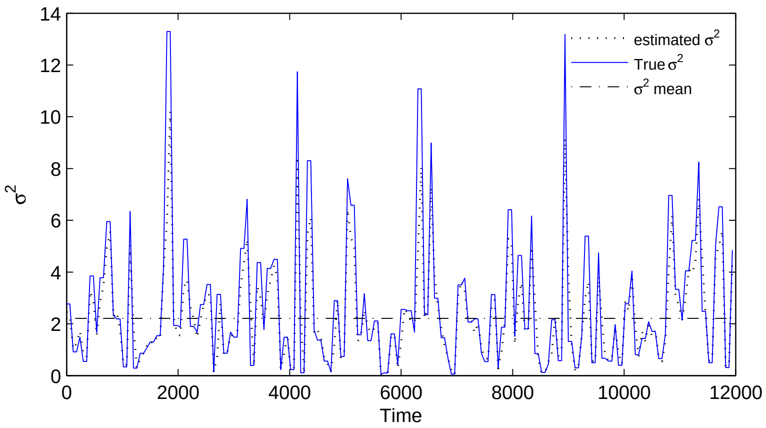

Figure 4.3: Estimation of time-varying σ^2 using the proposed PBP decoder, where a Chi-square distribution with $\sigma_h^2 = 1.6$ was used.

of sampled noise variances, which was shown by a dash dotted line. An accurate estimation of the noise variance σ^2 was found in Fig. 4.3, although the initial value used for estimation was far away from the true σ^2.

Finally, I investigated the decoding performances of BP decoder and PBP decoder with time varying SNR. By changing σ_h^2 from 0.5 to 2.3, different noise variance sequences with different mean values were sampled from the corresponding Chi-square distribution. These mean values were then used as initial values in the PBP decoder. Fig. 4.4 showed that the proposed PBP decoder obtained a much better performance than the known mean of time-varying SNR BP decoder. The gap between BP with and without the knowledge of true SNR was about 4 dB, however, the gaps between a known true SNR BP decoder and a PBP decoder were less than 0.5 dB and 0.1 dB at 10^{-4} and 10^{-5} BER levels, respectively. This result indicated that knowing only the mean of the time-varying SNR was not enough for a standard BP decoder to achieve its best decoding performance, if the SNR in a channel varied in bit-level. Moreover, in Fig. 4.4, PBP using MH showed a faster convergence speed and obtained about 0.1 dB performance gain compared with the normal PBP decoder.

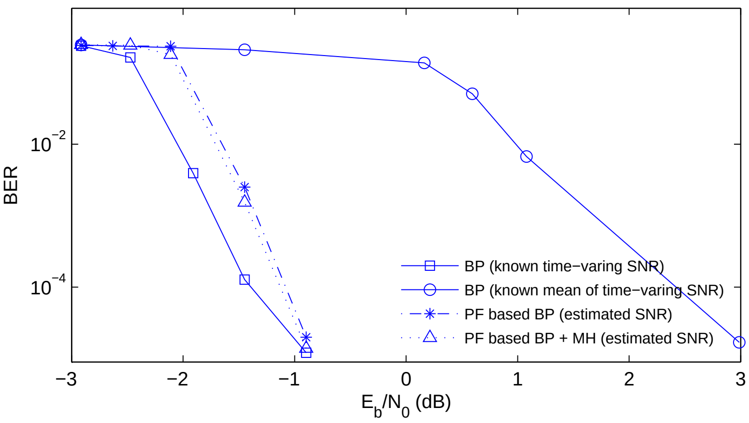

Figure 4.4: Error probability for LDPC codes over a time-varying AWGN channel with Chi-square distributed σ^2.

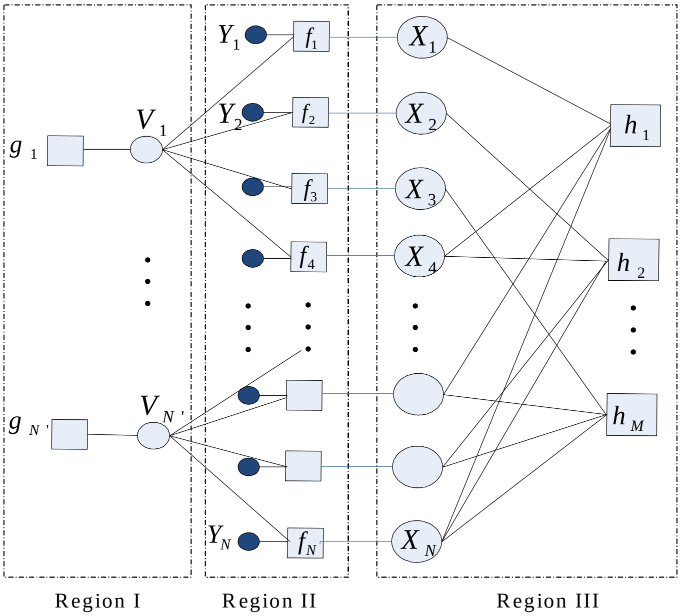

Figure 4.5: Factor graph of adaptive LDPC decoding based on EP.

4.3 Time-varying SNR Estimation using EP over AWGN Channels

In this section, I will introduce adaptive LDPC decoding using EP algorithm. Similar to Section 4.1.1 for the factor graph construction, to enable the online estimation of time-varying noise variance v_t, I introduce extra variable nodes V_j and factor nodes g_j, $j = 1, 2, \ldots, N'$ (see Region I of Fig. 4.5). Here, the connection ratio C, which is the number of factor nodes in Region II connecting to each variable node V_j, is equal to four in Fig. 4.5. In Region I, each variable node V_j is used to model the time-varying noise variance v_t of a block of C number of code bits. Moreover, the factor function $g_j(v_j)$ of factor node g_j corresponds to the *a priori* distribution for variable v_j and will be discussed in details later. Consequently, by introducing noise variance estimation in Region I, likelihood factor function is defined as

$$f_i\left(y_i; x_i, v_j\right) = \frac{1}{\sqrt{2\pi v_j}} \exp\left(-\frac{\left(y_i - x_i\right)^2}{2v_j}\right). \tag{4.8}$$

In Bayesian inference, the estimation of noise variance v_j corresponds to the estimation of its posterior distribution, i.e. $p(v_j|\mathbf{y}_j)$, where $\mathbf{y}_j = (y_i|i \in \mathcal{N}^{\backslash g_j}(V_j))$, and $\mathcal{N}^{\backslash g_j}(V_j)$ represents the set of neighbors' indices for a variable node V_j except the index of g_j. For the LDPC-based codes, it is easy to verify that

$$\begin{aligned} p(v_j|\mathbf{y}_j) &= \frac{\prod_{i \in \mathcal{N}^{\backslash g_j}(V_j)} p(v_j) p(y_i|v_j)}{\int_{v_j} \prod_{i \in \mathcal{N}^{\backslash g_j}(V_j)} p(v_j) p(y_i|v_j)} \\ &= \frac{\prod_{i \in \mathcal{N}^{\backslash g_j}(V_j)} \int_{x_i} p(v_j) p(x_i) p(y_i|x_i; v_j)}{\int_{v_j} \prod_{i \in \mathcal{N}^{\backslash g_j}(V_j)} \int_{x_i} p(v_j) p(x_i) p(y_i|x_i; v_j)}, \end{aligned} \tag{4.9}$$

where $p(v_j)$, the *a priori* distribution for v_j, is modeled by the factor function $g_j(v_j)$; $p(y_i|x_i; v_j)$, the likelihood for y_i, is modeled by the factor function $f(y_i; x_i, v_j)$; $p(x_i)$, the *a priori* distribution for code bit x_i, is captured by the message $m_{X_i \to f_i}(x_i)$ defined in [4]. Moreover, for binary sources in AWGN channels, variable x_i only takes ±1. Then the posterior distribution (4.9) can be written as

$$\begin{aligned} p(v_j|\mathbf{y}_j) &= \frac{1}{Z_j} g(v_j) \prod_{i \in \mathcal{N}^{\backslash g_j}(V_j)} \sum_{x_i \in \pm 1} f(y_i; x_i, v_j) m_{X_i \to f_i}(x_i) \\ &= \frac{1}{Z_j} m_{g_j \to V_j}(v_j) \prod_{i \in \mathcal{N}^{\backslash g_j}(V_j)} m_{f_i \to V_j}(v_j), \end{aligned} \tag{4.10}$$

where Z_j is a normalization constant, the value of message $m_{X_i \to f_i}(x_i)$ is updated iteratively by variable node X_i in Region III according to BP update rule (2.8), message $m_{g_j \to V_j}(v_j) = g(v_j)$ comes from prior factor node in Region I, and message $m_{f_i \to V_j}(v_j) = \sum_{x_i \in \pm 1} f(y_i; x_i, v_j) m_{X_i \to f_i}(x_i)$ comes from likelihood factor node in Region II according to the BP update rule (2.9).

Regarding to the selection of the prior function $g(v_j)$, it is reasonable to choose a conjugate prior of the likelihood function in (4.8) to make the approximation tractable. Since inverse gamma (IG) distribution is the conjugate prior to a likelihood distribution with variance as the parameter, I choose $g(v_j) = IG(v_j, \alpha_j^0, \beta_j^0)$ as the prior distribution for the likelihood function in (4.8). The IG distribution is defined as

$$IG(v_j, \alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} v_j^{-\alpha-1} \exp\left(-\frac{\beta}{v_j}\right), \tag{4.11}$$

where α and β are shape and scale parameters, respectively. Moreover, with respect to v_j, (4.8) can be written as

$$f_i\left(y_i; x_i, v_j\right) = \frac{1}{\sqrt{2\pi}} \frac{\Gamma(-\frac{1}{2})}{\left(\frac{(y_i - x_i)^2}{2}\right)^{-\frac{1}{2}}} IG\left(v_j, -\frac{1}{2}, \frac{(y_i - x_i)^2}{2}\right). \tag{4.12}$$

To calculate the posterior distribution (4.10), I cannot use BP directly, since the belief state for v_j is a mixture of 2^C IG distributions, and C can be a large number (e.g. 50 to 10,000 in my study), where $C = |\mathcal{N}^{\setminus g_j}(V_j)|$ is the connection ratio. Fortunately, EP [41] provides a fast and accurate approximation method by extending ADF to incorporate iterative refinement of approximations. Originally, EP algorithm in [41] is proposed to approximate a mixture of Gaussian distributions. In Section 4.3.1, I will extend EP algorithm to also handle a mixture of IG distributions, so that it can estimate the channel noise variance.

4.3.1 Posterior Approximation using EP

As mentioned in Section 4.3, to capture the time-varying noise variance of the AWGN channel while decoding, I introduced Region I (see Fig. 4.5) in the proposed factor graph and derived the posterior distribution for the time-varying noise variance. In this section, I propose a scheme based on the EP framework to speedily and accurately approximate the posterior distribution.

The key idea of EP is to sequentially compute approximate message $\tilde{m}_{f_i \to V_j}(v_j)$ in replace of true message $m_{f_i \to V_j}(v_j)$ in (4.10), then get a posterior on v_j by combining these approximations together. The formula of EP for the problem of variance estimation is given as follows:

1. Initialize the term approximation $\tilde{m}_{g_j \to V_j}(v_j)$ and $\tilde{m}_{f_i \to V_j}(v_j)$.

2. Compute the posterior approximation for v_j as:

$$q(v_j) = \frac{1}{Z_j} \tilde{m}_{g_j \to V_j}(v_j) \prod_{i \in \mathcal{N}^{\setminus g_j}(V_j)} \tilde{m}_{f_i \to V_j}(v_j), \tag{4.13}$$

   where $Z_j = \int_{v_j} \tilde{m}_{g_j \to V_j}(v_j) \prod_{i \in \mathcal{N}^{\setminus g_j}(V_j)} \tilde{m}_{f_i \to V_j}(v_j)$ is a normalization factor.

3. Until all $\tilde{m}_{f_i \to V_j}(v_j)$ converge:

   For each variable node V_j:

   For each factor node f_i, where $i \in \mathcal{N}^{\setminus g_j}(V_j)$

   (a) Remove the approximate message $\tilde{m}_{f_i \to V_j}(v_j)$ from the posterior approximation $q(v_j)$ to generate

$$q^{\setminus f_i}(v_j) \propto \frac{q(v_j)}{\tilde{m}_{f_i \to V_j}(v_j)}. \tag{4.14}$$

(b) To update $q(v_j)$, combine $q^{\backslash f_i}(v_j)$, the current message $m_{f_i \to V_j}(v_j)$ and the normalization constant Z_j to get a complex posterior $\hat{p}(v_j)$. Then minimize the KL-divergence $D(\hat{p}(v_j)||q(v_j))$ by performing moment matching ($\mathbf{Proj}(\cdot)$) (see Section 4.3.3 for more detail). Thus,

$$q(v_j) = \mathbf{Proj}(\frac{1}{Z_j} q^{\backslash f_i}(v_j) m_{f_i \to V_j}(v_j)), \tag{4.15}$$

where $Z_j = \int_{v_j} q^{\backslash f_i}(v_j) m_{f_i \to V_j}(v_j)$.

(c) Set approximate message

$$\tilde{m}_{f_i \to V_j}(v_j) = \frac{Z_j q(v_j)}{q^{\backslash f_i}(v_j)}. \tag{4.16}$$

4.3.2 IG Distribution Approximation using EP

Each iteration of EP based IG distribution approximation for the variance estimation problem proceeds as follows:

1. Initialize the prior messages for the noise variance variables

$$\tilde{m}_{g_j \to V_j}(v_j) = z_j^0 v_j^{-\alpha_j^0 - 1} \exp(-\frac{\beta_j^0}{v_j}) \tag{4.17}$$

with $\alpha_j^0 = 1$, $\beta_j^0 = v^0(\alpha_j^0 + 1)$ and $z_j^0 = \frac{\beta_j^{0\,\alpha_j^0}}{\Gamma(\alpha_j^0)}$, where v^0 is the initial variance for LDPC decoding, and β_j^0 and α_j^0 are scale and shape parameters for IG distribution, respectively[3]

[3] The selection of the initial values for the parameters guarantees the mode of prior distribution equals to v^0. Here, for a given v^0, a larger value of the shape parameter α_j^0 means a sharper prior distribution.

2. Initialize the likelihood messages from the channel output

$$\tilde{m}_{f_i \to V_j}(v_j) = z_{ij} v_j^{-\alpha_{ij}-1} \exp(-\frac{\beta_{ij}}{v_j}) \tag{4.18}$$

   with $\beta_{ij} = 0$, $\alpha_{ij} = -1$ and $z_{ij} = 1$, where the values selection for the above parameters guarantee that v_j in (4.18) is equality likely before LDPC decoding.

3. Initialize the posterior probability distributions of the noise variance variables

$$q(v_j) = Z_j v_j^{-\alpha_j^{\text{new}}-1} \exp(-\frac{\beta_j^{\text{new}}}{v_j}) \tag{4.19}$$

   with $\alpha_j^{\text{new}} = \alpha_j^0$, $\beta_j^{\text{new}} = \beta_j^0$ and $Z_j = z_j^0$.

4. Until all $\tilde{m}_{f_i \to V_j}(v_j)$ converge:

   For each variable node V_j

   For each factor node f_i, where $i \in \mathcal{N}^{\setminus g_j}(V_j)$

   (a) Remove $\tilde{m}_{f_i \to V_j}(v_j)$ from the posterior $q(v_j)$

$$\alpha_j^{\text{tmp}} = \alpha_j^{\text{new}} - (\alpha_{ij} + 1); \quad \beta_j^{\text{tmp}} = \beta_j^{\text{new}} - \beta_{ij}. \tag{4.20}$$

   (b) Update α_j^{new} and β_j^{new} according to moment matching. (See Section 4.3.3 for more details.)

   (c) Set approximate message

$$\begin{aligned} \alpha_{ij} &= \alpha_j^{\text{new}} - (\alpha_j^{\text{tmp}} + 1); \quad \beta_{ij} = \beta_j^{\text{new}} - \beta_j^{\text{tmp}} \\ z_{ij} &= Z_j \frac{{\beta_j^{\text{new}}}^{\alpha_j^{\text{new}}}}{\Gamma(\alpha_j^{\text{new}})} \frac{\Gamma(\alpha_j^{\text{tmp}})}{{\beta_j^{\text{tmp}}}^{\alpha_j^{\text{tmp}}}} \frac{\Gamma(\alpha_{ij})}{{\beta_{ij}}^{\alpha_{ij}}}. \end{aligned} \tag{4.21}$$

4.3.3 Moment Matching

By the technique of moment matching [51], $q(v_j)$ is obtained by matching the mean and variance of $q(v_j)$ to those of $\hat{p}(v_j)$. Then I get α_j^{new} and β_j^{new}, the parameters of $q(v_j)$. For the ease of exposition, I simplify the notations and let $\alpha = \alpha_j^{\text{new}}$, $\beta = \beta_j^{\text{new}}$, $\alpha' = \alpha_j^{\text{tmp}}$, $\beta' = \beta_j^{\text{tmp}}$, $v = v_j$, $\hat{p}(v) = \frac{1}{Z_j} q^{\setminus V_j}(v_j) m_{f_i \to V_j}(v_j)$, $Z = Z_j = \int_{v_j} q^{\setminus V_j}(v_j) m_{f_i \to V_j}(v_j)$, and $q(v) = q(v_j)$.

The mean and variance of IG distribution are matched by the following updates

$$m_1 = \frac{Lr(x) + (\frac{A_{-1}}{A_1})^{-\alpha'+\frac{1}{2}}}{(\alpha' - \frac{1}{2})\left(A_1^{-1} Lr(x) + (\frac{A_{-1}}{A_1})^{-\alpha'+\frac{1}{2}}(A_{-1})^{-1}\right)} \tag{4.22}$$

$$m_2 = \frac{Lr(x) + (\frac{A_{-1}}{A_1})^{-\alpha'+\frac{3}{2}}}{(\alpha' - \frac{1}{2})(\alpha' - \frac{3}{2})\left(A_1^{-2} Lr(x) + (\frac{A_{-1}}{A_1})^{-\alpha'+\frac{3}{2}}(A_{-1})^{-2}\right)} - (m_1)^2 \tag{4.23}$$

$$\alpha = \frac{(m_1)^2}{m_2} + 2 \qquad \beta = m_1 \left(\frac{(m_1)^2}{m_2} + 1\right) \tag{4.24}$$

where $A_1 = \beta' + \frac{(y-1)^2}{2}$ and $A_{-1} = \beta' + \frac{(y+1)^2}{2}$.

4.4 Results and Discussion

In this section, the decoding performances (in terms of bits error rate (BER)) of standard BP and EP based BP decoders are presented in the presence of a SNR mismatch. Here, I consider two different scenarios, constant SNR mismatch and time varying SNR mismatch over an AWGN channel.

I first study the performance of the proposed EP estimator for different values of constant channel SNR in Fig. 4.6. In this case, a $(3, 6)$-regular LDPC code with code length $N = 1000$ is used in the simulation. Since I consider the estimation of constant SNR, there is only one variable node in Region I, which means that the connection ratio C is set to 1000. All the results are obtained over 100 trials. Initial SNRs used for BP decoding are always -2 dB away from the true SNRs. Moreover,

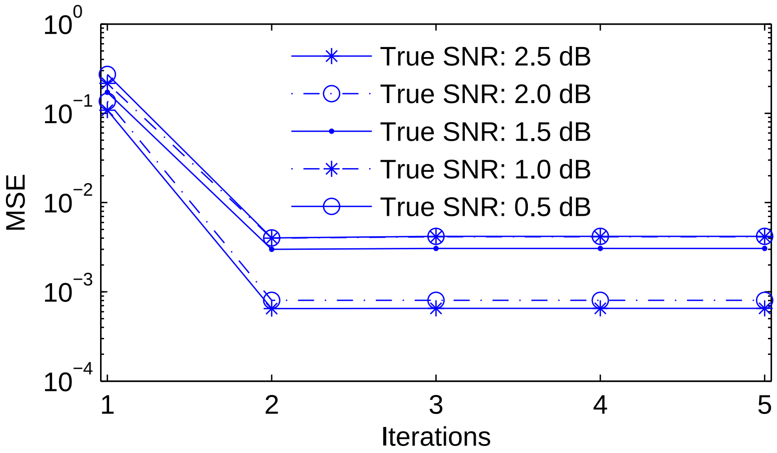

Figure 4.6: The performance of the EP estimator for different values of constant channel SNR.

the maximum number of iterations for BP decoding is $T_{\text{BP}} = 50$. Since I only focus on the study of estimation accuracy in this experiment, the EP estimator is only used once at the end of the BP decoding. To achieve the best decoding performance, new estimates can be sent back to LDPC decoder periodically, and this setup will be studied later in this section. In Fig. 4.6, it shows that the EP estimator can always converge within 3 or 4 iterations. Similar to the results in [1, 52], the estimation accuracy of the proposed EP estimator, in terms of MSE, increases as the true SNR increases. It is because that for a given LDPC code, a smaller true SNR may yield a larger number of BP decoding errors, which degrade the performance of the EP estimator.

Second, I study the performance of the proposed EP estimator for time-varying SNR in Fig. 4.7. In this section, since I consider an AWGN channel with PBSK modulation, the time-varying SNR is modeled as $10\log_{10}\left(\frac{1}{2v_tR}\right)$ [1] and $v_t = (\sigma_0 + N(0, \sigma^2_{\text{noise}}))^2$, where the code rate $R = 0.5$ is used in the simulation, σ_0 is a constant, and $N(0, \sigma^2_{\text{noise}})$, a Gaussian noise with parameter σ_{noise}, models the fluctuation on σ_0. Fig. 4.7 shows the estimate of a time-varying SNR, where $\sigma_0 = 0.8318$, $\sigma_{\text{noise}} = 0.1$, $N = 10,000$ and $C = 50$. Note that, other settings in this case are the same as the

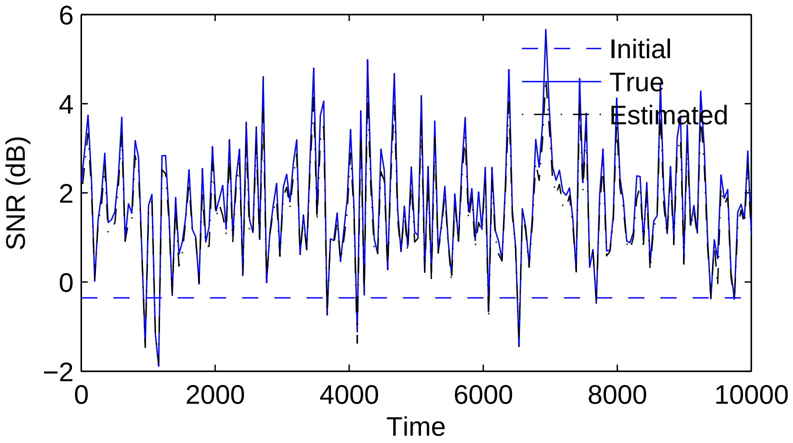

Figure 4.7: Estimation of time-varying SNR using the proposed EP estimator, where the initial SNR for BP decoding is −2 dB away from the mean of the time-varying SNR.

study in Fig. 4.6. It can be seen that the proposed EP estimator provides a precise estimate of the time-varying SNR, even though the initial SNR is far away from the mean of the time-varying SNR. Compared to [46], particle estimator usually needs a large number of iterations (> 200) to converge, however, the proposed EP estimator usually converges within 3 iterations. Moreover, the computation complexities of each iteration for both EP estimator and BP estimator are quite similar. Thus, the proposed EP estimator can approximate the correlation much more rapidly.

Third, in Fig. 4.8, I study the decoding performances of LDPC decoder with and without EP estimator. In this case, I assume that the channel SNR is constant for each simulation. The following results are obtained over 10,000 trials. The codeword lengths of 10^3 and 10^4 are studied, where the initial SNRs are -2 dB away from the true SNRs for both BP decoder and EP based BP decoders. In Fig. 4.8, a big performance gap between BP decoder and EP based BP decoder is observed in the presence of SNR mismatch. Moreover, the performance curve of BP decoder with knowledge of the true SNR is provided as the benchmark. Simulation results of different codeword block lengths showed no obvious degradation of performance

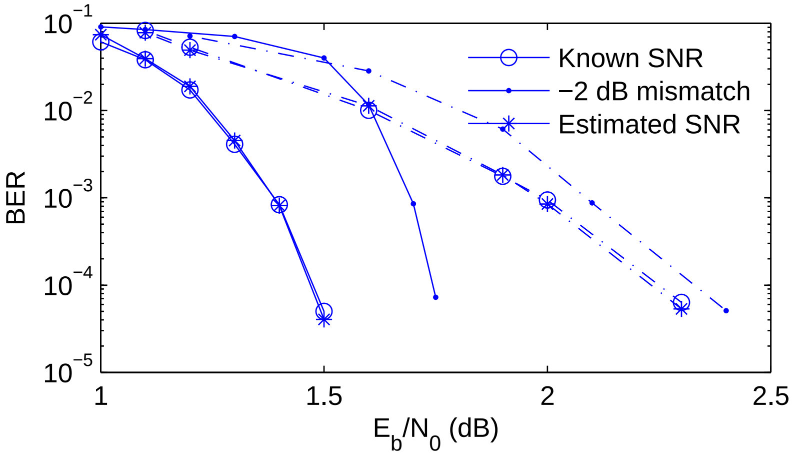

Figure 4.8: Performance of BER versus SNR for LDPC decoder using 1) knowledge of the true SNR, 2) knowledge of the initial SNR with -2 dB mismatch and 3) the SNR determined by EP estimator, where the SNR for each simulation is a constant. The codeword lengths of 10^3 (dashed-dot) and 10^4 (solid) were studied.

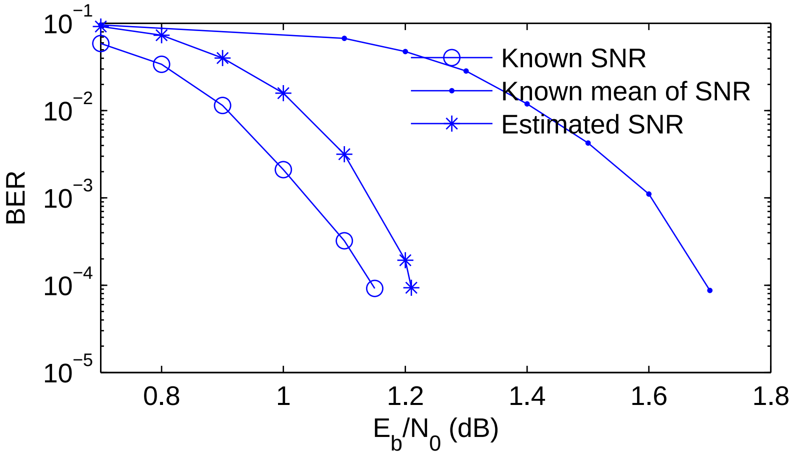

Figure 4.9: Performance of BER versus SNR for LDPC decoder using 1) knowledge of the true SNR, 2) knowledge of the initial SNR with the mean of the true SNR and 3) the SNR determined by EP estimator, where the time-varying SNR for each simulation varies according to the aforementioned model in the study of Fig. 4.7 by using different σ_0. The codeword length of 10^4 is studied.

between the proposed EP based BP decoder and the known SNR BP decoder.

Finally, I investigate the decoding performances of BP decoder and EP based BP decoder with time-varying SNR, where $\sigma_{\text{noise}} = 0.2$. By changing σ_0, different time-varying SNRs are sampled from the aforementioned model in the study of Fig.

4.7. The mean values of different time-varying SNRs used in this study change from 0.9 dB to 1.75 dB, and these mean values were then used as initial values in BP decoder and EP based BP decoder. In this case, the EP estimator starts working after 50 BP iterations. Moreover, I perform EP estimator periodically for every 20 BP iterations until BP decoder successfully decodes the codeword or reaches its maximum number of iterations (i.e. 150 in this experiment). Since EP and BP steps have similar complexity, the additional computational overhead of the proposed decoder is less than 10% of the standard BP decoder. Fig. 4.9 shows that the proposed EP based BP decoder obtained a much better performance than BP decoder with the known mean of time-varying SNR. The gap between BP with and without the knowledge of true SNR is about 0.55 dB, however, the gap between a known true SNR BP decoder and the EP based BP decoder is less than 0.06 dB. This result indicates that knowing only the mean of the time-varying SNR is not enough for a standard BP decoder to achieve its best decoding performance, if the SNR in a channel varies at the bit-level.

4.5 Summary

This chapter presents an adaptive LDPC decoding over stationary and time-varying AWGN channels by incorporating PF algorithm and EP algorithm, respectively. For both algorithms, not only the stationary, but also the time-varying channel SNR can be precisely tracked. Thus, the proposed algorithm is not sensitive to the initial estimation of the channel SNR, and therefore yields a better decoding performance (in terms of lower BER) than the standard BP algorithm. Moreover, the proposed EP estimator has the same decoding performance with PBP estimator, however, EP estimator shows a much faster convergence speed and lower computational complexity than PBP estimator.

CHAPTER 5

CORRELATION ESTIMATION IN DISTRIBUTED SOURCE CODING

DSC is a technique to compress correlated remote sources separately and decompress them jointly, which was first introduced by Slepian and Wolf [9] and then was studied by Wyner and Ziv [10, 27]. Due to the contributions of Slepian-Wolf and Wyner-Ziv, we usually refer lossless DSC to as an SW problem and lossy source coding with side information as a WZ problem. As mentioned in the introduction, since the decoding performance of DSC relies on the knowledge of correlation very much, the design of an online correlation estimation scheme becomes a significant task both in theoretical studies and practical applications. Moreover, it has been proven that channel coding can be used to implement DSC coding [53]. Thus in this chapter, I will discuss the correlation estimation in SW problem using PBP and EP, respectively, by extending the discussion in Chapter 4.

5.1 SW Decoding with Time-varying Correlation Estimation using PBP

5.1.1 PBP for Asymmetric SW Coding

In this section, I talk about adaptive SW decoding based on PBP for asymmetric SW case. The proposed approach is based on the syndrome based approach using LDPC code [54] as shown in Fig. 5.1 (see Regions II and III). At the encoder, a block of N input bits, $x_1, x_2, \cdots, x_N$, is compressed into M syndrome bits, $s_1, s_2, \cdots, s_M$, thus resulting in $M : N$ compression. The factor nodes f_l^{III}, $l = 1, 2, \cdots, M$ as shown in Region III of Fig. 5.1 take into account the constraint imposed by the received syndrome bits. For a factor node f_l^{III} in Region III, the corresponding factor function $f_l^{III}(\mathbf{x}_{f_l^{III}})$ is defined as

$$f_l^{III}(\mathbf{x}_{f_l^{III}}) = \begin{cases} 1, & \textit{If } s_l \oplus \bigoplus_{i \in \mathcal{N}(f_l^{III})} x_i = 0, \\ 0, & \text{otherwise,} \end{cases} \tag{5.1}$$

where $\mathcal{N}(f_l^{III})$ denotes the set of neighbors' indices of factor node f_l^{III}, $\bigoplus_{i \in N(f_l^{III})}$ represents the bitwise sum of all elements x_i with $i \in \mathcal{N}(f_l^{III})$, and for a factor node f_l^{III}, $\mathbf{x}_{f_l^{III}}$ indicate all variables connecting to f_l^{III}.

For the conventional SW coding, the correlation between a pair of sources, x_i and y_i, is handled by a correlation factor node f_i^{II}, $i = 1, 2, \cdots, N$ (see Region II of Fig. 5.1), where the corresponding factor function $f_i^{II}(y_i, x_i, p)$ is defined as

$$f_i^{II}(y_i, x_i, p) = \begin{cases} 1 - p, & \textit{If } x_i = y_i, \\ p, & \text{otherwise.} \end{cases} \tag{5.2}$$

With the variable and factor nodes defined and in place, one can estimate the values of $\mathbf{x}$ using the BP algorithm. While the source X can be compressed very closely to the SW limit $H(X|Y)$ in the classic BP approach [54], the crossover probability p is assumed to be constant and known *a priori*. The main contribution of the approach is to relax these constraints. Namely, I assume that p is unknown and varies slowly over time. To model this, I connect the factor node f_i^{II} to a variable $p_{i'}$, where $p_{i'}$ is now a variable instead of a constant. Thus, the factor function $f_i^{II}(y_i, x_i, p)$ in (5.2) will be updated to

$$f_i^{II}(y_i, x_i, p_{i'}) = \begin{cases} 1 - p_{i'}, & \text{if } x_i = y_i, \\ p_{i'}, & \text{otherwise.} \end{cases} \tag{5.3}$$

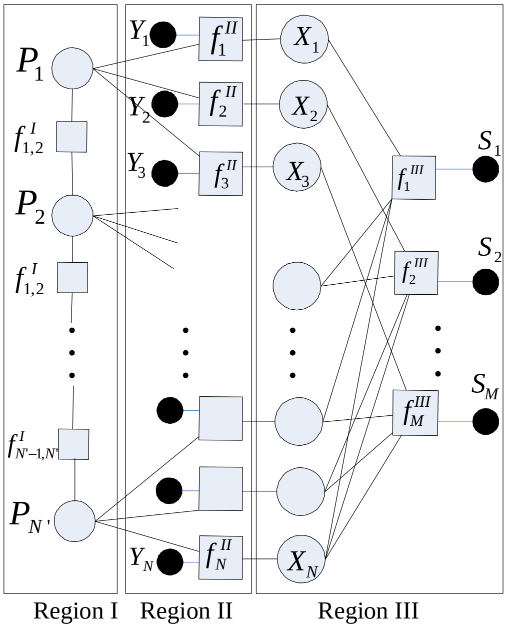

Figure 5.1: Factor graph representation of the proposed PBP algorithm, where the superscripts of f indicate the factor nodes in Regions I, II, III, respectively.

I call the number of correlation factor nodes connecting to each $p_{i'}$ the connection ratio, which is equal to three in Fig. 5.1.

Since I assume that p only varies slowly over time, the corresponding probability of any two variable nodes $P_{i'}$ and $P_{i'+1}$ in Region I, should be close. This is captured by the p-factor nodes $f^I_{1,2}, f^I_{2,3}, \cdots, f^I_{N'-1,N'}$ as shown in Region I of Fig. 5.1, where a p-factor function $f^I_{i',i'+1}(p_{i'}, p_{i'+1})$ is defined as

$$f^I_{i',i'+1}(p_{i'}, p_{i'+1}) = \exp\left(-\frac{(p_{i'} - p_{i'+1})^2}{\lambda_{i',i'+1}}\right), \tag{5.4}$$

where the process of estimating parameters $\lambda_{i',i'+1}$ is the same as that in Section 4.1.4.

With the factor functions defined in (5.1), (5.3), and (5.4), it may appear that the BP algorithm can be directly applied. However, $p_1, p_2, \cdots, p_{N'}$ are continuous and cannot be handled by standard BP. Nevertheless, by applying PBP, it is able to handle even continuous variables.

As described in Section 4.1.2, PBP can handle continuous variables by sampling a list of particles. Corresponding to the factor graph in Fig. 5.1 of this section, PBP is used to model each $p_{i'}$ in Region I with K particles $p_{i'}^{(1)}, \dots, p_{i'}^{(K)}$ and adjust particle locations and weights according to systematic resampling and MH random walk. Region II plays the role of connecting standard BP (Region III) and PBP (Region I) to exchange information between each other. The factor node message update from Region II to Region I can be written as

$$m_{f_i^{II} \to P_{i'}}(p_{i'}^{(k)}) \propto \sum_{x_i \in \{0,1\}} f_i^{II}\left(y_i, x_i, p_{i'}^{(k)}\right) m_{X_i \to f_i^{II}}(x_i), \tag{5.5}$$

while the factor node message update from Region II to Region III can be written as

$$m_{f_i^{II} \to X_i}(x_i) \propto \frac{1}{K} \sum_{k=1}^{K} f_i^{II}\left(y_i, x_i, p_{i'}^{(k)}\right) \frac{m_{P_{i'} \to f_i^{II}}\left(p_{i'}^{(k)}\right)}{W_{i'}(p_{i'}^{(k)})}, \tag{5.6}$$

where $f_i^{II}(y_i, x_i, p_{i'}^{(k)}) = \begin{cases} 1 - p_{i'}^{(k)}, & \text{if } x_i = y_i \\ p_{i'}^{(k)}, & \text{otherwise} \end{cases}$, and $W_{i'}(p_{i'}^{(k)})$ corresponds to the belief of particle $p_{i'}^{(k)}$. On one hand, we can see that the message $m_{X_i \to f_i^{II}}(x_i)$ from Region III is used to update the message $m_{f_i^{II} \to P_{i'}}(p_{i'}^{(k)})$ to Region I. Furthermore, the updated message in Region I can be used to update the value of each particle according to the belief $b\left(p_{i'}^{k}\right) \propto \prod_{f_i^{II} \in N(i')} m_{f_i^{II} \to P_{i'}}\left(p_{i'}^{(k)}\right)$. On the other hand, for Region III, not only the message from $m_{P_{i'} \to f_i^{II}}\left(p_{i'}^{(k)}\right)$ is used to update the message $m_{f_i^{II} \to X_i}(x_i)$, but also, more importantly, the updated value of each particle $p_{i'}^{(k)}$, which corresponds to the crossover probability, has played a role for updating the message $m_{f_i^{II} \to X_i}(x_i)$. Actually, updating message $m_{f_i^{II} \to X_i}(x_i)$ equals to the update of estimate of source correlation. Finally, by performing the aforementioned scheme iteratively, the source decoding and correlation estimation can be done simultaneously.

Complexity Discussion

The complexity of BP increases linearly with the degree of a variable node but exponentially with the degree of a factor node. However, we can easily incorporate the "method" of passing log-likelihood ratios $L_{f_i^{III}X_i} \triangleq \log\frac{m_{f_i^{III}\to X_i}(0)}{m_{f_i^{III}\to X_i}(1)}$ instead of probabilities as messages to reduce the complexity for the factor node updates in Region III [55]. The resulting complexity will be linear with respect to code length [56]. Note that the same method cannot be used in general for factor nodes in Regions I and II since the method can only be used to variables with alphabet size of two and there are generally more than two labels for the variable there. For example, I generally use more than two particles to represent $p_{i'}$ (i.e., each $p_{i'}$ can take more than two values). However, this does not have a significant impact to the complexity of the overall algorithm since the node degrees of the factor nodes in Regions I and II are only two as shown in Fig. 5.1.

5.1.2 PBP for Non-asymmetric SW Coding

Different attempts have been made to implement non-asymmetric SW coding, which includes: time-sharing, source splitting [57], and code partitioning [14,26]. However, like all aforementioned work, they assume the correlation statistics between the two sources is constant and known *a priori*.

The code partitioning approach effectively converts a SW coding problem into a channel coding problem. In [14], the code partitioning approach is implemented using irregular repeated accumulat (IRA) codes [58], a special case of LDPC codes. Being a form of LDPC codes, the IRA based SW coding can be decoded using BP, and the proposed PBP method can be directly applied. For completeness, a brief description about code partitioning approach is given as follows.

Let $H = [P|I] = [P_1 P_2|I]$ be the parity check matrix of a systematic linear block code, where the widths of P_1 and P_2 are N_1 and N_2, respectively, and I is an identity matrix of size $M \times M$. Therefore, H is of size $M \times N$, where $N = N_1 + N_2 + M$.

Now, we can partition the code into two subcodes with parity check matrices $H_1 = \begin{bmatrix} I & 0 & 0 \\ 0 & P_2 & I \end{bmatrix}$ and $H_2 = \begin{bmatrix} 0 & I & 0 \\ P_1 & 0 & I \end{bmatrix}$. Length-$N$ blocks drawn separately from the two correlated sources, $\mathbf{x}$ and $\mathbf{y}$, will be compressed to $\mathbf{u} = H_1\mathbf{x}$ and $\mathbf{v} = H_2\mathbf{y}$, respectively. For the ease of explanation, let us split $\mathbf{x}$ into $\mathbf{x}^1$, $\mathbf{x}^2$ and $\mathbf{x}^{\mathbf{3}}$, where their lengths are N_1, N_2 and M, respectively, and split $\mathbf{u}$ into $\mathbf{u}^1$ and $\mathbf{u}^2$ with lengths N_1 and M. Therefore, we have $\mathbf{u}^1 = \mathbf{x}^1$ and $\mathbf{u}^2 = P_2\mathbf{x}^2 + \mathbf{x}^3$. Similarly, $\mathbf{y}$ is split into $\mathbf{y}^1$, $\mathbf{y}^2$ and $\mathbf{y}^{\mathbf{3}}$, and $\mathbf{v}$ is split into $\mathbf{v}^1$ and $\mathbf{v}^2$. This gives us $\mathbf{v}^1 = \mathbf{y}^2$ and $\mathbf{v}^{\mathbf{2}} = P_1\mathbf{y}^1 + \mathbf{y}^3$.

At the decoder, the received bits of $\mathbf{u}$ and $\mathbf{v}$ will be rearranged and padded with zeros into $\mathbf{t}^1 = \begin{bmatrix} \mathbf{u}^1 \\ 0_{N_2 \times 1} \\ \mathbf{u}^2 \end{bmatrix}$ and $\mathbf{t}^2 = \begin{bmatrix} 0_{N_1 \times 1} \\ \mathbf{v}^1 \\ \mathbf{v}^2 \end{bmatrix}$. Then, it can be easily verified that $\mathbf{t}' \triangleq \mathbf{t}^1 + \mathbf{t}^2 + \mathbf{x} + \mathbf{y} = \begin{bmatrix} I \\ P \end{bmatrix} \begin{bmatrix} \mathbf{x}^2 \\ \mathbf{y}^1 \end{bmatrix}$. Note that $\begin{bmatrix} I \\ P \end{bmatrix}$ is actually a generator matrix of the original code. Thus, $\mathbf{t}' = \mathbf{t}^1 + \mathbf{t}^2 + \mathbf{x} + \mathbf{y}$ is the codeword encoded from the message $\begin{bmatrix} \mathbf{x}^{\mathbf{2}} \\ \mathbf{y}^1 \end{bmatrix}$. We can rewrite $\mathbf{t}'$ as $\mathbf{t}' = \mathbf{t} + \mathbf{z}$, where $\mathbf{t} = \mathbf{t}^1 + \mathbf{t}^2$ and $\mathbf{z} = \mathbf{x} + \mathbf{y}$. Therefore, given $\mathbf{t}$ (corresponding to the side information used in asymmetric case), the decoder can recover $\mathbf{t}'$ by taking $\mathbf{t}$ as a corrupted codeword passing through a channel with noise $\mathbf{z}$. Given $\mathbf{x}^2$ and $\mathbf{y}^1$ (obtained from the decoded $\mathbf{t}'$), $\mathbf{x}^3$ and $\mathbf{y}^3$ can be solved accordingly from $\mathbf{u}_2 = P_2\mathbf{x}^2 + \mathbf{x}^3$ and $\mathbf{v}_2 = P_1\mathbf{y}^1 + \mathbf{y}^3$, whereas $\mathbf{x}^1$ and $\mathbf{y}^2$ can be read out from $\mathbf{u}_1$ and $\mathbf{v}_1$ directly. Finally, by combining all the decoded information, both sources $\mathbf{x}$ and $\mathbf{y}$ can be recovered. According to the aforementioned description, we can see the factor graphs used for the non-asymmetric case are the same as the asymmetric case, except replacing the side

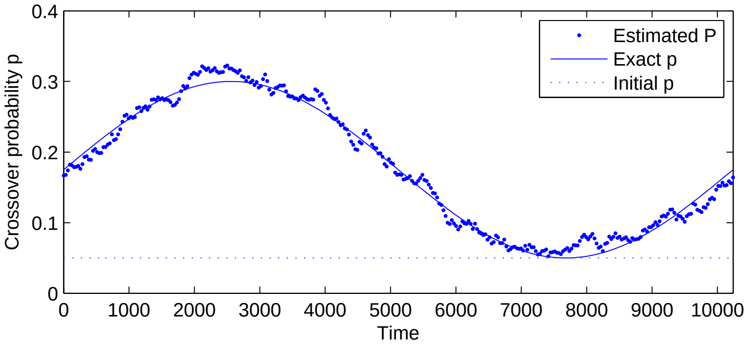

Figure 5.2: Estimation of crossover probabilities for sinusoidal changing correlations.

information $\mathbf{y}$ by $\mathbf{t}$, the decoding codeword $\mathbf{x}$ by $\mathbf{t}'$ and setting all the syndrome bits equal to 0. Then the inference problem for non-asymmetric case can be solved similarly as the asymmetric problem. More details about the implementation of non-asymmetric setup can be found in [35].

5.2 Results and Discussion

I first studied the asymmetric case, where SW codes were randomly generated by a 6000×10240 parity check matrix and the variable node degree is equal to 3. Moreover, 16 particles were assigned to each variable node in Region I. For the random walk step, I assumed $\sigma_r^2 = 0.0001$. The following results were obtained by averaging the estimated crossover probability of 200 different codewords. Fig. 5.2 shows the estimated results of a sinusoidally changing correlation, where the crossover probability p changes sinusoidally from 0.05 to 0.3 for each input codeword bit. The results verified that the proposed algorithm can generate a good estimation of a complexly changing correlation.

Next I analyzed how different settings of parameters effected the decoding performance of the proposed PBP algorithms. The following performance results were obtained by averaging 10000 independent simulations, where the code length was

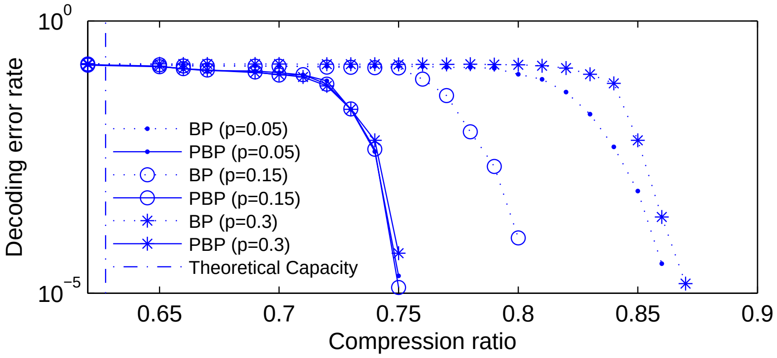

Figure 5.3: Decoding bit error rate for a sinusoidal changing correlation.

equal to 10240. Moreover, the theoretical capacity is calculated according to the equation $C = \frac{1}{N}\sum_{i=1}^{N} 1 - H(p_i)$, where $H(p_i) = -p_i \log(p_i) - (1-p_i)log(1-p_i)$ and $N = 10240$. The value of crossover probability p_i changed sinusoidally from 0.05 to 0.3 in the BSC. The number of particles was also equal to 16.

In Fig. 5.3, I compared the decoding performance between the proposed PBP algorithm and standard BP algorithm by using different initial estimations of p, namely, $p = 0.05, 0.15$ or 0.3. It shows that the gain was relatively small when the initial estimation is close to the true value of the crossover probability (e.g. $p = 0.15$, which was roughly equal to the mean of the time changing crossover probability). However, when the initial estimation was far away from the true value, the observed gain was significant. In comparison, we can see that the PBP algorithm is not sensitive to the initial estimation of p, since the results showed that all the PBP simulations yielded similar decoding performance.

I then proceeded to study the non-asymmetric case. I tried to compare the performance of the adaptive decoding algorithm with conventional IRA decoding. I fixed the code rates for both X and Y to be 0.75. I then compared decoding performance of the two schemes while varying the correlation parameter p. Unlike the first case, I let p to be a constant over all samples. Initial estimations of p were set 0.1 and 0.2. As shown in Fig. 5.4, the gain was relatively small when the

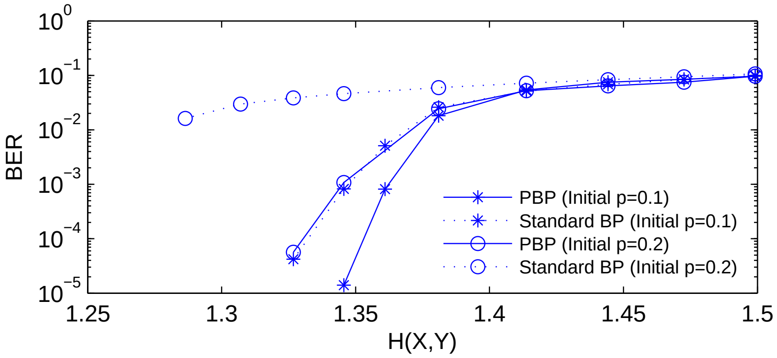

Figure 5.4: Change of BER as the correlation parameter p (and thus $H(X,Y)$) varies. Code length (N) $= 10,000$ and the rate pair is $(0.75, 0.75)$. The initial estimations of p are 0.1 and 0.2.

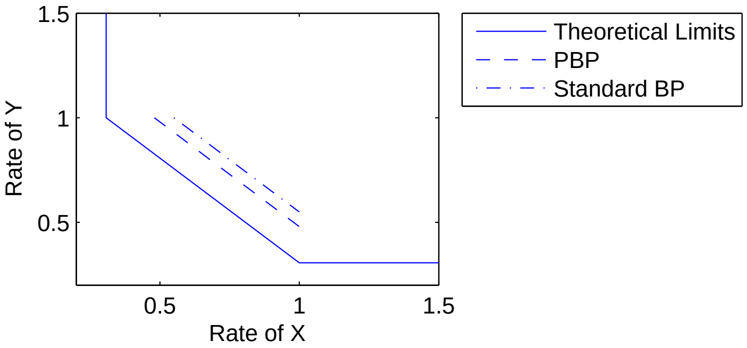

Figure 5.5: Comparing the proposed adaptive decoding algorithm and standard BP with the theoretical limits of SW code, where the correlation is changing sinusoidally with max value 0.07 and min value 0.05.

estimation p was not too far from its true value. However, when the estimation deviated significantly from its true value, the observed gain was substantial.

Finally, I compared the two algorithms for the case when there was some minor fluctuation in p, where p varies sinusoidally from 0.05 to 0.07. I approximated the sum rate where lossless compression was achieved when the probability of error fell below 10^{-4}. The result is shown in Fig. 5.5. We can see that the gain is rather significant even when the fluctuation of p is rather small.

5.3 WZ Decoding with Time-varying Correlation Estimation using PBP

5.3.1 WZ Problem

WZ coding [10] generalizes the SW setup in which coding of the source X is lossy with respect to a fidelity criterion rather than lossless. WZ coding can be treated as a degenerated case of DSC with two sources X and Y, where source Y is transmitted perfectly to the decoder and source X is then compressed and recovered with the help of the source Y acting as side information. For both discrete and continuous alphabets of X for the source X, side information Y and general distortion metrics $d\left(\cdot\right)$, the rate distortion function for this setup $R_{WZ}\left(D\right)$ is

$$R_{WZ}(D) = \inf I(U;X) - I(U;Y), \tag{5.7}$$

where U is an auxiliary random variable satisfying the Markov chain $Y \rightarrow X \rightarrow U$ and there exists a function $\hat{X} = \hat{X}(U,S)$ satisfying $E\left\{d\left(X,\hat{X}\right)\right\} \leq D$. Moreover, when the side information is available at both encoder and decoder, the rate-distortion function is

$$R_{X|Y}(D) = \inf_{\hat{X}\in X:Ed(X,\hat{X})\leq D} I(X;\hat{X} \mid Y), \tag{5.8}$$

In general, there is a rate loss with WZ coding that $R_{WZ}(D) \geq R_{X|Y}(D)$. However, if the sources are jointly Gaussian and mean square difference is taken as the distortion measure (quadratic Gaussian case), there is no rate loss as in the lossless (SW) case, that is $R_{WZ}(D) = R_{X|Y}(D)$.

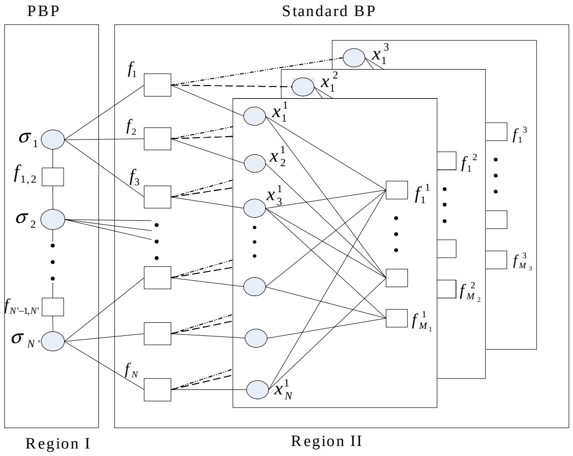

Figure 5.6: Factor graph of the joint WZ decoder.

5.3.2 Joint Source-channel Decoder that Combines SW Decoding and Dequantizing in a Single Step using BP

Let X and Y be two correlated continuous sources and the correlation between them can be modeled as a virtual channel, that is $X = Y + Z$, where $Z \sim N(0, \sigma_z^2)$ is an i.i.d. Gaussian random variable independent of Y. x_i and y_i, $i = 1, \ldots, N$ are sampled sequences of X and Y. For WZ coding, x_i is first quantized into $Q[x_i]$ using 2^q level quantization, and y_i is given at the decoder. Denote $x_i^1, x_i^2, \ldots, x_i^q$ as the binary format of the index $Q[x_i]$, and denote $\mathbf{B_j} = x_1^j, x_2^j, \ldots, x_N^j$ is the $j - th$ significant bit-plane. In the following description, the superscript j, $j = 1, \ldots, q$ represents the $j - th$ quantized bit or the $j - th$ bit-plane. In this work, each bit-plane is encoded independently by using LDPC codes, which are employed to implement SW coding for compressing $\mathbf{B_j}$, $j = 1, \ldots, q$, and compute the syndrome bits of each block.

The main idea of the proposed joint source-channel decoding scheme is illustrated in Region II, the subgraph of Fig. 5.6, where all circle nodes denote variable nodes and all square nodes denote factor nodes. The proposed approach is based on the syndrome based approach using LDPC codes. At the encoder, for the $j-th$ bit-plane, the input bits $\mathbf{B_j}$ are compressed into M_j syndrome bits, $\mathbf{S_j} = s_1^j, s_2^j \ldots, s_{M_j}^j$, thus resulting in $N : M_j$ compression. The factor nodes $f_1^j, f_2^j \ldots, f_{M_j}^j$ as shown in the Region II of Fig. 5.6, take into account the constraint imposed by the received syndrome bits. For the factor node $f_a^j, a = 1, \ldots, M_j,\ j = 1, \ldots, q$, the corresponding factor function is defined as

$$f_a^j(\mathbf{\tilde{x}_a^j}, s_a^j) = \begin{cases} 1, & \text{if } s_a^j \oplus \bigoplus \mathbf{\tilde{x}_a^j} = 0, \\ 0, & \quad\text{otherwise.} \end{cases} \tag{5.9}$$

where $\mathbf{\tilde{x}_a^j}$ denotes the set of neighbors of factor node f_a^j, and $\bigoplus \mathbf{\tilde{x}_a^j}$ denotes the binary sum of all elements of the set $\mathbf{\tilde{x}_a^j}$.

On the other hand, the correlation between a candidate quantized source $Q[x_i] = I\{x_i^1, x_i^2 \ldots, x_i^q\}$ and side information y_i is handled by a factor function $f_i,\ i = 1, \ldots, N$

$$\begin{aligned} f_i(Q[x_i], y_i, \sigma_z) = & \int_{P(Q[x_i])}^{P(Q[x_i]+1)} \frac{1}{\sqrt{2\pi\sigma_z^2}} e^{-\left(\frac{x-y_i}{\sqrt{2}\sigma_z}\right)^2} dx \\ & = \frac{1}{2}\text{erfc}(-\frac{P(Q[x_i]+1)-y_i}{\sqrt{2}\sigma_z}) \\ & -\frac{1}{2}\text{erfc}(-\frac{P(Q[x_i])-y_i}{\sqrt{2}\sigma_z}) \end{aligned} \tag{5.10}$$

where $I\{x_i^1, x_i^2 \ldots, x_i^q\}$, a binary to decimal conversion of $x_i^1, x_i^2 \ldots, x_i^q$, is equivalent to the quantization index of x_i, and $P(\bullet)$ denotes the lower boundary of quantization partition at index "$\bullet$", e.g. if a sampled source x_i satisfies $P(\bullet) \leq x_i < P(\bullet + 1)$, the quantization index $Q[x_i]$ of source x_i is equal to "$\bullet$".

According to the factor functions defined above, one can decode the values of $Q[\hat{x}_i] = I\{\hat{x}_i^1, \hat{x}_i^2 \ldots, \hat{x}_i^q\}$ using the BP algorithm, and finally estimate the source x_i

according to

$$\hat{x_i} = \int_{P(Q[\hat{x_i}])}^{P(Q[\hat{x_i}]+1)} \hat{x} \frac{1}{\sqrt{2\pi\sigma_z^2}} e^{-\left(\frac{\hat{x}-y_i}{\sqrt{2}\sigma_z}\right)^2} d\hat{x} \tag{5.11}$$

5.3.3 Adaptive Joint-source WZ Coding

To compress the source X close to the WZ bound in the standard BP approach, the correlation variance σ_z^2 must be assumed to be constant and known a priori. In practice, the correlation among the sources may vary over time. In this section, I will explain how to extend the proposed decoding algorithm to perform online correlation estimation by incorporating PBP [33]. Moreover, The proposed framework is universal and can be applied to any parametric correlation model.

Namely, I assume that σ_z is unknown and varies slowly over time. To model this, I introduce extra variable nodes $\sigma_1, \sigma_2, \ldots, \sigma_{N'}$, which are shown as circles in Region I of Fig. 5.6. Each factor node f_i in Region II is connected to an additional variable node corresponding to $\sigma_{i'}, i' = 1, \ldots, N'$ in Region I. Moreover, the factor function $f_i(Q[x_i], y_i, \sigma_{i'})$ of f_i between Region I and Region II is the same as (5.10). Here, the connection ratio is also defined as the number of factor nodes f_i in Region II that each variable node $\sigma_{i'}$ has connected, e.g. the connection ratio is equal to 3 in Fig. 5.6.

Since I assume that the correlation variance σ_z^2 only varies slowly over time, it is expected that adjacent variable nodes $\sigma_{i'}$ will not differ much in value. This characteristic is captured by additional factor nodes $f_{i',i'+1}, i' = 1, \ldots, N' - 1$ in Region I, which is defined as

$$f_{i',i'+1}\left(\sigma_{i'}, \sigma_{i'+1}\right) = e^{-\frac{\left(\sigma_{i'+1} - \sigma_{i'}\right)^2}{\lambda}} \tag{5.12}$$

where λ is a hyper-prior which can be chosen rather arbitrarily or estimated as in Section 4.1.4.

With the factor functions defined in (5.9), (5.10), (5.12) and dequantization function (5.11), I can estimate the values of x_i using BP algorithm. However, as mentioned before, standard BP algorithm, as an approximate technique for computing marginal probabilities by exchanging the message between adjacent neighboring nodes, can only handle discrete variables. Therefore, by incorporating PBP algorithm, which is described in Section 4.1.2, I can solve the problem and estimate the continuous correlation variance based on the factor graph show in Fig. 5.6.

5.4 Results and Discussion

In the simulation, I experimented the proposed scheme with a Gaussian correlation model. I generated a discrete, i.i.d. random Gaussian source $Y \sim \mathbf{N}(0, 1)$. Source $X = Y + Z$ was a noisy version of Y, where the correlation $Z \sim \mathbf{Z}(0, \sigma_z^2)$ was independent to Y. Moreover, I set the input source length N equal to $1,000$ and different quantization bits $q = 1, 2, \ldots, 6$ were used to quantize the input source. For LDPC coding, I used a regular parity check matrix with variable node degree equal to 2. 200 BP iterations and 10 particles for each variable node in Region I were used in the simulation. Moreover, each data point was averaged over 10,000 different codewords.

Fig. 5.7 plotted the rate-distortion performance of the proposed joint decoder when $\sigma_z^2 = 0.01$. I also gave the WZ rate-distortion function $R^*(D)(= R_{X|Y}(D))$ of X. I additionally included results based on Cheng's layered WZC scheme in [2] to benchmark the performance of the proposed joint decoder designs. Fig. 5.8 showed similar results when $\sigma_z^2 = 0.1$. According to the results in Fig. 5.7 and Fig. 5.8, the proposed joint "source-channel" decoder that combined SW decoding

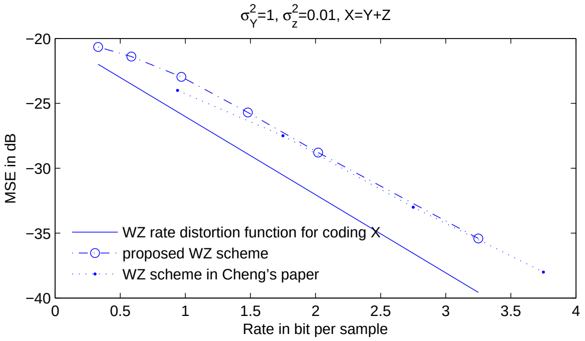

Figure 5.7: Rate-distortion performance of proposed WZ schemes and Cheng's WZ schemes [2], when $\sigma_z^2 = 0.01$. The WZ rate-distortion function $R^*(D)$ is also plotted for comparisons.

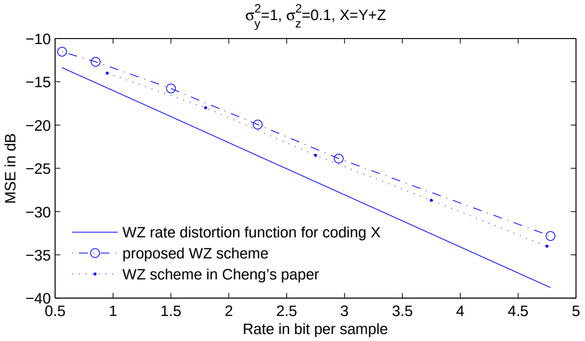

Figure 5.8: Rate-distortion performance of proposed WZ schemes and Cheng's WZ schemes [2], when $\sigma_z^2 = 0.1$. The WZ rate-distortion function $R^*(D)$ is also plotted for comparisons.

and dequantizing in a single step showed similar performance to Cheng's layered WZC scheme in [2], in which different bit-planes were decoded separately. I should note that the code length was 10^5 and well designed irregular profiles for different bit-planes are used in [2]. However, to reach a similar performance in the simulation, the code length was only $1,000$ and only a regular LDPC code design with a variable node degree 2 was used, which meant that the proposed joint decoder was more suitable for practical implementation.

Furthermore, I proceeded to study the decoding performance of the proposed joint decoder with a particle-based BP estimator. Here, I set $\lambda = 0.01$, $\sigma_r^2 = 0.005$, and the connection ratio equal to 16. For different decoding schemes, I set $\hat{\sigma} = 0.1$

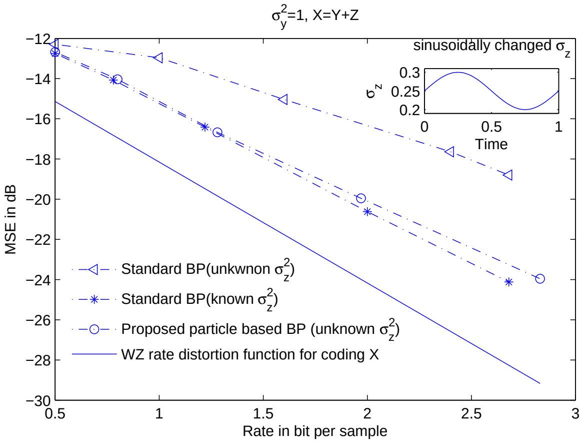

Figure 5.9: Rate-distortion performance of proposed WZ schemes with particle-based BP estimator of the correlation, when σ_z was changed sinusoidally. The WZ rate-distortion function $R^*(D)$ and proposed WZ schemes under standard BP with known and unknown σ_z are also plotted for comparisons.

as the prior knowledge of σ_z. In this simulation, I assumed that σ_z was changed sinusoidally from 0.2 to 0.3 between each pair of sources x_i and y_i, which was shown in the sub-figure of Fig. 5.9. Fig. 5.9 showed that the joint decoder using a particle-based BP estimator obtained better performance than the joint decoder using a standard BP without known σ_z, since the proposed particle-based BP estimator offered an online estimation of the source correlation. Moreover, we can see the performance of the joint decoder with a correlation estimator was very close to the performance of the decoder with a perfect knowledge σ_z, which meant the proposed estimator offered a accurate estimation of the true source correlation.

Finally, in Fig. 5.10, I presented the estimation accuracy of the proposed particle-based BP estimator. I set the initial value $\hat{\sigma} = 0.1$, while the true value was sinusoidally changed from 0.2 to 0.3. We can see the adaptive decoding algorithm can quite faithfully estimate the true value of σ_z.

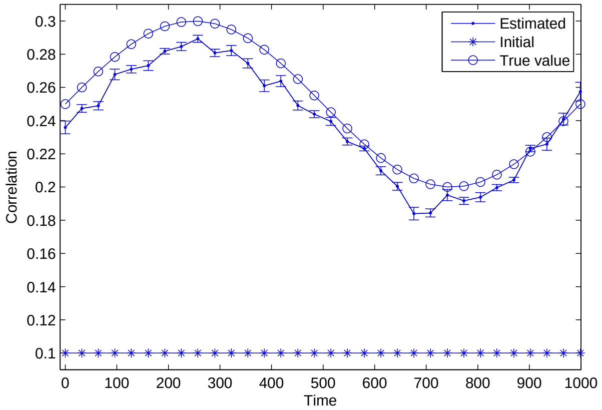

Figure 5.10: Estimation of the sinusoidal correlation, where the initial estimation of σ_z is 0.1.

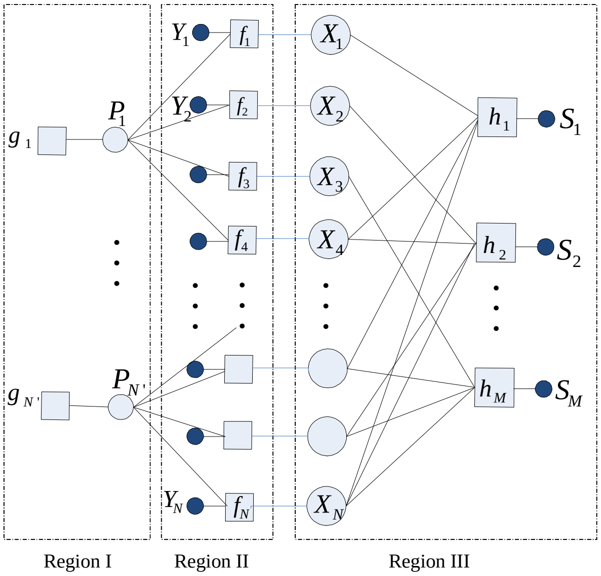

Figure 5.11: Factor graph of adaptive SW decoding based on EP.

5.5 SW Decoding with Time-varying Correlation Estimation using EP

In this section, I talk about the adaptive SW decoding with correlation estimation based on EP algorithm (see the factor graph in Fig. 5.11). Here, I also let X and Y be two correlated binary sources (taking values 0 and 1) and the correlation between them be symmetric in such a way that Y can be considered as the output of X passing through a BSC with unknown crossover probability ρ_t, That is,

$$Y = \begin{cases} X & \text{with probability } 1-\rho_t \\ X \oplus 1 & \text{with probability } \rho_t, \end{cases} \tag{5.13}$$

I assume that the crossover probability ρ_t may drift over time.

Note that if ρ_t is known *a priori*, X can be compressed very close to the SW limit $H(X|Y)$ using syndrome based approach and LDPC codes [3]. At the SW encoder, the syndrome $\mathbf{s}$ of one source block $\mathbf{x}$ is computed and transmitted to the decoder. At the SW decoder, the other source block $\mathbf{y}$ is treated as the output of $\mathbf{x}$ passing through a correlation channel. SW decoding is almost identical to conventional LDPC decoding. However, rather than decoding to a codeword, the decoder approximates the estimated source block $\mathbf{x}$ as a code vector with the received syndrome. Thus, just as channel decoding, the source block $\mathbf{x}$ can be reconstructed using BP algorithm over the corresponding factor as shown in Region II and III of Fig. 5.11. Since we need to take into the constraint imposed by the received syndrome, the factor function in Region III is redefined as follows

$$h_k(\mathbf{x}_k) = \begin{cases} 1 & \text{if } s_k \oplus \bigoplus_{i \in \mathcal{N}(h_k)} x_i = 0, \\ 0 & \qquad\text{otherwise,} \end{cases} \tag{5.14}$$

where s_k is the k-th bit of the received syndrome $\mathbf{s}$.

In addition, since I consider binary SW coding, where correlation is modeled as a virtual BSC, the likelihood factor function in Region II need to be modified as follows

$$f_i(\rho_j, x_i, y_i) = \rho_j^{x_i \oplus y_i}(1-\rho_j)^{1 \oplus x_i \oplus y_i}. \tag{5.15}$$

To perform correlation estimation, I also introduce additional factor nodes to model the time-varying crossover probability in Region I of Fig. 5.11. In this case, I also need to choose a conjugate prior of likelihood function to make the approximation tractable. Since beta distribution is the conjugate prior of likelihood function in (5.15), the corresponding factor function in Region I should be changed into beta distribution, which is defined as $\text{Beta}(\rho, \alpha, \beta) = \frac{1}{\text{beta}(\alpha,\beta)}\rho^{\alpha-1}(1-\rho)^{\beta-1}$ with shape parameters α and β. Then, the factor function in Region I has the following form

$$g(\rho_j, \alpha_j, \beta_j) = \text{Beta}(\rho_j, \alpha_j, \beta_j). \tag{5.16}$$

Finally, with these new defined factor functions, SW decoding with time-varying correlation estimation using EP can be implemented according to the similar workflow of LDPC decoding with time-varying AWGN channel estimation using EP in Chapter 4.

5.6 Results and Discussion

In this section, the decoding performances (in terms of bits error rate (BER)) of standard BP and EP based BP decoders are presented in the presence of a crossover probability mismatch. Here, I consider two different scenarios, constant crossover probability mismatch and time-varying crossover probability mismatch of the BSC modeling the correlation between two binary sources.

I first study the performance of the proposed EP estimator for different values of constant crossover probability in Fig. 5.12. In this case, SW coding based on an irregular LDPC code with the degree profile in [50], code length $N = 1000$ and code rate $R = 0.5$ is used in the simulation. Since I consider the estimation of constant crossover probability, there is only one variable node in Region I, which means that the connection ratio C is set to 1000. All the results are obtained over

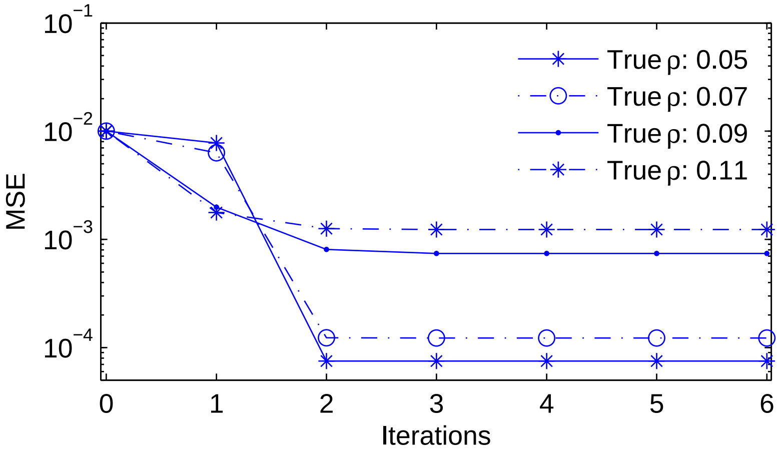

Figure 5.12: The performance of the EP estimator for different values of constant crossover probability.

100 trials. Initial crossover probabilities used for BP decoding are always 0.1 above true crossover probabilities. Moreover, the maximum number of iterations for BP decoding is $T_{\text{BP}} = 50$. Since I only focus on the study of estimation accuracy in this case, the EP estimator is only used once at the end of the BP decoding. To achieve the best decoding performance, new estimates can be sent back to LDPC based SW decoder periodically, and this setup will be studied later in this section. In Fig. 5.12, we can see that the EP estimator can always converge within 3 or 4 iterations. The estimation accuracy of the proposed EP estimator, in terms of MSE, increases as the true crossover probability decreases. It is because that for a given LDPC based SW decoder, a larger true crossover probability may yield a larger number of BP decoding errors, which degrade the performance of the EP estimator.

Second, I study the performance of the proposed EP estimator for time-varying crossover probability in Fig. 5.13. In this section, since I consider a BSC modeled correlation, the time-varying crossover probability is modeled as $\rho_t = \rho_0 + N(0, \sigma^2_{\text{noise}})$, where the code rate $R = 0.5$ is used in the simulation, ρ_0 is a constant,

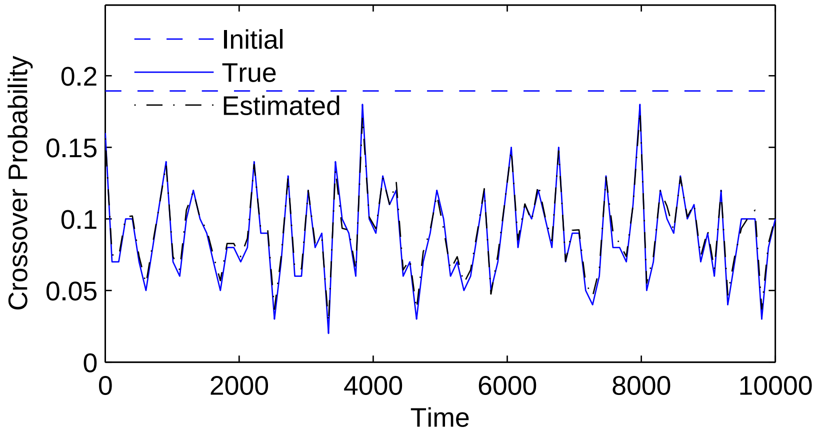

Figure 5.13: Estimation of time-varying crossover probability using the proposed EP estimator, where the initial crossover probability for BP decoding is 0.1 above the mean of the time-varying crossover probability.

and $N(0, \sigma^2_{\text{noise}})$, a Gaussian noise with parameter σ_{noise}, models the fluctuation on ρ_0. Fig. 5.13 shows the estimate of a time-varying crossover probability, where $\rho_0 = 0.09$, $\sigma_{\text{noise}} = 0.01$, $N = 10,000$ and $C = 50$. Note that, other settings in this case are the same as the study in Fig. 5.12. It can be seen that the proposed EP estimator provides a precise estimate of the time-varying crossover probability, even though the initial crossover probability is far away from the mean of the time-varying crossover probability. Compared to [33], particle estimator usually needs a large number of iterations (> 200) to converge, however, the proposed EP estimator usually converges within 3 iterations. Moreover, the computation complexities of each iteration for both EP estimator and BP estimator are quite similar. Thus, the proposed EP estimator can approximate the correlation much more rapidly.

Third, in Fig. 5.14, I study the decoding performances of LDPC based SW decoder with and without EP estimator. In this case, I assume that the crossover probability is constant for each simulation. The following results are obtained over 10,000 trials. The codeword lengths of 10^3 and 10^4 are studied, where the initial crossover probabilities are 0.1 above the true crossover probabilities for both BP

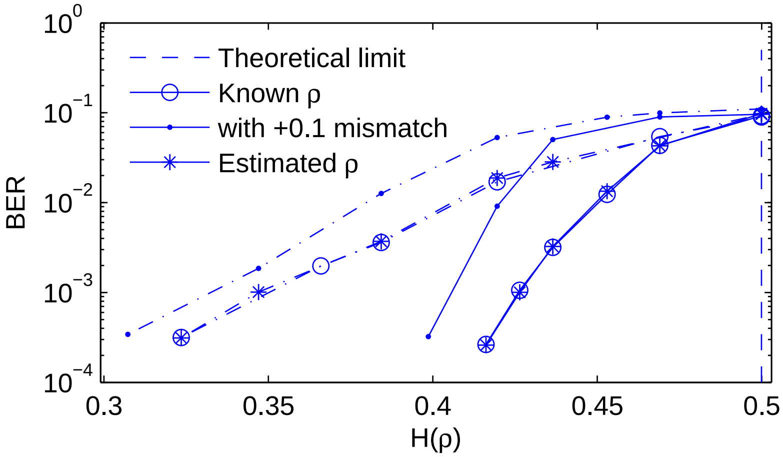

Figure 5.14: Performance of BER versus $H(\rho)$ for LDPC decoder using 1) knowledge of the true crossover probability, 2) knowledge of the initial crossover probability with +0.1 mismatch and 3) the crossover probability determined by EP estimator, where the crossover probability for each simulation is a constant. The codeword lengths of 10^3 (dashed-dot) and 10^4 (solid) were studied.

decoder and EP based BP decoders. In Fig. 5.14, we can see that a big performance gap between BP decoder and EP based BP decoder is observed in the presence of crossover probability mismatch. Moreover, the performance curve of BP decoder with knowledge of the true crossover probability is provided as the benchmark. Simulation results of different codeword block lengths showed no obvious degradation of performance between the proposed EP based BP decoder and BP decoder with known crossover probability.

Finally, I investigate the decoding performances of BP decoder and EP based BP decoder with time-varying crossover probability, where $\sigma_{\text{noise}} = 0.03$. By changing ρ_0, different time-varying crossover probabilities are sampled from the aforementioned model in the study of Fig. 5.13. The mean values of different time-varying crossover probabilities used in this study change from 0.079 to 0.11. The initial values in BP decoder and EP based BP decoder are 0.1 above these mean values. In this case, the EP estimator starts working after 50 BP iterations. Moreover, I

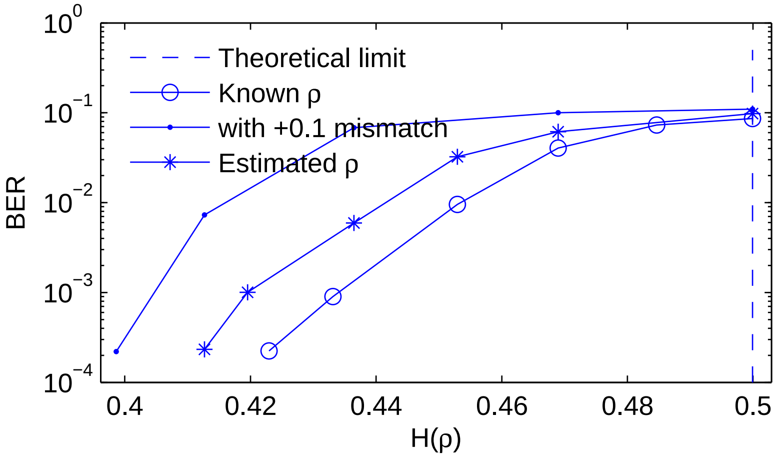

Figure 5.15: Performance of BER versus $H(\rho)$ for LDPC decoder using 1) knowledge of the true crossover probability, 2) knowledge of the initial crossover probability 0.1 above the mean of the true crossover probability and 3) the crossover probability determined by EP estimator, where the time-varying crossover probability for each simulation varies according to the aforementioned model in the study of Fig. 5.13 by using different ρ_0. The codeword length of 10^4 is studied.

perform EP estimator periodically for every 20 BP iterations until BP decoder successfully decodes the codeword or reaches its maximum number of iterations (i.e. 150 in the experiment). Fig. 5.15 shows that the proposed EP based BP decoder obtained a much better performance than BP decoder with the known 0.1 mismatch of the mean of time-varying crossover probability. The gap between BP with and without the knowledge of true crossover probability is about 0.024 bits, however, the gap between a known true SNR BP decoder and the EP based BP decoder is less than 0.01 bits.

5.7 Summary

In this chapter, I first proposed an adaptive decoding scheme for SW coding using PBP. The scheme has been tested on both asymmetric SW coding and non-asymmetric SW coding A precise estimation of correlation between the two sources using the adaptive decoding algorithm has been observed from the experiments.

Moreover, I have observed a significant gain of my algorithm over the standard BP algorithm even when there is a slight fluctuation of the correlation among sources.

Second, to handle the correlation estimation problem of WZ coding, I design a joint bit-plane model, by which the PBP algorithm can be applied to tracking the correlation between non-binary sources. Through experimental results, the proposed correlation estimation approaches significantly improve the compression performance of DSC.

Finally, I designed an adaptive SW decoding scheme using EP algorithm. By comparing the proposed EP based approach with PBP algorithm, the results show that the proposed EP estimator obtains the comparable estimation accuracy with less computational complexity than the PBP method.

CHAPTER 6

APPLICATIONS ON ONBOARD SOLAR IMAGES

Acquiring and processing astronomical images is becoming increasingly important for accurate space weather prediction and expanding our understanding about the Sun and the Universe. These images are often rich in content, large in size and dynamic range. Efficient, low-complexity compression solutions are essential to reduce onboard storage, processing, and communication resources. Distributed compression is a promising technique for onboard coding of solar images by exploiting correlation between successively acquired images. In this section, the idea of the proposed algorithm in Section 5.3 is tested on the stereo solar images captured by the twin satellites system of NASA's STEREO project, where an adaptive distributed compression solution using PBP is used to track correlation, as well as perform disparity estimation, at the decoder side.

6.1 Background

Onboard data processing has always been a challenging and critical task in remote-sensing applications due to severe computational and/or power limitations of onboard equipment. This is especially the case in deep-space applications, where mission spacecrafts are collecting a vast amount of image data that is stored and/or communicated to the observation center. In such emerging applications, efficient low-complexity image compression is a must. While conventional solutions, such as JPEG, have been successfully used in many prior missions, demand for increasing image volume and resolution as well as increased space resolution and wide-swath imaging calls for a larger coding efficiency at reduced encoding complexity.

NASA's STEREO (Solar TErrestrial RElations Observatory), launched in Oct. 2006, has very recently and still is providing ground-breaking images of the Sun using two space-based observatories [59]. These images aim to reveal the processes in the solar surface (photosphere), through the transition region into the corona and provide the 3D structure of coronal mass ejections (CME). CMEs are violent eruptions solar plasma into space, which, if directed towards the Earth and reaches it as an interplanetary CME along with solar flares of other origins, are known to have catastrophic effects on the radio transmissions, satellites, power grids resulting in large-scale and long-lasting power outages, and on humans travelling in airplanes at high altitude.

The data streams that are transmitted 24 hours per day as weather beacon telemetry from each spacecraft have to be heavily compressed [59]. The reconstructed images are available online, immediately after reception. Due to compression, many image artifacts have been spotted that led to wrong conclusions (see e.g., [60]). Another, scientific stream is recorded and transmitted daily using NASA Deep Space Network lightly compressed. These images are becoming available 2-3 days after arrival in the Flexible Image Transport System (FITS) and/or JPEG format.

A variety of image compression tools are currently used in deep-space missions, ranging from Rice and lossy wavelet-based compression tools (used in PICARD mission by CNES2009), Discrete Cosine Transform (DCT) + scalar quantization + Huffman coding (Clementine, NASA1994), ICER (a low-complexity wavelet-based progressive compression algorithm used in Mars mission, NASA2003) to (12-bit) JPEG-baseline (Trace NASA1998, Solar-B JAXA2006) [61]. The compression algorithms have mainly been implemented in hardware (ASIC or FPGA implementation), but some of them run as software on DSP processors (e.g., ICER). The key characteristics of these algorithms are relatively-low encoding power consumption, coding efficiency, and error resilience features. Latest Earth observation satellites usually employ JPEG2000 [62] or similar wavelet-based bitplane coding methods implemented on FPGA, which might be too prohibitive for deep-space missions. Note that all current missions, including STEREO, use 2D, mono-view image compression trading off computational cost and compression performance. Since STEREO images are essentially multi-view images, with high inter-view correlation, current compression tools do not provide an optimum approach. Thus, in this chapter, I propose a distributed multi-view image compression (DMIC) scheme for such emerging remote-sensing set-ups.

When an encoder can access images from multiple views, a joint coding scheme [63] achieves higher compression performance than schemes with separate coding, since multi-view images are usually highly correlated. However, due to the limited computing and communication power of space imaging systems, it is not feasible to perform high-complexity, power-hungry onboard joint encoding of captured solar images. Although, intuitively, this restriction of separate encoding seems to compromise the compression performance of the system, DSC theory [9,10] proves that distributed independent encoding can be designed as efficiently as joint encoding as long as joint decoding is allowed, propelling DSC as an attractive low-complexity onboard source coding alternative.

The proposed DMIC image codec is characterised by low-complexity image encoding, and relatively more complex decoding meant to be performed on the ground. A novel joint bit-plane decoder is described, that integrates PF with standard BP decoding to perform inference on a single joint 2-D factor graph. I test the lossy DMIC setup with grayscale stereo solar images obtained from NASA's STEREO mission [59] to demonstrate high compression efficiency with low encoding complexity and non power-hungry onboard encoding, brought about by DSC. DSC has been used for onboard compression of multispectral and hyperspectral images in [64], [65], where DSC is used to exploit efficiently inter-band correlation. In [65], for example, a low-complexity solution robust to errors is proposed using scalar coset codes to encode the current band and the previous bands as decoder side information. The algorithms of [65] are implemented using FPGA, and simulations on AVIRIS images show promising results.

The key contributions of this chapter can be summarized as:

- An adaptive distributed multi-view image decoding scheme, which can estimate the blockwise correlation and disparity change between two correlated images, and also recover the images simultaneously;
- A PBP decoder to estimate blockwise correlation changes, since standard BP cannot handle continuous variables (except linear Gaussian model) such as the correlation parameter. This extends my previous work [34], [66] from 1-D correlation estimation to 2-D and from time varying correlation estimation to spatial varying correlation estimation;
- A joint bit-plane decoder (compared to the traditional separate bit-plane decoder [2]), that allows the estimation of the correlation and the disparity between two pixels directly rather than just the correlation between a corresponding pair of bits of the pixels;
- A decoding scheme that offers greater feasibility for rate selection than the joint bit-plane encoder/decoder design used in [67] since the received syndromes of each bit-plane are independent due to separate bit-plane encoding.

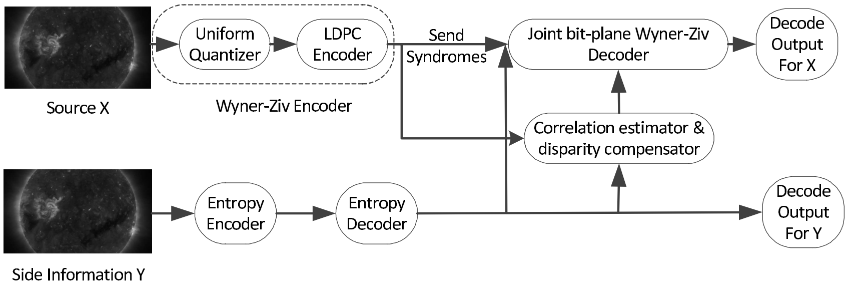

Figure 6.1: Lossy DMIC setup with disparity and correlation estimation.

6.2 Related Work

Since the proposed DMIC scheme intersects several research topics, I group prior work into three categories.

The first category relates to work in the area of low-complexity onboard/remote multi-view image coding. This area is still in its infancy, and I found only two relevant contributions. In [68], lossy compression of Earth orbital stereo imagery used for height detection with three or four views based on motion compensation and JPEG-2000 [62] and JPEG-LS is proposed. Note that motion compensation+ JPEG-2000 might still be considered as power expensive for remote sensing, including deep-space missions. In [69], a modification of the mono-view ICER image coder, employed in the Mars Exploration mission, is proposed. The proposed coder optimizes a novel distortion metric that reflects better stereoscopic effects rather than conventional mean-square error (MSE) distortion. The results reported in [69] show improved stereo ranging quality despite the fact that correlation information between the left and right image pair was not exploited in any way or form. See also [70].

The second relevant topic is correlation tracking in DSC applications. Most DSC designs, including Distributed Video Coding (DVC), so far (with few exceptions) usually simplify the problem by modeling correlation noise, i.e., the difference between the source and side information, as Laplacian random variables and estimate the distribution parameters either based on training sequences or previously decoded data. This imposes certain loss especially for images or sequences that are very different or non-stationary. Non-stationarity of the scene has been dealt mainly by estimating correlation noise (e.g., on the pixel or block level) from previously decoded data and different initial reliability is assigned to different pixels based on the amount of noise estimated both in pixel- and transform-domains [71–75]. In [66], we proposed an efficient way of estimating correlation between the source and side information for pixel-domain DVC by tightly incorporating the process within the SW decoder via SW code factor graph augmentation to include correlation variable nodes with particles such that particle filtering is performed jointly with BP over the augmented factor graph during the SW decoding process. Note that the BP-based SW decoding and correlation statistics estimation are considered jointly. The proposed correlation estimation design was tested on a transform-domain DVC [76] with a feedback channel, but with joint bit-plane coding. This work extends this result from mono-view to low-encoding complexity multi-view coding.

The third relevant topic is multi-view image coding using DSC principles. Despite the potential of DSC, attaining its ultimate performance relies on the assumption that both the correlation and the disparity among multi-view images are known a *priori* at the decoder. Direct measurement of the correlation and the disparity at the encoder side is both expensive in terms of computation and impossible without communication among imaging sensors. Thus, estimating correlation and disparity at the decoder becomes the main challenge in DMIC. For disparity estimation in DMIC, the idea of motion compensation [77], [78] used in DVC offers a possible solution. However, these motion compensation methods usually require an excessive amount of computation. Thus, some low complexity disparity learning schemes for DMIC have been proposed in the literature. In [67], Varodayan *et. al.* developed an Expectation Maximization (EM) based algorithm at the decoder to learn block-based disparity for lossless compression [67] and then extended it to lossy case [79]. In comparison with the system without disparity compensation, a better compression performance is observed when disparity compensation is employed at the decoder [67, 79].

Thus, knowing the correlation among multi-view images is a key factor in determining the performance of a DMIC scheme. This correlation is generally nonstationary (spatially varying) and should be handled adaptively. For example, in [80] an edge-based correlation assignment method is proposed, where the correlation parameters of blocks with and without edges are assigned to different values. However, even the aforementioned work is based on the assumption that the correlation among images is known a *priori*. Similarly to disparity compensation, dynamic correlation estimation given at the decoder could also yield significant improvement in performance. However, most studies of correlation estimation in DSC focus on the correlation estimation of stationary binary sources [81], [82], which are not suitable for the non-binary image sources in the DMIC case.

Several other approaches were proposed in literature [83–86], neither of which uses correlation tracking. A review on multiview-video coding based on DSC principles can be found in [87, 88].

6.3 Distributed Multi-view Image Coding

Let X and Y be a pair of correlated multi-view images with size M by N. Assuming that a horizontal disparity shift D exists between pixels of X and pixels of Y, the relationship between X and Y can be modeled as

$$X_{(x,y)} = Y_{\left(x-D_{(x,y)},y\right)} + Z_{(x,y)}, \tag{6.1}$$

where $x = 1, 2, \ldots, M$ and $y = 1, 2, \ldots, N$ denote the coordinates of pixels, and $Z_{(x,y)}$ satisfies a Laplace distribution $\mathcal{L}(Z_{(x,y)}|\sigma) = \frac{1}{2\sigma}\text{exp}\left(-\frac{|Z_{(x,y)}|}{\sigma}\right)$.

I consider a lossy DMIC setup as shown in Fig. 6.1, where image Y, the side information, is perfectly known at the decoder through a conventional image coding. At the encoder side, image X is first quantized into $Q[X_{(x,y)}]$ using 2^q level uniform nested scalar quantization (NSQ) [89] and then is encoded based on LDPC codes [2], where each bit-plane is independently encoded into syndrome bits of an LDPC code. Denote $X^1_{(x,y)}, X^2_{(x,y)}, \ldots, X^q_{(x,y)}$ as the binary format of the index $Q[X_{(x,y)}]$, and denote $\mathbf{B^j} = X^j_{(1,1)}, X^j_{(1,2)}, \ldots, X^j_{(M,N)}$ as the j^{th} significant bit-plane, where the superscript $j = 1, \ldots, q$ is used to represent the j^{th} quantized bit or the j^{th} bit-plane in the rest of this chapter. At the LDPC decoder, the BP algorithm is employed to decode image X using the received syndrome bits, the given correlation, and the side information Y reordered by the given disparity information. Finally, when the BP algorithm converges, image X can be recovered based on the output belief for each pixel [2].

6.4 Adaptive Joint Bit-plane WZ Decoding of Multi-view Images with Disparity Estimation

6.4.1 Joint Bit-plane WZ Decoding

In popular layer WZ approaches such as [2], each bit plane of the quantized source is recovered *sequentially* and this makes it difficult and inefficient for the decoder to perform the disparity and correlation estimation. In order to facilitate the disparity

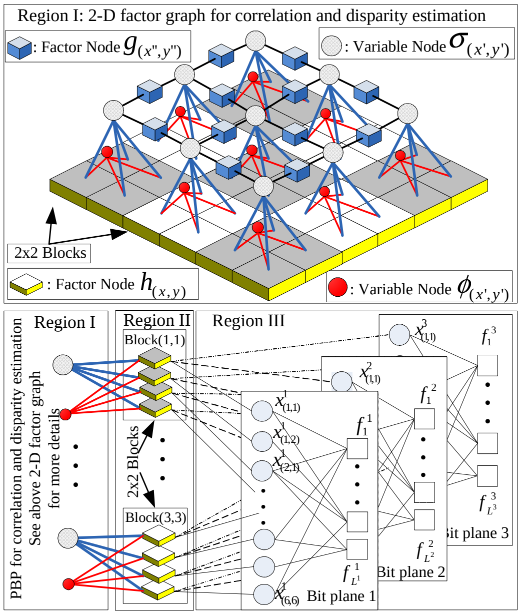
Region I: 2-D factor graph for correlation and disparity estimation
: Factor Node $g_{(x'',y'')}$
: Variable Node $\sigma_{(x',y')}$
2x2 Blocks
: Factor Node $h_{(x,y)}$
: Variable Node $\phi_{(x',y')}$
Region I
PBP for correlation and disparity estimation
See above 2-D factor graph
for more details
Region II
Block(1,1)
2x2 Blocks
Block(3,3)
Region III
$x^3_{(1,1)}$
f^3_1
$x^2_{(1,1)}$
f^2_1
$x^1_{(1,1)}$
$x^1_{(1,2)}$
$x^1_{(2,1)}$
f^1_1
$f^3_{L^3}$
Bit plane 3
$f^2_{L^2}$
Bit plane 2
$f^1_{L^1}$
$x^1_{(6,6)}$
Bit plane 1

Figure 6.2: Factor graph of joint bit-plane decoder with disparity and correlation estimation.

and the correlation estimation, I introduce a joint bit-plane WZ decoding scheme, which can adaptively exploit the disparity and the correlation between a non-binary source and side information during the decoding process. The main idea of the proposed joint bit-plane WZ decoding scheme is illustrated in Regions II and III of Fig. 6.2, where all circle nodes denote variable nodes and all square nodes denote factor nodes. The encoder used in this chapter is the traditional syndrome-based approach using LDPC codes [2]. At the encoder side, a given bit plane $\mathbf{B^j}$ is compressed into L^j number of syndrome bits, $\mathbf{S^j} = s^j_1, s^j_2 \ldots, s^j_{L^j}$, thus resulting in $(M \times N) : L^j$ compression.

At the joint bit-plane decoder, the factor nodes $f^j_1, f^j_2 \ldots, f^j_{L^j}$ as shown in the Region III of Fig. 6.2, take into account the constraint imposed by the received syndrome bits. For a factor node $f^j_a, a = 1, \ldots, L^j$, $j = 1, \ldots, q$, the corresponding factor function is defined as

$$f_a^j(\tilde{\mathbf{X}}_a^j, s_a^j) = \begin{cases} 1, & \text{if } s_a^j \oplus \bigoplus \tilde{\mathbf{X}}_a^j = 0, \\ 0, & \quad \text{otherwise.} \end{cases} \tag{6.2}$$

where $\tilde{\mathbf{X}}_a^j$ denotes the set of neighbors of the factor node f_a^j, $\oplus$ represents the bitwise addition and $\bigoplus \tilde{\mathbf{X}}_a^j$ denotes the bitwise sum of all elements of the set $\tilde{\mathbf{X}}_a^j$.

Then in Region II, the relationship among a candidate quantized source $Q[X_{(x,y)}]$, side information $Y_{(x,y)}$ and disparity compensation $D_{(x,y)}$ can be modeled by the factor function

$$\begin{aligned} &h_{(x,y)}(Q[X_{(x,y)}], Y_{(x,y)}, \sigma, D_{(x,y)}) = \\ &\qquad \int_{P\left(Q[X_{(x,y)}]\right)}^{P\left(Q[X_{(x,y)}]+1\right)} \frac{1}{2\sigma} \exp\left(\frac{|\mathcal{X} - Y_{(x-D_{(x,y)},y)}|}{\sigma} \right) d\mathcal{X}, \end{aligned} \tag{6.3}$$

where $P(\mathcal{Q})$ denotes the lower boundary of quantization partition at index "$\mathcal{Q}$", e.g., if a pixel $X_{(x,y)}$ satisfies $P(\mathcal{Q}) \le X_{(x,y)} < P(\mathcal{Q}+1)$, the quantization index $Q[X_{(x,y)}]$ of pixel $X_{(x,y)}$ is equal to "$\mathcal{Q}$". Then given the estimation of the correlation σ and the disparity $D_{(x,y)}$, standard BP can be used to perform joint bit-plane decoding based on the proposed factor graph (see Regions II and III in Fig. 6.2) and the corresponding factor functions (6.2) and (6.3).

6.4.2 Joint Bit-plane WZ Decoding with Disparity Estimation

I assume that each block includes $n \times n$ pixels and shares the same disparity, which yields $\lceil \frac{M}{n} \rceil \times \lceil \frac{N}{n} \rceil$ number of blocks for an $M \times N$ image, where $\lceil \bullet \rceil$ represents the ceiling of "$\bullet$" that rounds "$\bullet$" toward positive infinity. Then the horizontal disparity field is a constant within a block and will be denoted as $D_{(x',y')} \in \{-l, \cdots, 0, \cdots, l\}$, where $x' = 1, \cdots, \lceil \frac{M}{n} \rceil$ and $y' = 1, \cdots, \lceil \frac{N}{n} \rceil$ are the block indices. Thus, in the rest

$$m_{h_{(x,y)}\to\phi_{(x',y')}}\left(D_{(x',y')}\right) \propto \sum_{Q[X_{(x,y)}]\in[0,2^q]} h_{(x,y)}(Q[X_{(x,y)}], Y_{(x,y)}, \sigma, D_{(x',y')}) \prod_{j=1}^{q} m_{X^j_{(x,y)}\to\phi_{(x',y')}}\left(X^j_{(x,y)}\right), \tag{6.4}$$

$$m_{h_{(x,y)}\to X^j_{(x,y)}}\left(X^j_{(x,y)}\right) \propto \sum_{D_{(x',y')}} h_{(x,y)}(Q[X_{(x,y)}], Y_{(x,y)}, \sigma, D_{(x',y')}) m_{\phi_{(x',y')}\to h_{(x,y)}}\left(D_{(x',y')}\right) \prod_{j'\in[1,q]/j} m_{X^{j'}_{(x,y)}\to h_{(x,y)}}\left(X^{j'}_{(x,y)}\right), \tag{6.5}$$

of this chaper, I will use $D_{(x',y')}$ to represent the disparity $D_{(x,y)}$ of a pixel $X_{(x,y)}$ that lies inside the Block(x', y'). For example, in the 2-D factor graph of Fig. 6.2, a 6×6 image is divided into 3×3 number of blocks with 2×2 pixels in each block.

In order to estimate the disparity between images, I introduce extra variable nodes $\phi_{(x',y')}$ in Region I (see the 2-D factor graph in Fig. 6.2). Each factor node $h_{(x,y)}$ in Region II is connected to an additional variable node $\phi_{(x',y')}$ in Region I. Here, I define the connection ratio as the number of factor nodes $h_{(x,y)}$ in Region II which each variable node $\phi_{(x',y')}$ is connected to, e.g., the connection ratio is equal to 4 in Fig. 6.2. According to the BP update rules, the factor node update from Region II to the variable node $\phi_{(x',y')}$ in Region I can be written as (6.4), where $\mathcal{N}(\phi_{(x',y')})/h_{(x,y)}$ denotes all the neighboring factor nodes of a variable node $\phi_{(x',y')}$ except $h_{(x,y)}$. Moreover, (6.4) can be interpreted as the E-step algorithm used in [67]. Similarly, the factor node update from Region II to Region III can be written as (6.5), where (6.5) can also be interpreted to the M-step algorithm used in [67].

6.4.3 Joint Bit-plane Wyner-Ziv Decoding with Correlation Estimation

To compress image X close to the WZ bound in the standard BP approach, the correlation parameter σ must be known a *priori*. However, in practice, the correlation between the colocated pixels of the pair of correlated images X and Y is unknown, and making the situation even more challenging, this correlation may vary over space. Thus, besides the proposed disparity estimation, I introduce an additional correlation estimation algorithm to perform online correlation tracking by extending my previous correlation estimation model [34] from 1-D to 2-D and from time-varying to spatial varying. Moreover, the proposed framework is universal and can be applied to any parametric correlation model.

Namely, I assume that σ is unknown and varies block-by-block over space, where the same blockwise assumption is also used in Section 6.4.2. To model this, I introduce another set of extra variable nodes $\sigma_{(x',y')}$ in Region I (see the 2-D factor graph in Fig. 6.2). Now, each factor node $h_{(x,y)}$ in Region II will be connected to an additional variable node $\sigma_{(x',y')}$ in Region I. Here the connection ratio used for correlation estimation is the same as that for disparity estimation in Section 6.4.2. Moreover, the correlation parameter σ used in the factor function $h_{x,y}(Q[X_{(x,y)}], Y_{(x,y)}, \sigma, D_{(x',y')})$can be modified accordingly by replacing σ as $\sigma_{(x',y')}$, since I assume σ varies over space.

Furthermore, the correlation changes among adjacent blocks may not be arbitrary [80]. The ability to capture correlation changes among adjacent blocks can significantly increase the stability of correlation tracking of each block. To achieve this, I introduce additional factor nodes $g_{(x'',y'')}$ in Region I (see the 2-D factor graph in Fig. 6.2), where $x'' = 1, \cdots, \lceil \frac{M}{n} \rceil - 1$ and $y'' = 1, \cdots, \lceil \frac{N}{n} \rceil - 1$ denote block indices just as x' and y'. The corresponding factor function can then be modeled as

$$g_{(x'',y'')}\left(\sigma_{(x'',y'')}, \sigma_{(x''+c,y''+d)}\right) = \exp\left(-\frac{\left(\sigma_{(x'',y'')} - \sigma_{(x''+c,y''+d)}\right)^2}{\lambda}\right), \tag{6.6}$$

where the offset $(c,\ d)$ is restricted to $\{(0,1),(0,-1),(1,0),(-1,0)\}$ according to the defined 2-D factor graph, and λ is a hyper-prior and can be chosen rather arbitrarily.

Since standard BP can only handle discrete variables with small alphabet sizes or continuous variables with linear Gaussian model, it cannot be applied directly for estimating the continuous correlation parameters. However, by incorporating PF with BP shown in Section 4.1.2, BP can be extended to handle continuous variables. Then the proposed factor graph model can be used to estimate the continuous correlation.

6.5 Results and Discussion

To verify the effect of correlation and disparity tracking for DMIC, I tested the above setup with grayscale stereo solar images [59] captured by two satellites of the NASA's STEREO project, where the twin satellites are about 30 million miles apart, and the viewing angle is about $6-8$ degrees. For the purposes of illustrating accurate tracking of the correlation and the disparity, the simulations for the SW code use only a low-complexity regular LDPC code with variable node degree 5. More complex irregular codes would further improve the overall peak signal-to-noise ratio (PSNR) performance.

The following constant parameters are used in the simulations: image height $M = 128$ pixels, image width $N = 72$, maximum horizontal shift $l = 5$, block size $n = 4$, hyper-prior $\lambda = 10$, initial correlation for Laplace distribution $\sigma = 5$ and initial distribution of disparity

$$p(D_{(x',y')}) = \begin{cases} 0.75, & \text{if } \ D_{(x',y')} = 0; \\ 0.025, & \text{otherwise}; \end{cases} ,$$

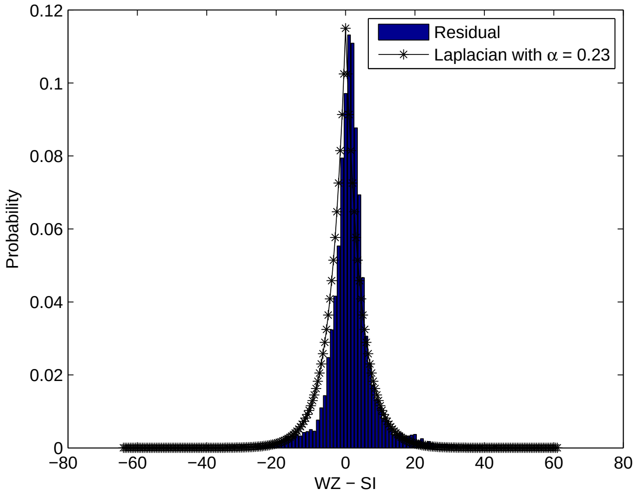

Figure 6.3: Residual histogram for solar images in SET 1.

where the selection of initial values follows [67]. Moreover, I have tested two sets of solar images in the simulation results, where I refer to as solar image SET 1 and solar image SET 2, respectively.

First, I verified the Laplacian assumption of the correlation between correlated images X and Y in Fig. 6.3. By setting $\alpha = 0.23$, Laplace distribution provides an accurate approximate to the residual between images X and Y.

Then, I examine the rate-distortion performance of the proposed adaptive DMIC scheme, where PSNR of the reconstructed image is calculated as an indicator of the distortion. I consider the following five different setups.

a). Adaptive correlation DMIC with a known disparity, which is used as the benchmark performance.

b). Adaptive correlation and disparity DMIC, which is the proposed scheme in this chapter.

c). Adaptive disparity DMIC with a known fixed correlation, which corresponds to the setup used in [67], [79].

d). Non-adaptive joint bit-plane DMIC with known fixed correlation only, where the correlation and disparity estimators are not available at the decoder.

e). Non-adaptive separate bit-plane DMIC with known fixed correlation only, which corresponds to the setup used in [2].

In Fig. 6.4 and Fig. 6.5 (corresponding to solar images in SET 1 and SET 2, respectively), as expected, the benchmark setup in case a) shows the best rate-distortion performance, since the reference disparity is known before decoding. Comparing cases b) and c), I find a significant performance gain due to the improved knowledge of correlation statistics due to dynamic estimation. Moreover, all the adaptive DMIC schemes (cases a), b) and c)) outperform the non-adaptive schemes (cases d) and e)). Besides, in the case without adaptive decoding, I find that the performance of joint bit-plane DMIC in case d) is still better than separate bit-plane DMIC in case e). One possible reason for this is that the joint bit-plane DMIC in case d) can exploit the correlation between two non-binary sources much better, since in case e), each bit-plane is decoded separately.

The performance of JPEG2000 codec is also shown as references in Fig. 6.4 and Fig. 6.5. One can see that JPEG2000 is still unreachable due to its high compression efficiency and used arithmetic entropy coding at the cost of high encoding complexity. Note that in contrast to JPEG2000, the proposed DMIC scheme does not employ any transform. In addition, I also plot the theoretical rate as in [2].

Fig. 6.6 and Fig. 6.7 show the final estimate of the correlation and the disparity for solar images in SET 1 and SET 2, respectively, where the reference disparity and residual after 4×4 block matching between source and side information are provided as references. We can see that the proposed adaptive DMIC scheme outputs a good estimate for both correlation and disparity. This also explains why the rate-

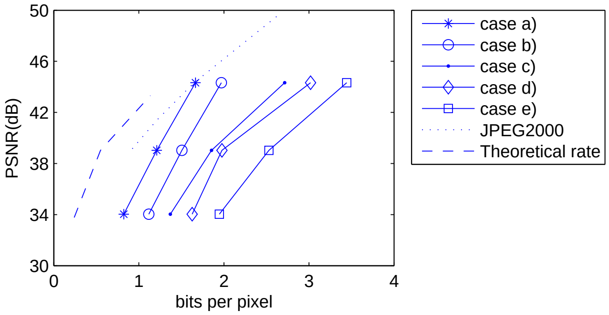

Figure 6.4: Rate-distortion performance of the proposed adaptive DMIC scheme for solar images in SET 1.

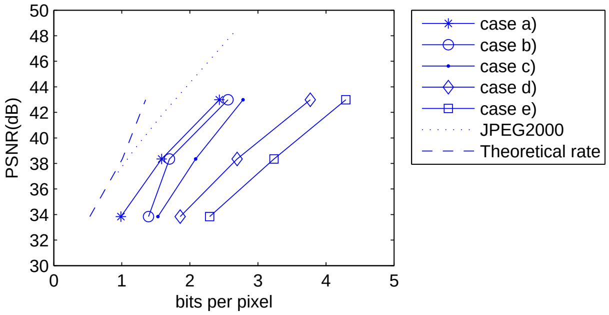

Figure 6.5: Rate-distortion performance of the proposed adaptive DMIC scheme for solar images in SET 2.

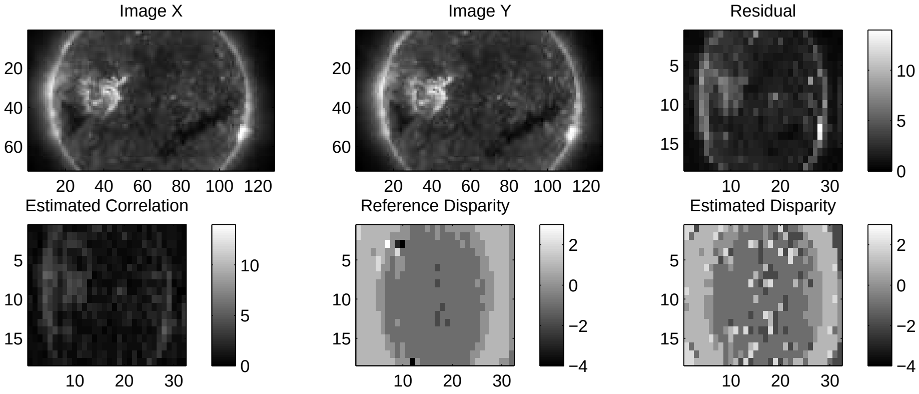

Figure 6.6: The final estimate of the correlation and the disparity for solar images in SET 1, where the true disparity and residual after 4×4 block matching between source and side information are provided as references.

distortion performance of adaptive decoding outperforms the non-adaptive decoding scheme in Fig. 6.4 and Fig. 6.5.

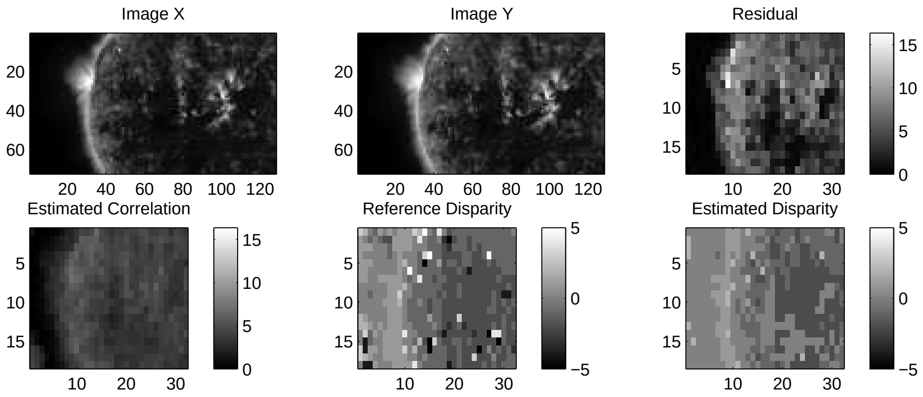

Figure 6.7: The final estimate of the correlation and the disparity for solar images in SET 2, where the true disparity and residual after 4×4 block matching between source and side information are provided as references.

6.6 Summary

This chapter is motivated by the limited onboard processing and communications requirements of correlated images captured by different telescopes or satellites. Traditionally, these images are compressed independently using state-of-the-art, low-complexity compression algorithms such as JPEG without considering the spatial and temporal correlation among images captured by deep-space satellites. In order to exploit the correlation among the multiple views acquired from a solar event and enhance compression without jeopardizing the encoding onboard complexity and independent encoding process to minimize communication complexity, I proposed an adaptive DMIC algorithm, which can estimate the correlation and disparity between stereo images, and decode image sources simultaneously. To handle spatially-varying correlation between stereo images, I extend my previous PBP work [34] for correlation estimation to the 2-D case. Moreover, the correlation and disparity estimation algorithms are all based on an augmented factor graph, which offers great flexibility for problem modeling in remote-sensing applications. Through the results, a significant decoding performance gain has been observed by using the proposed adaptive scheme, when comparing with the non-adaptive decoding scheme. While the proposed scheme performs worse than JPEG2000, the latter has significantly higher encoding complexity comparing to mine.

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 Conclusion

In this dissertation, I mainly focus on the study of online channel noise estimation and correlation estimation by using both stochastic and deterministic approximation inference on factor graphs.

In channel coding, BP is a powerful algorithm to decode LDPC codes over AWGN channels. However, the traditional BP algorithm cannot adapt efficiently to the statistical change of SNR in an AWGN channel. To solve the problem, two common workarounds in approximate inference are stochastic methods (e.g. PF) and deterministic approximation methods (e.g. EP). Generally, deterministic approximation methods are much faster than stochastic methods. However, stochastic methods are much more flexible and suitable for any distribution. In this dissertation, I proposed two adaptive LDPC decoding schemes, which are able to perform online estimation of time-varying SNR at the bit-level by incorporating PF and EP algorithms. Through experimental results, I compare the performance between the proposed PF based and EP based approaches, which shows that the proposed EP based approach obtains the comparable estimation accuracy with less computational complexity than PF based method for both stationary and time-varying SNR, and enhances the BP decoding performance simultaneously. Moreover, the proposed EP estimator shows a very fast convergence speed, and the additional computational overhead of the proposed decoder is less than 10% of the standard BP decoder.

Moreover, since the close relationship between source coding and channel coding, the proposed ideas are extended to source correlation estimation. First, I study the correlation estimation problem in lossless DSC setup, where I consider both asymmetric and non-asymmetric SW coding of two binary correlated sources. The aforementioned PF and EP based approaches are extended to handle the correlation between two binary sources, where the relationship is modeled as a virtual BSC with a time-varying crossover probability. Moreover, to handle the correlation estimation problem of WZ coding, a lossy DSC setup, I design a joint bit-plane model, by which the PF based approach can be applied to tracking the correlation between non-binary sources. Through experimental results, the proposed correlation estimation approaches significantly improve the compression performance of DSC.

Finally, due to the property of ultra-low encoding complexity, DSC is a promising technique for many tasks, in which the encoder has only limited computing and communication power, e.g. the space imaging systems. In this dissertation, I consider a real-world application of the proposed correlation estimation scheme on the onboard low-complexity compression of solar stereo images, since low-complexity compression solutions are essential to reduce onboard storage, processing, and communication resources. In this dissertation, I propose an adaptive distributed compression solution using particle filtering that tracks correlation, as well as performing disparity estimation, at the decoder side. The proposed algorithm is tested on the stereo solar images captured by the twin satellites system of NASAs STEREO project. The results show the significant PSNR improvement over traditional separate bit-plane decoding without dynamic correlation and disparity estimation.

7.2 Further Work

The idea of time/spatially-varying correlation estimation using EP can be easily extended to improve the decoding performance of other distributed coding system, such as distributed video coding (DVC), distributed multi-view coding (DMVC) and so on. Moreover, similar idea would be extended to estimate other hidden parameters, e.g. the disparity of among stereo images. The further implementation of EP for distributed coding system will be studied in the future research work.

BIBLIOGRAPHY

[1] T. A. Summers and S. G. Wilson, “SNR mismatch and online estimation in turbo decoding,” *IEEE Trans. Commun*, vol. 46, pp. 421–423, 1998.

[2] S. Cheng and Z. Xiong, “Successive refinement for the wyner-ziv problem and layered code design,” *IEEE Transactions on Signal Processing*, vol. 53, no. 8 Part 2, pp. 3269–3281, 2005.

[3] R. Gallager, “Low-density parity-check codes,” *IEEE Transactions on Information Theory*, vol. 8, no. 1, pp. 21–28, 1962.

[4] F. R. Kschischang, B. J. Frey, and H. A. Loeliger, “Factor graphs and the sum-product algorithm,” *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498–519, 2001.

[5] D. J. C. MacKay, S. T. Wilson, and M. C. Davey, “Comparison of constructions of irregular gallager codes,” *IEEE Transactions on Communications*, vol. 47, no. 10, pp. 1449–1454, 1999.

[6] D. J. C. MacKay and R. M. Neal, “Near shannon limit performance of low density parity check codes,” *Electronics Letters*, vol. 32, no. 18, 1996.

[7] Z. Xiong, A. Liveris, and S. Cheng, “Distributed source coding for sensor networks,” *IEEE Signal Process. Magazine*, vol. 21, pp. 80–94, Sep. 2004.

[8] S. S. Pradhan, J. Kusuma, and K. Ramchandran, “Distributed compression in a dense microsensor network,” *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 51–60, 2002.

[9] D. Slepian and J. Wolf, “Noiseless coding of correlated information sources,” *IEEE Trans. Inform. Theory*, vol. 19, pp. 471–480, Jul. 1973.

[10] A. Wyner and J. Ziv, “The rate-distortion function for source coding with side information at the decoder,” *IEEE Trans. Inform. Theory*, vol. 22, pp. 1–10, Jan. 1976.

[11] C. E. Shannon, “A mathematical theory of communication.” *Bell System Technical Journal*, vol. 27, pp. 379-423 and 623-656, Jul and Oct, 1948.

[12] S. S. Pradhan and K. Ramchandran, “Distributed source coding using syndromes (discus): design and construction,” in *Proc. DCC*, pp. 158–167, 1999.

[13] A. Wyner, “Recent results in the Shannon theory,” *IEEE Trans. Inform. Theory*, vol. 20, pp. 2–10, Jan. 1974.

[14] V. Stankovic, A. D. Liveris, Z. Xiong, and C. N. Georghiades, “On code design for the Slepian-Wolf problem and lossless multiterminal networks,” *Information Theory, IEEE Transactions on*, vol. 52, no. 4, pp. 1495–1507, 2006.

[15] S. Pradhan and K. Ramchandran, “Generalized coset codes for distributed binning,” *Information Theory, IEEE Transactions on*, vol. 51, no. 10, pp. 3457–3474, 2005.

[16] M. Marcellin and T. Fischer, “Trellis coded quantization of memoryless and gaussian-markov sources,” *IEEE Trans. Communications*, vol. 38, pp. 82–93, Jan. 1990.

[17] G. Ungerboeck, “Channel coding with multilevel/phase signals,” *IEEE Trans. Inform. Theory*, vol. 28, pp. 55–67, Jan 1982.

[18] Y. Yang, S. Cheng, Z. Xiong, and Z. Wei, “Wyner-Ziv coding based on tcq and LDPC codes,” in *Proc. Asilomar*, vol. 1, pp. 825–829, 2003.

[19] A. Liveris, Z. Xiong, and C. Georghiades, “Nested convolutional/turbo codes for the binary Wyner-Ziv problem,” in *Proc. ICIP’03*, (Barcelona, Spain), Sep 2003.

[20] J. Chou, S. Pradhan, and K. Ramchandran, “Turbo and trellis-based constructions for source coding with side information,” in *Proc. DCC’03*, (Snowbird, UT), Mar 2003.

[21] D. Rebollo-Monedero and R. Zhang and B. Girod, “Design of optimal quantizers for distributed source coding,” in *Proc. DCC’03*, (Snowbird, UT), Mar 2003.

[22] H. Feng, Q. Zhao, and M. Effros, “Network source coding using entropy constrained dithered quantization,” in *Proc. DCC’03*, (Snowbird, UT), Mar 2003.

[23] P. Mitran and J. Bajcsy, “Coding for the Wyner-Ziv problem with turbo-like codes,” in *Proc. ISIT’02*, (Lausanne, Switzerland), Jun 2002.

[24] X. Wang and M. Orchard, “Design of trellis codes for source coding with side information at the decoder,” in *Proc. DCC’01*, (Snowbird, UT), Mar 2001.

[25] S. Servetto, “Lattice quantization with side information,” in *Proc. DCC’00*, (Snowbird, UT), Mar 2000.

[26] D. Schonberg, K. Ramchandran, and S. S. Pradhan, “Distributed code constructions for the entire Slepian-Wolf rate region for arbitrarily correlated sources,” in *Data Compression Conference, 2004. Proceedings. DCC 2004*, pp. 292–301, 2004.

[27] A. Wyner, "The rate-distortion function for source coding with side information at the decoder-II: General sources," *Information and Control*, vol. 38, no. 1, pp. 60–80, 1978.

[28] S. Wang, L. Cui, S. Cheng, Y. Zhai, M. Yeary, and Q. Wu, "Noise adaptive LDPC decoding using particle filtering," *IEEE Transactions on Communications*, vol. 25, pp. 1–4, 4 2011.

[29] S. Wang, L. Cui, and S. Cheng, "Noise Adaptive LDPC Decoding Using Expectation Propagation," in *IEEE Global Telecommunications Conference*, 2011.

[30] S. Wang, L. Cui, and S. Cheng, "Noise Variance Estimation over AWGN Channel using Approximate Inference," *submitted to IEEE Transactions on Communications.*

[31] L. Cui, S. Wang, S. Cheng, and M. Yeary, "Adaptive binary Slepian-Wolf decoding using particle based belief propagation," *IEEE Transactions on Communications*, no. 99, pp. 1–6, 2011.

[32] S. Wang, L. Cui, L. Stnakovic, V. Stankovic, and S. Cheng, "Adaptive correlation estimation with particle filtering for distributed video coding," *accepted by IEEE Transactions on Circuits and Systems for Video Technology.*

[33] S. Cheng, S. Wang, and L. Cui, "Adaptive Slepian-Wolf decoding using particle filtering based belief propagation," in *47th Annual Conference on Communication, Control, and Computing*, pp. 607–612, IEEE, 2009.

[34] S. Wang, L. Cui, and S. Cheng, "Adaptive Wyner-Ziv decoding using particle-based belief propagation," in *2010 IEEE Global Telecommunications Conference GLOBECOM*, IEEE, 2010.

[35] S. Cheng, S. Wang, and L. Cui, "Adaptive nonasymmetric Slepian-Wolf decoding using particle filtering based belief propagation," in *2010 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP),*, pp. 3354–3357, IEEE, 2010.

[36] L. Cui, S. Wang, and S. Cheng, "Adaptive Slepian-Wolf decoding based on expectation propagation," *Accepted by IEEE Communications letter.*

[37] S. Wang, L. Cui, S. Cheng, L. Stankovic, and V. Stankovic, "Onboard Low-complexity Compression of Solar Images," in *IEEE International Conference on Image Processing*, 2011.

[38] S. Wang, L. Cui, S. Cheng, L. Stankovic, and V. Stankovic, "Onboard low-complexity compression of solar stereo images," *submitted to IEEE Transactions on Image Processing.*

[39] C. Bishop, *Pattern recognition and machine learning*, vol. 4. Springer New York:, 2006.

[40] H. Loeliger, J. Dauwels, J. Hu, S. Korl, L. Ping, and F. Kschischang, "The factor graph approach to model-based signal processing," *Proceedings of the IEEE*, vol. 95, no. 6, pp. 1295–1322, 2007.

[41] T. Minka, "Expectation propagation for approximate Bayesian inference," in *Uncertainty in Artificial Intelligence*, vol. 17, pp. 362–369, Citeseer, 2001.

[42] P. Felzenszwalb and D. Huttenlocher, "Efficient belief propagation for early vision," *International journal of computer vision*, vol. 70, no. 1, pp. 41–54, 2006.

[43] S. Kullback and R. Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[44] M. Bolic, P. M. Djuric, and S. Hong, "Resampling algorithms for particle filters: A computational complexity perspective," *EURASIP Journal on Applied Signal Processing*, vol. 15, no. 5, pp. 2267–2277, 2004.

[45] S. Chib and E. Greenberg, "Understanding the Metropolis-Hastings algorithm," *American Statistician*, vol. 49, no. 4, pp. 327–335, 1995.

[46] L. Cui, S. Wang, S. Cheng, and Q. Wu, "Noise adaptive LDPC decoding using particle filter," in *43rd Annual Conference on Information Sciences and Systems, 2009.*, pp. 37–42, IEEE, 2009.

[47] X. Wang and R. Chen, "Adaptive Bayesian multiuser detection for synchronous CDMA with Gaussian and impulsive noise," *IEEE Transactions on Signal Processing*, vol. 47, no. 7, p. 2013, 2000.

[48] X. Wei and T. Weber, "MMSE detection based on noise statistics with random noise variance," in *IEEE 68th Vehicular Technology Conference*, pp. 1–5, 2008.

[49] A. Papoulis, S. Pillai, and S. Unnikrishna, *Probability, Random Variables, and Stochastic Processes.* McGraw-Hill New York, 2002.

[50] T. Richardson, M. Shokrollahi, and R. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 619–637, 2001.

[51] M. Opper, "A Bayesian approach to on-line learning," *On-line learning in neural networks*, pp. 363–378, 1998.

[52] V. Ramon, C. Herzet, L. Vandendorpe, and M. Moeneclaey, "EM algorithm-based estimation of amplitude, carrier phase and noise variance in multiuser turbo receivers," in *ISSSTA*, pp. 550–554, 2004.

[53] A. Wyner, “Recent results in the Shannon theory,” *Information Theory, IEEE Transactions on*, vol. 20, no. 1, pp. 2–10, 1974.

[54] A. Liveris, Z. Xiong, and C. Georghiades, “Compression of binary sources with side information at the decoder using LDPC codes,” *IEEE Communications Letters*, vol. 6, pp. 440–442, Oct. 2002.

[55] A. Liveris, Z. Xiong, and C. Georghiades, “Joint source-channel coding of binary sources with side information at the decoder using IRA codes,” in *Proc. MMSP’02 IEEE Workshop on Multimedia Signal Process.*, (St. Thomas, US Virgin Islands), Dec. 2002.

[56] T. Moon, *Error Correction Coding: Mathematical Methods and Algorithms.* Wiley-Blackwell, 2005.

[57] B. Rimoldi and R. Urbanke, “Asynchronous Slepian-Wolf coding via source-splitting,” in *ISIT’97*, (Ulm, Germany), p. 271, 1997.

[58] H. Jin, A. Khandekar, and R. McEliece, “Irregular repeat-accumulate codes,” in *Proc. 2nd Int. Symp. Turbo Codes and Related Topics*, pp. 1–8, 2000.

[59] `http://www.nasa.gov/mission\_pages/stereo/mission/index.html`.

[60] “http://24sevenpost.com/paranormal/ufo-spotted-visiting-sun-nasa-image,” *Accessed May 17th, 2011.*

[61] G. Yua, T. Vladimirova, and M. Sweeting, “Image compression systems on board satellites,” *Elsevier Acta Astronautica*, vol. 64, pp. 988–1005, 2009.

[62] I. 15444-1, “Information technology—JPEG2000 image coding system, part 1: Core coding system,” 2000.

[63] S. Seo and M. Azimi-Sadjadi, “A 2-D filtering scheme for stereo image compression using sequential orthogonal subspace updating,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 1, pp. 52–66, 2002.

[64] Y. Wang, S. Rane, P. Boufounos, and A. Vetro, “Distributed compression of zerotrees of wavelet coefficients,” in *Proc. ICIP-2011 IEEE International Conference on Image Processing*, 2011.

[65] A. Abrardo, M. Barni, E. Magli, and F. Nencini, “Error-resilient and low-complexity onboard lossless compression of hyperspectral images by means of distributed source coding,” *IEEE Trans. Geoscience and Remote Sensing*, vol. 48, no. 4, pp. 1892–1904, 2010.

[66] L. Stankovic, V. Stankovic, S. Wang, and S. Cheng, “Distributed Video Coding with Particle Filtering for Correlation Tracking,” *Proc. EUSIPCO, Aalborg, Denmark*, 2010.

[67] D. Varodayan, A. Mavlankar, M. Flierl, and B. Girod, "Distributed grayscale stereo image coding with unsupervised learning of disparity," in *IEEE Data Compression Conference*, pp. 143–152, 2007.

[68] R. Nagura, "Multi-stereo imaging system using a new data compression method," in *Proc. IGARSS-2006 IEEE International Geoscience and Remote Sensing Symposium*, 2006.

[69] A. Kiely, A. Ansar, A. Castano, M. Klimesh, and J. Maki, "Lossy image compression and stereo ranging quality from Mars rovers," *IPN Progress Report 42-168*, 2007.

[70] "Image compression algorithm altered to improve stereo ranging," *Tech Brifs, available at http://www.techbriefs.com/component/content/article/2673*, 2007.

[71] P. Meyer, R. Westerlaken, R. Gunnewiek, and R. Lagendijk, "Distributed source coding of video with non-stationary side-information," in *Proc. SPIE*, vol. 5960, pp. 857–866, Citeseer, 2005.

[72] M. Dalai, R. Leonardi, and F. Pereira, "Improving turbo codec integration in pixel-domain distributed video coding," in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, vol. 2, pp. II–II, IEEE.

[73] C. Brites and F. Pereira, "Correlation noise modeling for efficient pixel and transform domain Wyner–Ziv video coding," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 18, no. 9, pp. 1177–1190, 2008.

[74] X. Fan, O. Au, and N. Cheung, "Adaptive correlation estimation for general Wyner-Ziv video coding," in *Image Processing (ICIP), 2009 16th IEEE International Conference on*, pp. 1409–1412, IEEE, 2009.

[75] X. Huang and S. Forchhammer, "Improved virtual channel noise model for transform domain Wyner-Ziv video coding," pp. 921–924, 2009.

[76] X. Artigas, J. Ascenso, M. Dalai, S. Klomp, D. Kubasov, and M. Ouaret, "The DISCOVER codec: architecture, techniques and evaluation," in *Picture Coding Symposium*, Citeseer, 2007.

[77] A. Aaron, S. Rane, and B. Girod, "Wyner-Ziv video coding with hash-based motion compensation at the receiver," in *IEEE ICIP'04*, vol. 5, pp. 3097–3100, 2005.

[78] G. Petrazzuoli, M. Cagnazzo, and B. Pesquet-Popescu, "Fast and efficient side information generation in distributed video coding by using dense motion representations," *EUSIPCO, Aalborg, Denmark*, 2010.

[79] D. Varodayan, Y. Lin, A. Mavlankar, M. Flierl, and B. Girod, "Wyner-Ziv coding of stereo images with unsupervised learning of disparity," in *Proc. Picture Coding Symp., Lisbon, Portugal*, Citeseer, 2007.

[80] D. Chen, D. Varodayan, M. Flierl, and B. Girod, "Distributed stereo image coding with improved disparity and noise estimation," in *IEEE ICASSP*, pp. 1137–1140, 2008.

[81] A. Zia, J. P. Reilly, and S. Shirani, "Distributed parameter estimation with side information: A factor graph approach," in *IEEE ISIT*, pp. 2556–2560, 2007.

[82] V. Toto-Zarasoa, A. Roumy, and C. Guillemot, "Maximum Likelihood BSC parameter estimation for the Slepian-Wolf problem," *Communications Letters, IEEE*, vol. 15, no. 2, pp. 232–234, 2011.

[83] N. Gehrig and P. Dragotti, "Distributed compression of multi-view images using a geometrical coding approach," in *Proc. ICIP-2007*, 2007.

[84] X. Zhu, A. Aaron, and B. Girod, "Distributed compression for large camera arrays," in *Proc. SSP-2003*, 2003.

[85] A. Jagmohan, A. Sehgal, and N. Ahuja, "Compression of light-field rendered images using coset codes," in *Proc. 37th Asilomar Conference on Signals, Systems, and Computers*, 2003.

[86] G. Toffetti, M. Tagliasacchi, M. Marcon, A. Sarti, S. Tubaro, and K. Ramchandran, "Image compression in a multi-camera system based on a distributed source coding approach," in *Proc. Eusipco-2005*, 2005.

[87] L. Stankovic, V. Stankovic, and S. Cheng, "Distributed compression: Overview of current and emerging multimedia applications," in *ICIP-2011 IEEE International Conference on Image Processing*, IEEE, 2011.

[88] C. Guillemot, F. Pereira, L. Torres, T. Ebrahimi, R. Leonardi, and J. Ostermann, "Distributed monoview and multiview video coding: basics, problems and recent advances," *IEEE Signal Processing Magazine*, vol. 24, no. 5, pp. 67–76, 2007.

[89] R. Zamir, S. Shamai, and U. Erez, "Nested linear/lattice codes for structured multiterminal binning," *IEEE Trans. Inform. Theory*, vol. 48, no. 6, pp. 1250–1276, 2002.